Principles of Simulation

Dr. Jane L. Harvill ©
Department of Statistical Science
Baylor University

September 24, 2012

BAYLOR
UNIVERSITY

COLLEGE OF ARTS & SCIENCES
Department of Statistical Science

1 Introduction

In statistics, Monte Carlo simulation is a tool used to investigate the properties of a statistic under what is often called "repeated sampling." Bootstrap sampling is a tool used to investigate similar properties under "resampling." In this tutorial, we discuss the types of properties that might be investigated, and what is meant by the phrases "repeated sampling" and "resampling." We begin with intuitive expositions on populations and samples, Section 2. We end that section with a brief exposition on the most important distribution in statistics; the "normal distribution." Section 3 introduces basic concepts of inferential statistics, primarily focusing on sampling distributions and one of the most important applicable theorems in the science of statistics, the "Central Limit Theorem." With this background, we explain the reasoning behind Monte Carlo simulation in Section 5. For illustration, we use the *R* function `CLT.sim`. A listing of that function is provided in Appendix B. We complete the monograph with a brief, and elementary exposition on topics in bootstrap sampling.

2 Populations, Samples, and Statistics

One of the primary focuses in the study and application of statistics is finding a reliable method for taking a sample from a larger collection of objects, called the population, and using information from the sample to draw conclusions about the population. Consider, for example, the study on childhood obesity in Texas by Aron (2011). In this study, many conclusions are drawn about why such a high percentage of children in the state are overweight or obese. Aron also draws conclusions about the cost – from both personal health and a public financial costs perspectives – of childhood obesity. To make such conclusions, she and her research team did not go across the entire state and weigh every child and ask their parents probing questions about the personal toll of being overweight, or about the health of their children, or the associated costs. She also did not look at the medical records and billings for all obese people in the State of Texas. Such a task is not even realistic. Instead, she based her conclusions on a subset of the children in the state, and the findings of other researchers. For example, one conclusion drawn by Aron is that 42% of fourth grade children in the state are overweight or obese. In this specific example, the population is all children in the State of Texas. The sample is the subset of children in the state who were used to draw the conclusions in the study. Aron used the subset (sample) of children to make a statement about percentage of all fourth graders in the state who are overweight or obese.

Definition 2.1 Inferential Statistics.
The process of taking information from a sample and using information in the sample to draw conclusions about the population from which the sample was drawn is called **inferential statistics**.

Consider that many numerical measurements can be taken on any individual or object; for example, the body mass index (BMI) of a single fourth grade child in Texas. Although it's impossible to have the BMI of all fourth grade children in the state of Texas, that does

not mean that every fourth-grade child does not have a BMI. Of course, *every* child has a BMI! In fact, the BMI will change from one child to the next.

Definition 2.2 Variable.
A numerical characteristic that changes from one individual or object to the next is called a **variable**.

Methods of inferential statistics can only be properly applied when the units (objects or individuals) included in the sample are selected according to some random mechanism, thus resulting in a random sample. From that perspective, the variables to be measured are themselves considered to be random.

Definition 2.3 Random Sample and Random Variable.
A sample selected by using a random device to determine which members of the population are included in the sample is a **random sample**. *A* **random variable** *is a numerical variable whose value depends upon chance.*

It is convention to denote random variables using capital letters. In contrast, when referring to a specific numerical value the random variable might be, the notation is a lower-case letter that is the same letter as the random variable. For example, suppose the variable of interest is the BMI of all fourth grade children in the state of Texas. If a fourth grade child is selected at random from all fourth grade children in Texas, one might ask the question, "What is the chance that the BMI of a randomly selected fourth grade child (the random variable X) is equal to $x = 17.2$?" Note: it is not necessary to specify 17.2 to use a lower-case letter. The point is that, when a lower-case letter is used, it is referring to some numerical value, whereas a capital letter is indicating a variable that is random (has a value that depends upon chance).

One important property of a random variable is a mathematical description of the the pattern of all the values; for example, the pattern of BMIs for all fourth grade children in the state of Texas. Although it may be impossible to observe every value in a population, it is usually possible to describe the pattern that all of those values follow.

Definition 2.4 Distribution Function.
For a random variable X, a mathematical function $f(x)$ satisfying

1. *$f(x) \geq 0$ for all values of x, and*
2. *the area under $f(x)$ is equal to one,*

is called a **probability distribution function**.

There are many mathematical formulas that satisfy those two conditions. The mathematical formulas obviously include the variable of interest. But they also include other quantities that summarize the population as a whole.

Definition 2.5 PARAMETER.
A **parameter** *is any numerical characteristic of a population that summarizes a particular feature of the probability function or equivalently, of the population of interest. Lower-case Greek letters are often used to denote parameters.*

For example, the average BMI of all fourth-graders in the state is a numerical characteristic that summarizes a particular characteristic (the average) of the population of all fourth-graders' BMIs. A very popular notation to represent the mean of a population is the lower-case Greek letter μ. So we could write $\mu =$ the average BMI of all fourth-grades in Texas.

A particularly important probability distribution is the normal distribution. The parameters of a normal distribution are the mean, denoted by μ, and standard deviation, denoted by σ. The normal distribution is commonly known as the "bell-shaped curve." Figure 1 contains three curves, each of a different normal distribution. The three curves are all a

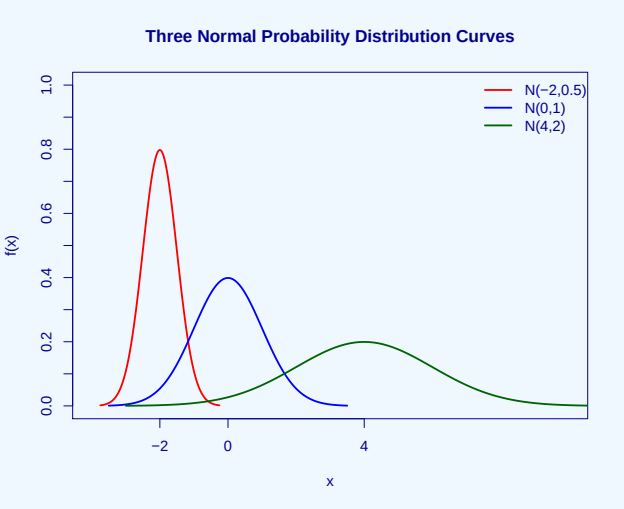

Figure 1: Three normal probability distribution curves. The red curve is a normal distribution with mean $\mu = -2$ and standard deviation $\sigma = 0.5$. The blue curve is a normal distribution with mean $\mu = 0$ and $\sigma = 1$ (the standard normal distribution). The green curve is a normal distribution with mean $\mu = 4$ and standard deviation $\sigma = 2$.

function of a numerical variable x. The general shape of the curve gives us an idea of how the random variable X behaves across all possible numerical values that it can be. They all three have the same general shape (a bell-shaped curve). The height of all three curves is greatest at the center of the curve, indicating that values closest to the center are more dense (or more likely to occur) than values farther from the center. All three curves are greater than zero. The area under each of the three curves is equal to one. So they all have similar *overall* characteristics. But they are also different in some *specific* ways.

One difference between the three curves is that their centers are located over different values on the horizontal axis. The red curve's center is over the value -2; the blue curve's

center is over zero; and the green curve has a center over four. For any normal distribution, the center of the curve is located over the average. In fact, for any distribution curve, including the normal curve, the mean can accurately be interpreted as the center of mass of the distribution curve. Consequently, the mean is often called a "location parameter," or a "measure of center."

Another difference in the three normal curves in Figure 1 is their height, and correspondingly their width. Loosely speaking, the horizontal distance of the curve, as measured from the mean (center), is measured in units called the standard deviation. The standard deviation of a population distribution is usually denoted by a lower-case Greek σ. A reasonable interpretation of one standard deviation is that it is the typical (standard) difference (deviation) of a single value from the population mean. Notice that as the standard deviation decreases the height of the normal curve increases, and its width decreases. These changes must occur simultaneously because the area under the curve must be one for the curve to be a valid probability distribution function. Because the standard deviation of a normal distribution has an effect on the height of the distribution, it is often called the "scale parameter."

For a population that has a normal distribution, the standard deviation can be interpreted in a very specific way. This is such a special result, it has its own name: the Empirical Rule.

Theorem 2.1 The Empirical Rule.
If a population follows a normal distribution, then

1. *68.26% of all values in the population are within one standard deviation of the population mean; that is, lie in the interval* $(\mu - \sigma, \mu + \sigma)$
2. *95.45% of all values in the population are within two standard deviations of the population mean; that is, lie in the interval* $(\mu - 2\sigma, \mu + 2\sigma)$*, and*
3. *99.73% of all values in the population are within three standard deviations of the population mean; that is, lie in the interval* $(\mu - 3\sigma, \mu + 3\sigma)$.

Since the area under the curve must be equal to one, these percentages can be converted to decimals. In that form, they represent the area under the curve between standard deviations. This is illustrated graphically in Figure 2. The really amazing feature of the Empirical Rule is that it is true for any value of μ and any positive value of σ! In other words, it doesn't matter what value the mean is, or what value of the standard deviation is (as long as it's greater than zero), the Empirical Rule holds for every normal distribution.

3 Sampling Distributions

In the practice of inferential statistics, it is often the case that we are willing to assume we have some information about the shape of the population distribution function, and we want to estimate the parameters of that distribution. In the case of BMI of all fourth-graders, it might be reasonable to assume that the pattern of all the BMIs follow a bell-shaped curve

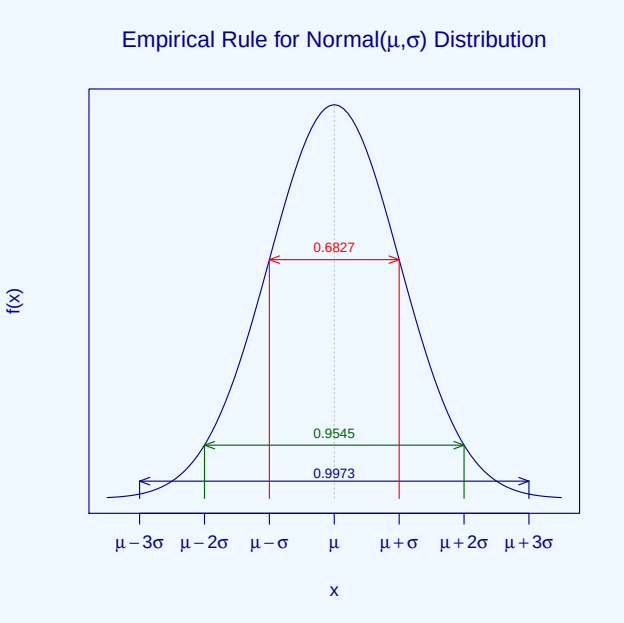

Figure 2: Graphical illustration of the Empirical Rule (Theorem 2.1). The area under the curve within one standard deviation of the mean, between $\mu-\sigma$ and $\mu+\sigma$ (represented in red) is 0.6826. The area under the curve within two standard deviations of the mean, between $\mu-2\sigma$ and $\mu+2\sigma$ (represented in dark green) is 0.9545. The area under the curve within three standard deviations of the mean, between $\mu-3\sigma$ and $\mu+3\sigma$ is 0.9973 (represented in dark blue). The light gray dotted vertical line is over the mean μ of the normal distribution curve.

(a normal distribution)[1]. On the other hand, it is not reasonable to think that we know the population mean (μ) BMI of all fourth-graders or the population standard deviation (denoted by σ) of the BMI of all fourth-graders. So to estimate these, we take a random sample of fourth-graders, measure their BMI, and use the average of the sample to estimate the average of the population or the standard deviation of the sample to estimate the standard deviation of the population.

Definition 3.1 Statistic.
A value computed from data in a sample is called a **statistic**.

Definition 3.2 Sample Size and Sample Mean.
Consider taking a random sample of n objects and measuring a numerical characteristic on each object in the sample. Denote those n measurements by $x_1, x_2, \ldots, x_n$. The value of n is called the **sample size**. *The* **sample mean** *is denoted by $\bar{x}$, and is computed using the*

[1]Work by the Belgian astronomer, mathematician, statistician, and sociologist Adolphe Quételet established the definition of BMI, as well as why it is reasonable to assume that all values of BMI follow a normal distribution.

formula

$$\bar{x} = \frac{1}{n}(x_1 + x_2 + \cdots + x_n) = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

Or more simply put, the sample average is the sum of the values from the sample divided by the number of individuals in the sample. The notation $\sum_{i=1}^{n} x_i$ is called "summation notation," and is just a short-hand way of writing $x_1 + x_2 + \ldots + x_n$. Notice the bottom index on the $\sum$ corresponds to the subscript on the x; the smallest value of the i is 1, and i goes up to n.

Although we will not focus much on the sample standard deviation, we present its definition here for the sake of completeness.

Definition 3.3 Sample Variance and Sample Standard Deviation.
Consider taking a random sample of n objects and measuring a numerical characteristic on each object in the sample Denote these n measurements by $x_1, x_2, \ldots, x_n$. The **sample variance** *is denoted by s^2 and is computed using the formula*

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2.$$

The **sample standard deviation**, *denoted s, is equal to the positive square root of the sample variance; that is*

$$s = \sqrt{s^2}.$$

Now that the basics of populations, samples, and statistics have been discussed, the next layer in preparing to explain Monte Carlo simulation in the context of statistical science is the concept of a "sampling distribution." Sampling distributions describe the pattern of all possible values of a statistic that arise from the many different possible random samples from the population. The concept of observing many different samples of size n from the same population is what is often referred to as "repeated sampling."

In the remainder of our discussion on the sampling distribution of a statistic (under repeated sampling), we will restrict our discussion primarily to the behavior of the sample mean. It should be noted however, that similar logical arguments can be made for any statistic (like the sample variance or sample standard deviation),even though the details will not be the same.

We begin illustrating exactly what is meant by a "sampling distribution" and by "repeated sampling" with Example 3.1. In this example, we take all possible samples of size $n = 3$ from a population of size $N = 10$ individuals. Admittedly, the population in this example is small, and there is no need to sample from it over and over again. However, we use this small population to aid as an illustration for what happens in realistic situations when the population is millions of individuals.

Example 3.1 Illustration of a Sampling Distribution.
Suppose we were interested in estimating the mean of a population of the 10 measurements

given below. (We're pretending that, for some reason, we can't compute $\mu = 18.0958$ *or that* $\sigma = 1.1856$*.)*

$$17.090,\ 19.101,\ 17.883,\ 19.134,\ 20.622,\ 16.509,\ 18.459,\ 16.996,\ 17.849,\ 17.316. \qquad (1)$$

From this "population" of 10 measurements, there are exactly 120 possible samples of size $n = 3$*. For the sake of discussion, associate with each measurement in the population the integers* $1, 2, 3, \ldots, 10$*. A partial listing the 120 samples (using the integers associated with the measurements) is*

$$(1,2,3), \quad (1,2,4), \quad (1,2,5), \quad \cdots, \quad (4,8,9), \quad \cdots, \quad (8,9,10).$$

For each of the 120 different samples of three measurements from the 10 in the population, there is a slightly different value of $\bar{x}$*. For example, if the first three listed measurements were the sample that happened to be selected, then*

$$\bar{x} = \frac{1}{3}(17.090 + 19.101 + 17.883) = 18.024.$$

On the other hand, if measurements labeled 4, 7, and 8 were the measurements selected for the sample, then

$$\bar{x} = \frac{1}{3}(19.134 + 18.459 + 16.996) = 18.196.$$

If we were so inclined, we could compute all 120 values of $\bar{x}$*, and draw a histogram of those values. We used a computer to do just that. The corresponding histogram is seen in Figure 3. This histogram is the sampling distribution of the 120 values of* $\bar{x}$*. Appendix C contains a*

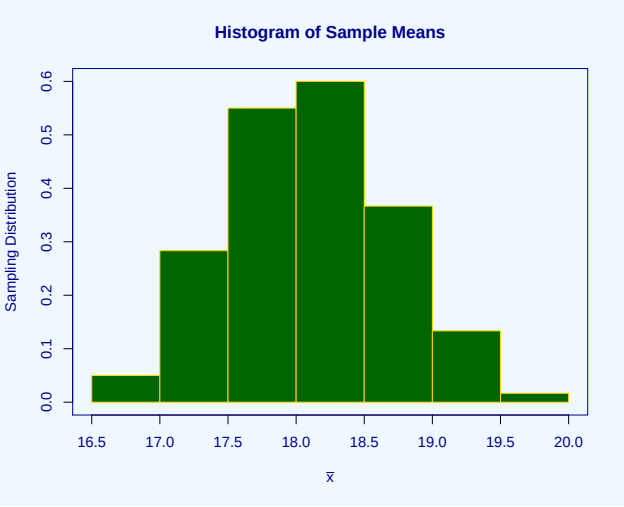

Figure 3: Histogram of 120 possible sample means based on samples of size $n = 3$ from the population of 10 measurements in (1.)

listing of all of the 120 samples using the integers associated with the measurements. The following page contains the corresponding 120 values of $\bar{x}$. Point of interest: *We averaged all the 120 values of $\bar{x}$ (listed in the appendix), and the result was 18.0958. How does this compare the the value of the population mean that the sample mean is intended to estimate? Your observation (that the two numbers are identical) is not a coincidence. We'll explain more in Section 3.1.*

This example aptly illustrates that different simple random samples of size n from the same population have differing values of the sample mean. For extremely large populations (like all fourth-grade children in the State of Texas), the number of samples is unimaginably large. So there are an unimaginable number of different values of the sample mean. Since samples are randomly selected, then before any sample is selected, the sample mean can be considered a random variable, since its value depends upon chance. The same statement can be said of any quantity that is computed from a sample – to any statistic. This leads to the following definition.

Definition 3.4 Sampling Distribution.
The **sampling distribution of a statistic** *is a probability function that describes the pattern of behaviors of the values of the statistic arising from all possible samples of size n from the same population.*

Notice that a sampling distribution is a probability distribution. The adjective "sampling" is used to emphasize that the randomness in the statistic arises from the sample that is selected.

In general, sampling distributions have three characteristics that are of interest:

1. a formula for the mean of the sampling distribution,
2. a formula for the standard deviation of the sampling distribution, and
3. a formula for the shape of the sampling distribution.

In the following discussion, notice that the sample mean is being denoted by a capitalized $\bar{X}$; not a lower-case $\bar{x}$ as was used in the previous discussions. That is because in the following discussion, we are attempting to describe the behavior of the sample mean before the sample is selected. Since samples are randomly selected, the value of $\bar{X}$ is determined by a random mechanism, and so is a random variable. Random variables are properly denoted by capital letters.

3.1 The Mean of the Sampling Distribution of a Statistic

Since each sample of size n will result in a different value of a statistic that is computed based on the sample, then it is possible to envision that, for every possible sample of size n, the average of all possible values of the statistic could be computed. This line of reason can apply to any statistic (the sample mean, the sample median, the sample variance, etc.)

The histogram of all the values of the statistic will be centered around that average. While for an example like 3.1 that is not difficult to accomplish (since there were only 120 samples of size $n = 3$ from our population of 10 individuals), for a typical situation, such a task is impossible. However, there is mathematical theory that will provide mathematical formula for the mean of a statistic from all possible samples.

Definition 3.5 Mean of the Sampling Distribution of $\bar{X}$.
The **mean of the sampling distribution of the sample mean** *is the value around which the sampling distribution of $\bar{X}$ is centered. Suppose that a random sample of size n is taken from a population whose mean is denoted by μ. Then the mean of the sampling distribution of $\bar{X}$ is denoted by $\mu_{\bar{X}}$ and is*

$$\mu_{\bar{X}} = \mu.$$

In other words, the mean of all possible values of the sample mean is the population mean! This implies that all possible values of $\bar{X}$ are centered around the population parameter μ that they are intended to estimate.

Definition 3.6 Unbiased Estimator.
When the mean of the sampling distribution of a statistic is the parameter it is intended to estimate, the statistic is called an **unbiased estimator** *of the parameter.*

Recall from Example 3.1, we pointed out that the mean of the 120 values of $\bar{x}$ was equal to 18.0958, which is the exact same value as the population mean μ. Examine the histogram in Figure 3. Notice that it is centered over 18.0958. That is no coincidence. The formula in Definition 3.5 tells us that $\bar{X}$ is an unbiased estimator of μ, so its sampling distribution will be centered around μ.

The idea of unbiasedness is often likened to accuracy. The sample mean is an unbiased estimator of μ. Therefore the different values of $\bar{X}$ will be centered around μ. Consider the illustration in Figure 4. The exact center of the target represents the parameter value that is to be estimated. Taking a shot at the target represents the action of taking a sample and calculating a statistic to "hit" the center of the target; that is, to estimate the parameter. Each small dark dot represents where the sample "hit" the target. An unbiased estimator will have shots centered around the target. The two targets in the left column illustrate two estimators that are unbiased. The top left target is an estimator that is also precise. On the right, the shots are all centered away from the bull's eye, illustrating a biased estimator. The distance of the center of the cluster of shots from the center of the target is the bias of the estimator.

3.2 The Standard Deviation of the Sampling Distribution of a Statistic

The standard deviation of the sampling distribution of a statistic is, loosely speaking, how far, on average, a typical value of a statistic will deviate from the mean of the sampling distribution of the statistic.

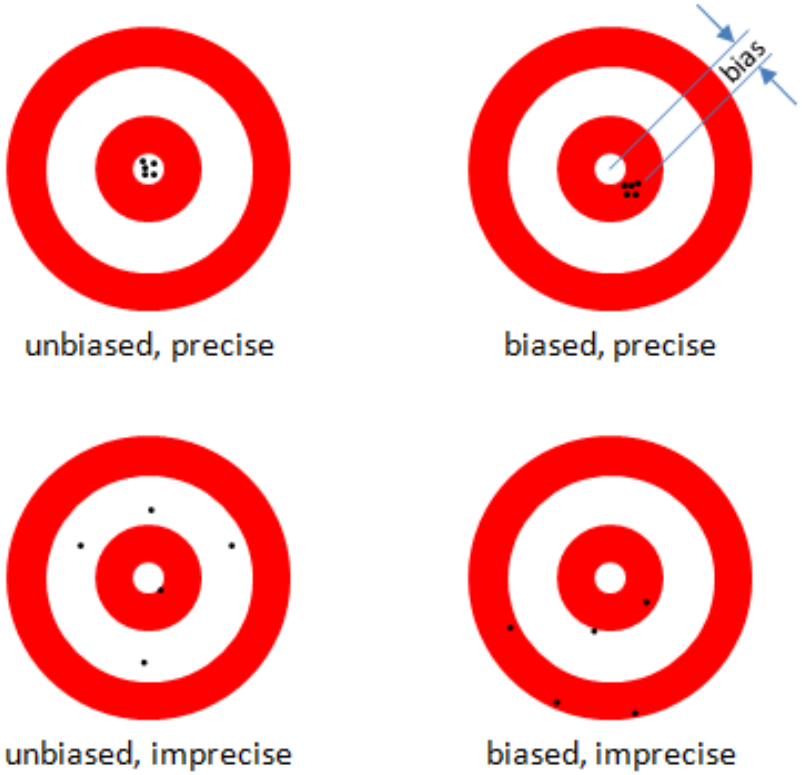

Figure 4: Target illustrating unbiased, precise estimation. The points on the top left target illustrate an unbiased and consistent estimator; the top right target illustrates a precise, but biased estimator; the bottom left illustrates imprecise but unbiased estimator; the bottom right illustrates an imprecise and biased estimator. (Figure from `http://www.statisticalengineering.com/Weibull/precision-bias.html`.)

Definition 3.7 Standard Deviation of the Sampling Distribution of $\bar{X}$.
The **standard deviation of the sampling distribution of the sample mean** *is the amount, on average, that a typical value of $\bar{x}$ will deviate from $\mu_{\bar{X}}$. Suppose that a random sample of size n is taken from a population whose standard deviation is denoted by a σ. Then the standard deviation of the sampling distribution of $\bar{X}$ is denoted by $\sigma_{\bar{X}}$ and is*

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}}.$$

The standard deviation of the sampling distribution is also referred to as the **standard error of the statistic**.

For Example 3.1, computing the standard deviation of all the 120 values of $\bar{x}$ gives 0.6062. This is different from the value provided in the formula in Definition 3.7 because the population size of $N = 10$ is small for a population. When the population size is small, then to compute the standard deviation of the sampling distribution of the statistic, the formula in Definition 3.7 should be multiplied by $\sqrt{(N-n)/(N-1)}$, the "finite population correction factor." Notice that as N gets very large, this quotient gets very close to one, resulting in the same value as the formula from Definition 3.7.

Finally, notice that as the sample size n gets very, very large, the standard deviation of the sample mean gets very close to zero; that is

$$\lim_{n\to\infty} \frac{\sigma}{\sqrt{n}} = 0.$$

Definition 3.8 Consistent Estimator.
A statistic that is an unbiased estimator of a parameter, and that has a standard error with a limit of zero as $n \to \infty$ the statistic is called a **consistent estimator** *of the parameter.*

Since the sample mean $\bar{X}$ is an unbiased estimator of the population mean μ, and the limit (as $n \to \infty$) of the standard deviation of $\bar{X}$ is zero. So $\bar{X}$ is a consistent estimator of μ.

Consistency is a highly desirable property in an estimator. It is likened to precision. Referring back to Figure 4, the two targets across the top represent precise estimators; the values are clustered together tightly. The top left represents a consistent estimator because not only are the values clustered tightly together, they are also centered on the target.

When using a consistent estimator to determine the value of a parameter, the estimator becomes increasingly accurate and precise for larger and larger samples. Thinking about this from the perspective of the sample mean, larger samples provide better estimators of μ (through $\bar{X}$). They are better because they will be more accurate (closer) to the true value of $\bar{X}$, and they will have a smaller standard error than a sample mean based on a smaller sample. Therefore, if precision and accuracy are important in estimation, they can be improved by collecting more data – as long as sufficient resources are available for doing so.

3.3 The Shape of the Sampling Distribution of a Statistic

The third characteristic of sampling distributions is the shape of the distribution, or more precisely, the mathematical formula of the probability function that describes the pattern of all possible values of the statistic. The discussion in this section focuses entirely on the sampling distribution of the sample mean $\bar{X}$. In determining the sampling distribution, there are two cases to be considered.

Case I. If the distribution of the population from which the sample is selected is normally distributed, then the distribution of all possible values of $\bar{X}$ is also normally distributed.

Case II. Case II is one of the most important results in statistics, and is referred to as the **Central Limit Theorem**, stated below in Theorem 3.1.

Theorem 3.1 Central Limit Theorem.
If the distribution of the population from which the sample is selected is not normally distributed, then the distribution of all possible values of $\bar{X}$ may be well-approximated using a normal distribution as long as the sample size n is sufficiently large.

We will spend more time in Section 5 illustrating exactly what is implied in the statements of Cases I and II though Monte Carlo simulation.

3.4 Uses for the Sampling Distribution

The sampling distribution provides the information needed to develop statistical inference procedures – hypothesis tests and confidence intervals – on a population parameter. We discuss both of these procedures in the following subsections, again relying on what we know about the sampling distribution of $\bar{X}$ for purposes of illustration.

3.4.1 Confidence Intervals

In the previous sections, we saw that $\bar{X}$ has a normal distribution with mean $\mu = \mu_{\bar{X}}$ and standard deviation $\sigma_{\bar{X}} = \sigma/\sqrt{n}$. Therefore, we can apply the Empirical Rule (Theorem 2.1). In applying this theorem, "all values in the population" refers to "all values of $\bar{X}$", and the formulas for $\mu_{\bar{X}}$ and $\sigma_{\bar{X}}$ define precisely the mean and standard deviation. So the Empirical Rule, applied to $\bar{X}$ becomes

Theorem 3.2 The Empirical Rule for $\bar{X}$.
Consider taking a random sample of size n from a population that has a normal distribution with mean μ and standard deviation σ. Then (because the sampling distribution of $\bar{X}$ is normal with mean μ and standard deviation $\sigma/\sqrt{n}$),

1. *68.26% of all values of $\bar{X}$ are within one standard deviation of the population mean; that is, lie in the interval $(\mu - \sigma/\sqrt{n}, \mu + \sigma/\sqrt{n})$*

2. *95.45% of all values of $\bar{X}$ are within two standard deviations of the population mean; that is, lie in the interval $(\mu - 2\sigma/\sqrt{n}, \mu + 2\sigma/\sqrt{n})$, and*

3. *99.73% of all values of $\bar{X}$ are within three standard deviations of the population mean; that is, lie in the interval $(\mu - 3\sigma/\sqrt{n}, \mu + 3\sigma/\sqrt{n})$.*

If the population from which the samples are selected is not normally distributed, but the sample size is sufficiently large, then the intervals remain the same, but the percentages of $\bar{X}$ that fall within the intervals are approximate percentages.

For the sake of discussion, we'll focus on part 2 of Theorem (3.2). Essentially this statement means that of all possible random samples of size n, 95.45% of them will result in a value of $\bar{x}$ that is between $\mu - 2\sigma/\sqrt{n}$ and $\mu + 2\sigma/\sqrt{n}$. Using this, we could write, for 95.45% of samples,

$$\mu - 2\frac{\sigma}{\sqrt{n}} \leq \bar{x} \leq \mu + 2\frac{\sigma}{\sqrt{n}}.$$

With a little bit of algebra, we can turn this inequality "inside-out" to get for 95.45% of all random samples,

$$\bar{x} - 2\frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{x} + 2\frac{\sigma}{\sqrt{n}}$$

In other words, we can estimate μ with 95.45% confidence using the interval $(\bar{x} \pm 2\sigma/\sqrt{n})$. Moreover, for all random samples of size n, 95.45% of those samples will yield a value of $\bar{x}$ that results in an interval that contains the value of μ the population mean. So $(\bar{x} \pm 2\sigma/\sqrt{n})$ is a 95.45% confidence interval for μ.

The percentage 95.45% is a bit awkward, and so it's typical that the percentage is first specified and then the coefficient (that replaces 2) is found using the sampling distribution of the statistic. Common percentages for confidence intervals are 90%, 95%, and 99%. For each of these, the coefficients are 1.645, 1.96, and 2.575. To find coefficients for specified percentages, using the following procedure. In this procedure, we will use 95% as the example percentage.

1. Convert the percentage into a decimal. For example, 95% is 0.95.

2. Complete the following set of algebraic steps, replacing the 0.95 (and other decimal values) with the appropriate decimal value from Step 1.

$$\begin{aligned} 1 - \alpha &= 0.95 \\ \alpha &= 0.05 \\ \frac{\alpha}{2} &= 0.025 \\ 1 - \frac{\alpha}{2} &= 0.975 \end{aligned}$$

3. Use a computer package to get the coefficient using the quantile of the sampling distribution. In *R*, since the sampling distribution is the normal distribution, this would be

```
> qnorm(0.975)
[1] 1.959964
```

The coefficient obtained is typically denoted with $z_{1-\alpha/2} = z_{0.975} = 1.959964$. As noted in Theorem 3.2, if the population from which the sample is selected isn't normal, then the coefficients are the same, but the percentages become approximate percentages.

Definition 3.9 Confidence Interval, Confidence Coefficient, Confidence Level. *Consider taking a simple random sample of size n from a normally distributed population having mean μ and standard deviation σ. Then the* $(1-\alpha)\times 100\%$ **confidence interval for** μ *is*

$$\left(\bar{x} - z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}, \bar{x} + z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}\right). \tag{2}$$

The value $z_{1-\alpha/2}$ is called the **confidence coefficient** *and $(1-\alpha)$ is called the* **confidence level**.

Figure 5 illustrates the proper interpretation of this statement via a Monte Carlo simulation. The simulation generated 100 samples of size $n = 20$ from a normal distribution. For each of the 100 samples, a 95% confidence interval for the population mean μ was computed using (2). The value of μ is represented by a vertical line, and each of the 100 confidence intervals is represented by a horizontal line. For this run of the simulation, 96% of the samples resulted in a confidence interval that contained μ. The confidence intervals are represented by the dark blue horizontal lines. The red horizontal lines represent the 4% of the simulated confidecne intervals that did not contain μ.

3.4.2 Hypothesis Tests

Hypothesis testing is a procedure that uses the sampling distribution of a statistic to decide between two opposing statements about the value of the population parameter. The null hypothesis, denoted H_0, is typically a statement of "no change," or "no difference," while the alternative hypothesis (denoted H_A) represents the opposite of H_0. From a scientific application, it is usually the alternative hypothesis that is of interest.

For the sake of discussion, consider a new fertilizer for corn. The developer of the fertilizer hopes that it will increase the average yield per acre. Let μ be the average yield per acre using the newly developed fertilizer, and μ_0 = the average yield per acre fertilized with a standard fertilizer. For corn $\mu_0 = 90$ bushels per acre. Then the developer wants to test

$$H_0 : \mu \leq 90 \text{ bushels per acre} \quad \text{versus} \quad H_A : \mu > 90 \text{ bushels per acre}.$$

The null hypothesis includes equality of μ to the "standard" average yield (90). The alternative does not contain a statement of equality. We don't know which of H_0 or H_A is true, but one of them must be. *If* the null hypothesis is true, then the worst case scenario is that $\mu = 90$, and in this case, we also know the sampling distribution of $\bar{X}$; for sufficiently

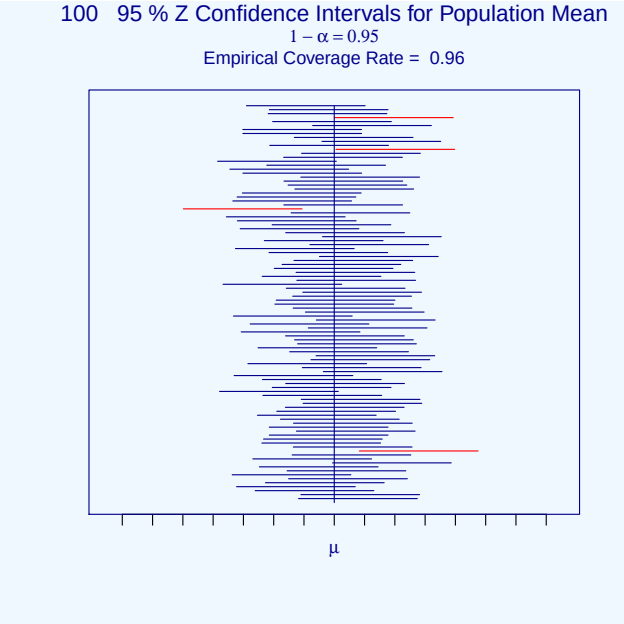

Figure 5: Illustration of 95% confidence: the blue horizontal lines represent confidence intervals which contain the value of μ; red horizontal lines represent confidence intervals which do not contain the value of μ. For this set of simulated 95% confidence interval 96% of them resulted in a confidence interval that captured μ.

large n, the sampling distribution is approximately normal with mean $\mu_{\bar{X}} = 90$ and standard deviation $\sigma/\sqrt{n}$. We can use this knowledge to construct a procedure for deciding between which of H_0 and H_A is most likely to be true using the information in the selected sample. If the null is false, then $\mu_{\bar{X}} > 90$, but we don't know what $\bar{X}$ could possibly be, other than it's greater than 90.

To decide between H_0 and H_A, a sample of size n is selected, and based on that sample, we attempt to determine which of the two hypothesis is most likely to be correct. Just like with confidence intervals, there will be some percentage of samples that result in an erroneous conclusion. In hypothesis testing, there are two possible realities, each with two possible results.

1. Possible reality #1. The null hypothesis H_0 is true.
   (a) Possible result #1. The sample concludes H_0 is false. This is a Type I error. It has probability α. This probability is called the **level** of the test.
   (b) Possible result #2. The sample concludes that H_0 could be true. This is a correct decision, with probability $1 - \alpha$.
2. Possible reality #2. The null hypothesis H_0 is false.

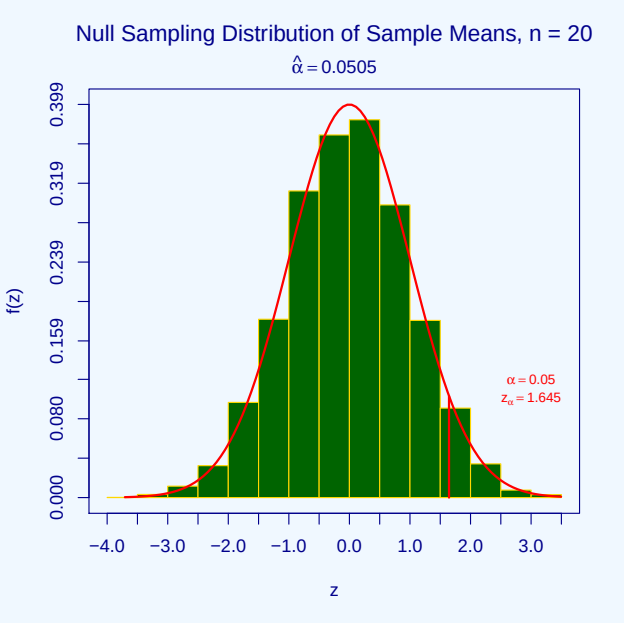

Figure 6: Sampling distribution of Z when H_0 is true. The samples that result in a value of Z greater than 1.645 are those values that would result in a Type I error. For this simulated 10,000 values of Z, 505 were too large.

(a) Possible result #1. The sample concludes H_0 is false. This is a correct decision. It has probability $1 - \beta$, called the **power of the test**.

(b) Possible result #2. The sample concludes that H_0 could be true. This is a Type II error, with probability β.

The procedure for conducting the test works under the presumption that H_0 is true, and uses evidence in the data to determine if this is a reasonable presumption. Keeping this in mind, the procedure follows along these lines. Calculate the **test statistic**

$$Z = \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}}. \tag{3}$$

If the null hypothesis is true, then Z has a standard normal sampling distribution. Compare the value of Z to the quantile of the standard normal distribution that is extreme enough so that α, the probability of a Type I error is small. In other words, *if* H_0 is true, we want to have a small chance of making a Type I error (concluding H_0 is false). Typical values of α are 0.05, 0.10, and 0.01. Since the alternative is that $\mu > 90$, we want to compare Z to large values of the standard normal distribution. For $\alpha = 0.05$, the null hypothesis will be judged as false if the value of $Z > z_{0.95} = 1.645$. If H_0 is false (our presumption is incorrect), then Z will be large anyway, and we'll reject H_0 as we should. Figure 6 illustrates what happens to the many possible values of a test statistic if the null hypothesis is true. Ten-thousand

samples were generated under a true null hypothesis $\mu = 90$. Then for each of the samples, the value of Z was computed, and a histogram of them was constructed. The normal theory curve was superimposed (in red), along with the cut-off value beyond which H_0 is rejected. For this set of 10,000 simulated samples 505 of them resulted in a value of Z that erroneously resulted in rejecting H_0.[2]

4 Principles of Simulation

Simulation is the tool by which statisticians (mathematicians, physicists, chemists, and many other scientists) are able to numerically generate situations which model real-world situations. Those situations are replicated over and over. To each replication a new scientific method is applied, and its performance is evaluated and, perhaps, compared to existing methods applied to the same set of replications. This type of numerical investigation is particularly helpful when theoretical results are difficult to obtain, or when certain properties (like the value of the sample size in the Central Limit Theorem) need to be quantified. There are many good and thorough texts on computational statistics and simulation. Two such recommended texts are written by James E. Gentle. They are *Computational Statistics* (2009)and *Elements of Computational Statistics* (2002). In the next two subsections, we will discuss two main principles of a well-designed simulation.

4.1 Specification of Important Physical Properties

In a carefully designed simulation, the investigator is able to specify properties that are suspected to affect the outcome of the result. Take for example, the Central Limit Theorem. It says, "If the distribution of the population from which the sample is selected is not normally distributed, then the distribution of all possible values of $\bar{X}$ may be well-approximated using a normal distribution as long as the sample size n is sufficiently large." In reading this, questions that may arise are

1. The population distribution isn't normal. So how might different population distributions affect how close the normal approximation is to the true sampling distribution of $\bar{X}$?

2. The sample size has to be "sufficiently large." How large is "sufficiently large?"

3. How does the non-normal population distribution and the "sufficiently large" sample size work together to affect how close the normal approximation is to the true sampling distribution of $\bar{X}$.

For each of these situations, it might also be of interest to investigate (1) the numerical mean of the simulated sampling distribution of $\bar{X}$ – it ought to be very close to μ – and the

[2] If the directions of the inequalities in the null and alternative hypotheses were reversed, then instead of values of Z being too large, we would be concerned with values of Z being too small, specifically for $\alpha = 0.05$, being less than -1.645.

standard deviation of the simulated sampling distribution of $\bar{X}$ – which should be almost equal to $\sigma/\sqrt{n}$. Knowing these two properties gives us a way to help verify that the computer code that has been written is working properly.

4.2 Ability to Replicate

Another property of a carefully designed simulation is that the user should have the ability to repeat exactly the numbers that are "randomly" generated. In statistics, when investigating the effect of sample size, this ability is especially important. The principle behind investigating how sample size affects results is to think along the lines of, "If I had data of a certain sample size, and I add more data to the sample, how much better does the method perform?"

Almost every random number generation algorithm uses a quantity called a "seed" to initialize the process of generating "random numbers." In some way, the seed is transformed, and then an appropriate function applied to the transformed seed so that the overall pattern of the transformed numbers matches the population distribution. If the seed is set from one replication to another, then the values generated can be replicated exactly. It is typical to use a large integer, that must be positive, to set the seed.[3]. To see the result of setting a seed, try to following *R* code.

```
set.seed(12345)
rnorm(5)
```

Take note of the result. Now run the same code again, but replace `rnorm(5)` with `rnorm(10)`; that is, issue the commands

```
set.seed(12345)
rnorm(10)
```

Notice that the first five numerical values from the second call to `rnorm` are exactly the same as the five numbers from the first call.

5 Monte Carlo Simulation

Monte Carlo simulation is a simple, but effective simulation procedure that is especially useful in statistical studies.[4] For illustration, we will continue using the sampling distribution of $\bar{X}$. Recall in our discussion of sampling distributions we relied heavily on the idea of "repeated sampling." In Example 3.1, all samples of size $n = 3$ were generated using a computer – that is, we sampled repeatedly from the (small) population until we had enumerated all possible 120 samples of size $n = 3$.

[3]For specific random number generators available in *R*, at the *R* prompt, type `help(set.seed)`. For more on this important topic, see the second edition of James E. Gentle's text *Random Number Generation and Monte Carlo Methods* (2003)

[4]For a thorough exposition of the principles of Monte Carlo simulation, see Gentle (2003).

In "real life" situations, with huge populations, and with samples much larger than 3 individuals, the number of samples is vast. So being able to enumerate all possible samples of size n from a realistic population is not something that can be done. Monte Carlo simulation is a procedure that enables us to actually perform repeated sampling using a computer to generate many possible samples of size n from the same probability distribution. Using these many samples, the probabilistic properties of a statistic can be empirically investigated. It is important that the user be able to determine specified conditions on the population from which the samples will be generated.

For this paragraph, pretend we do *not* know that the sampling distribution of $\bar{X}$ is normal, but we do know that the mean of $\bar{X}$ is $\mu_{\bar{X}} = \mu$ and its variance $\sigma^2_{\bar{X}} = \sigma^2/n$. We will run a Monte Carlo simulation to see what we can learn about the shape of the sampling distribution. The population from which the samples will be generated have the specified conditions that it has a normal distribution with mean $\mu = 3$ and variance $\sigma^2 = 0.25$.

The R code below is a simple example of how the Monte Carlo simulation might be accomplished. Take time to type these into R yourself, taking note of the graphical and numerical results you obtain from the last four lines of code.

```
set.seed(12345)
n <- 25
x.samples <- matrix(rnorm(1000*n, mean = 3, sd = 0.5),
                    nrow = 1000, ncol = n)
sample.means <- rowMeans(x.samples)
hist(sample.means)
mean(sample.means)
var(sample.means)*(length(sample.means)-1)/length(sample.means)
```

The code above performs the following operations. The third complete line of code, beginning `x.samples <-` requires two lines on paper.

1. Set the seed.
2. Set the sample size to 25.
3. Generate 1000 samples, each of size $n = 25$, from a normal distribution with mean $\mu = 3$ and standard deviation $\sigma = 0.5$. Store the samples in 1000 rows of a matrix called `x.samples`.
4. Compute the sample means ($\bar{x}$) of each of the 1000 samples and store the sample means in the vector called `sample.means`.
5. Draw a histogram of the 1000 sample means.
6. Compute the mean of the 1000 sample means.
7. Compute the variance of the 1000 sample means.

This simple set of seven lines is a Monte Carlo simulation that allows empirical investigation the following properties of the sample mean when the population from which the sample is being drawn is a normal distribution with mean $\mu = 3$ and variance $\sigma^2 = 0.25$.

1. The line `hist(sample.means)` allows us to investigate the *the shape of the sampling distribution of sample means.* The shape of the histogram should be symmetric and bell-shaped. The center of the histogram on the horizontal axis should be very close to 3 (the value of μ), and the lower and upper extremes of the histogram (on the horizontal axis) should be very close to 2.7 and 3.3 ($\mu_{\bar{X}} - 3\sigma_{|barX}$ and $\mu_{\bar{X}} + 3\sigma_{\bar{X}}$ – from the Empirical Rule applied to the sampling distribution of $\bar{X}$).

2. The line `mean(sample.means)` allows us to numerically investigate the *mean of the sampling distribution of the sample means.* The value you observe from this command should be very close to $\mu_{\bar{X}} = \mu = 3$.

3. The line `var(sample.means)*(length(sample.means)-1)/length(sample.means)` allows us to numerically investigate the *variance of the sampling distribution of the sample means.* The value you observe from issuing this command should be very close to $\sigma^2_{\bar{X}} = \sigma^2/n = 0.25/25 = 0.01$

Notice that each of these three empirical results are what we expected based on the discussion of the sampling distribution of the sample mean in Section 3.

As another illustration of Monte Carlo simulation, we will apply these same ideas to a more realistic situation. The simulation will be designed for investigating how large the sample size n needs to be for the sampling distribution of $\bar{X}$ to be well approximated by a normal distribution; in other words, we will use Monte Carlo simulation to investigate properties of the Central Limit Theorem. The function `CLT.sim` was written for this purpose. Documentation for using the function is supplied in Appendix A. The call to the function is

```
CLT.sim(n, dist, params, reps, seed)
```

The arguments are defined as follows.

`n` a positive integer scalar containing the size of the samples. The default value is `n = 30`.

`dist` a character variable indicating the distribution from which to generate samples. Possible values are `"normal"`, `"uniform"`, and `"gamma"`. The default is `dist = "normal"`.

`params` a real vector of length two containing the values of the parameters for the distribution specified in `dist`. The default value is `params = c(0, 1)`, in accordance with the standard normal distribution (mean of zero, and standard deviation one). If `dist = "uniform"`, the default value remains `c(0, 1)`, corresponding to the lower and upper bounds of a uniform(0, 1) distribution. If `dist = "gamma"`, the default value will be changed (within the program) to `c(1, 1)`, corresponding to a gamma distribution with shape parameter = 1 and scale parameter = 1 (an exponential(1) distribution).

`reps` a positive integer scalar indicating the number of samples of size `n` from distribution specified in `dist`. The default value is `reps = 10000`.

`seed` an single integer specifying the seed for the random number generator. If `seed = 0`, then no seed is set.

A complete listing of the function is provided in Appendix B. If you read through the listing, you will see that it is more detailed than the six lines in the previous example. Notice that the function follows the two principles of Sections 4.1 and 4.2. The two main physical properties – population distribution and sample size – can be specified by the user. The user has the ability to replicate the numbers exactly through setting a seed. Type these lines into *R* one at a time and for each call, taking careful note of the numerical and graphical information returned by the function.

```
CLT.sim(seed = 12345)
CLT.sim(n = 10, seed = 12345)
CLT.sim(n = 5, dist = "uniform", params = c(-1,1), seed = 43234)
CLT.sim(n = 15, dist = "uniform", params = c(-1,1), seed = 43234)
CLT.sim(n = 100, dist = "gamma", reps = 25000)
```

For each of these calls, there is similar information returned so that the behaviors of interest can be compared under a variety of circumstances. In particular, for every call to `CLT.sim`, a graphics window opens that contains two figures. The figure on the left is a graph of the density function of the population from which samples are generated as specified by the arguments `dist` and `params`. The figure on the left contains a histogram of the `reps` sample means from samples of size `n` generated from the specified population. The histogram is a representation of the true sampling distribution of $\bar{X}$. The left graph also contains two curves superimposed on the histogram. The dark blue curve is the kernel density estimate of the sampling distribution of $\bar{X}$. The red curve is the normal normal distribution curve used to approximate the true sampling distribution of the sample means. Finally, the function returns a list containing the following real, scalar numerical results.

`pop.mean` the mean μ of the population from which the samples were generated. For

`dist = "normal"`, $\mu =$ `params[1]`,

`dist = "gamma"`, $\mu =$ `params[1]*params[2]`, and

`dist = "uniform"`, $\mu =$ `(params[1] + params[2])/2`.

`pop.stdev` the standard deviation σ of the population from which the samples were generated. For

`dist = "normal"`, $\sigma =$ `params[2]`,

`dist = "gamma"`, $\sigma =$ `sqrt(params[1])*params[2]`, and

`dist = "uniform"`, $\sigma =$ `(params[2] - params[1])/sqrt(12)`.

`sterr.xbar` the standard deviation of the sampling distribution of $\bar{X}$, computed using $\sigma/\sqrt{n}$, where σ is defined as described in `pop.stdev`.

`xbar.mean` the mean of the `reps` sample means.

`stdev.xbar` the standard deviation of the `reps` sample means.

6 The Bootstrap

The bootstrap is a simulation technique that is useful for making certain kinds of statistical inference. We will present a few of those here. For more information, the reader is referred to the books by Efron and Tibshirani (1993) or Gentle (2009)[5]. Bootstrap is a type of *resampling* that involves the use of many samples, each takes from the single sample that was taken from the population of interest. The basic idea in bootstrap resampling is that, because the observed sample contains all the valuable information about the underlying population, the observed sample can be considered *to be* the population. Consequently the distribution of any relevant statistic can be simulated by using random samples from the "population" (the original sample).

The basic bootstrap method formulated by Efron (1979) uses the distribution represented by the sample to study the unknown distribution from which the sample came. The basic tool is the **empirical distribution function** (of the sample) which is used as a model of the underlying population of interest. This is in direct contrast to the Monte Carlo methods that we discussed which generate samples from a known underlying population distribution.

Since bootstrap procedures consider the observed sample as a suitable proxy for population, it must also include an algorithm for estimating population parameters. That algorithm is called the "plug-in principle." The functional of the population distribution that defines the parameter can usually be expressed as some function of the population probability distribution function. The **plug-in estimator** is the same functional of the empirical distribution function. Sometimes finding the correct plug-in estimator can be quite straight-forward, and at other times not. To see the mathematical details, the reader is referred to Efron and Tibshirani (1993) or Gentle (2009). Various properties of the plug-in estimator can be estimated by the use of "bootstrap samples," each of the form $\{x_1^*, \ldots, x_n^*\}$, where the x_i^*'s are chosen from the original sample x_i, with replacement.

Corresponding to each bootstrap sample is a **resampling vector** p^* that is a sequence of proportions of elements in the original sample that are given in the bootstrap sample. For example, consider a sample of elements x_1, x_2, x_3, x_4. Each of these four sample points has attached to it an empirical probability of $1/n = 1/4$. Now, if a bootstrap sample is selected and has elements $x_1^*, x_1^*, x_3^*, x_4^*$, the resampling vector p^* for this bootstrap sample is

[5]This section relies heavily on two resources. The first is the classic text *An Introduction to the Bootstrap*, published in 1993, co-authored by Bradley Efron and Robert J. Tibshirani; the second is *Computational Statistics* by Gentle (2009). Using these as our primary references, we paraphrase only the most basic concepts of bootstrap sampling.

$p^* = \{1/2, 0, 1/4, 1/4\}$. The bootstrap replication of the plug-in estimator will be a function of p^*. The resampling vector can be used to estimate the variance of the bootstrap estimator.

The **bootstrap principle** involves repeating the process that leads from a population distribution function to an empirical distribution function. Taking the empirical distribution function, call it P_n, to the be distribution function of the population, and resampling, we have an empirical distribution function for the new sample, call it $P_n^{(1)}$. The difference is we know more about $P_n^{(1)}$ than we do about P_n. Our knowledge of $P_n^{(1)}$ comes from the simple distribution of $x_1, \ldots, x_n$ (each having empirical probability $1/n$); whereas our knowledge about P_n depends upon the assumption of knowledge about the underlying population distribution.

Suppose the functional of the population distribution that we wish to estimate is μ. Then (mathematically) it can be shown that the plug-in estimator is $\bar{x}$. In the remaining sections, we will explain the algorithms for using bootstrap resampling to estimate the bias and variance of the plug-in estimator, how to compute a bootstrap confidence interval, or conduct a hypothesis test using bootstrap resampling.

6.1 Bootstrap Bias Correction

Recall that an estimator is unbiased if the mean of all possible values of the estimator is the parameter it is intended to estimate. (See Definition 3.6.) Continuing with the idea of $\bar{x}$ being the plug-in estimator for μ, suppose we were interested in estimating μ^2. A natural candidate to estimate μ^2 would be $\bar{x}^2$. The question is, "Is the mean of all possible values of $\bar{x}^2$ equal to μ^2?" Although it may seem obvious that the answer is yes, it is really not quite so simple. Under the right circumstances, the answer can be obtained mathematically. But if it cannot, bootstrap resampling can be used to investigate the presence of bias.

The computational procedure explained below for bias correction is often called the "nonparametric Monte Carlo bootstrap" procedure since it follows the Monte Carlo algorithm of repeated sampling. However, the repeated samples are drawn from the original sample without replacement which is bootstrap. It is nonparametric because there are no distributional assumptions. The procedure is as follows. In this, we let $T = \bar{x}^2$.

- Take B random samples each of size n *with replacement* from the given set of data (the original sample $x_1, \ldots, x_n$);
- For each sample, compute the plug-in estimate T^{*j} of the same functional form as the original estimate T.
- Compute the mean of the T^{*j}, $j = 1, 2, \ldots, B$, call it $\bar{T}^*$.

The distribution of T^{*j} is related to the distribution of T. The bias of T as an estimate of μ^2 can be assessed by the mean of $T^{*j} - T$.

The lines of code below give an example of how this is accomplished in R. The base version of R includes a number of built-in data sets. We will use the data set call `USArrests`. It contains statistics, per 100,000 residents for assault, murder and rape in each of the 50

U.S. States in 1973. Also given is the percentage of the population living in urban areas. The first lines of the code below load that data set and provide easy access to the data columns in that data set. We will work the data in the variable `Murder`.

```
data(USArrests)
attach(USArrests)
names(USArrests)
T <- mean(Murder)^2
B <- 1000
n <- length(Murder)
xstar <- matrix(0, nrow = B, ncol = n)
for(j in 1:B) xstar[j, 1:n] <- sample(Murder, replace = TRUE)
Tstar <- apply(xstar, 1, mean)^2
bias <- mean(Tstar - T)
bias
```

6.2 Bootstrap Estimation of Variance

Suppose we have a sample $x_1, \ldots, x_n$ from a population with distribution P. We have an estimator T (like $\bar{x}^2$) of some parameter θ (like μ^2). The distribution of the estimator is P_n. The bootstrap estimate of some function of T is a plug-in estimate that uses the empirical distribution P_n in place of P. This is the bootstrap principle, and the bootstrap estimate is called the **ideal bootstrap**. The estimator T^* is the same function as T, but computed on a bootstrap sample. T^* is called a **bootstrap observation** of T.

To estimate the variance of T as an estimator of some parameter, we can find the variance of the T^*. The bootstrap estimate of the variance is the sample variance of the T^* based on the B bootstrap samples of size n.

$$\hat{V}(T) = \hat{V}(T^*) = \frac{1}{B-1} \sum_{j=1}^{B} \left(T^{*j} - \bar{T}^*\right)^2,$$

where T^{*j} is the j-th bootstrap observation of T. This can be completed via the same type of nonparametric Monte Carlo bootstrap methods as with bias correction.

Continuing with our previous example, to estimate the variance of $\bar{x}^2$ as an estimator of μ^2, the code would be

```
var.est <- var(Tstar)
var.est
```

6.3 Bootstrap Confidence Intervals

Recall from Section 3.4.1, a confidence interval for the population mean is given by $(\bar{x} \pm z_{1-\alpha/2}\sigma/\sqrt{n})$. The confidence coefficient $z_{1-\alpha/2}$ was obtained from the sampling distribution of $\bar{X}$. We now consider how bootstrap resampling might be used to construct a confidence

interval for any parameter, θ, say. There are a variety of methods for using bootstrap resampling for computing confidence intervals. There are many nuances to be considered, entirely too many to be included in this brief exposition. There are also many variations on the theme described below. For details, the reader is again pointed to the previously mentioned two references.

Given a random sample $x_1, \ldots, x_n$ from an unknown distribution, we want an interval estimate of some parameter, call it θ, that is a functional of the population distribution (like $\theta = \mu^2$). A bootstrap estimator for θ is T^*, based upon the bootstrap sample $x_1^*, \ldots, x_n^*$. To get the bootstrap confidence coefficients, we use the bootstrap observations $T*$ to obtain the distribution of T^*. Then the upper p-th confidence limit using the distribution of T^* is the value $t^*_{(p)}$ such that $p\%$ of the T^* are less than or equal to $t^*_{(p)}$. Therefore, the $(1-\alpha)\times 100\%$ bootstrap confidence interval is

$$\left(t^*_{(\alpha/2)}, t^*_{(1-\alpha/2)}\right).$$

This confidence interval based on the ideal bootstrap and is the "probability-symmetric" bootstrap percentile confidence interval. There are occasions when it is improper to use this confidence interval. Recognize that $t^*_{(p)}$ is simply the p-th order statistic of the B bootstrap observations.

To compute the bootstrap percentile confidence interval for μ^2, we continue with the previous example. However, note that we take additional bootstrap samples, increasing the number of bootstrap replications from 1,000 to 10,000. This is necessary if we wish to more precisely estimate percentiles via the ideal bootstrap.

```
T <- mean(Murder)^2
B <- 10000
n <- length(Murder)
xstar <- matrix(0, nrow = B, ncol = n)
for(j in 1:B) xstar[j, 1:n] <- sample(Murder, replace = TRUE)
Tstar <- apply(xstar, 1, mean)^2
quantile(Tstar, c(0.025, 0.975))
```


6.4 Hypothesis Testing with the Bootstrap

Along the same lines as bootstrap confidence intervals, we will use the quantiles of the empirical distribution of the bootstrap replications to decide whether to reject or fail to reject a null hypothesis. As with confidence intervals, there are many variations on the them presented here, and so for details, the user is referred to Efron and Tibshirani (1993).

Given a random sample $x_1, \ldots, x_n$ from an unknown distribution, we want to make a decision about the hypotheses

$$H_0 : \theta \leq \theta_0 \quad \text{versus} \quad H_A : \theta > \theta_0.$$

T is the estimator of θ; T^* is the bootstrap observation of T. To make a decision about H_0 using bootstrap resampling, use the bootstrap observations $T*$ to obtain the distribution

of T^*. Obtain the percentage of $T^* > \theta_0$ to get the **achieved significance level**. If the achieved significance level is less than α, the null hypothesis is rejected (since that is indicative of the achieved Type I error probability being less than the probability α). This method is based on the ideal bootstrap.[6]

R code for conducting such a test, continuing the previous confidence interval example is below. Let μ represent the the murder rate in the U.S. in 1973. In 2011, the average murder rate was reported to be 6.7 per 100,000. Is there evidence that the murder rate was falling? (Note that we've removed the square from μ.) Then we want to test the hypotheses

$$H_0 : \mu \geq 6.7 \quad \text{versus} \quad H_A : \mu < 6.7,$$

This can be accomplished via a bootstrap simulation. The code is below.

```
B <- 10000
n <- length(Murder)
xstar <- matrix(0, nrow = B, ncol = n)
for(j in 1:B) xstar[j, 1:n] <- sample(Murder, replace = TRUE)
Tstar <- apply(xstar, 1, mean)
ASL <- length(which(Tstar < 6.7))/B
ASL
```

`ASL` is 0.0369. Therefore we reject H_0 and conclude that there is sufficient evidence to conclude the average murder rate in the U.S. is declining. Notice that the direction of the inequality in the sixth line of the code matches the direction of the inequality in the alternative hypothesis.

6.5 Bootstrap in *R*

There are a number of libraries for *R* that are specifically designed for bootstrap resampling, estimation, and inference. One such package is `mosaic`, another is `boot`, and yet another is `tsboot` for time series data. Many other libraries in *R* have built-in bootstrap functions. If the task is simple, the methods presented here are sufficient. However, if they become more complicated as described in Section 6.6, then it may be preferable to use the tools provided.

6.6 Final Notes

In this brief exposition on bootstrap resampling, we considered only univariate data from a random sample (and so independent, identically distributed observations). When the data is dependent – like in time series – or has more than one variable, where the two variables are related – like in a problem on regression – then care must be taken in bootstrap resampling to not destroy the structure in the data. We also limited the discussion to only a single-valued parameter. When there is more than one parameter to be estimated, the

[6] If the hypotheses are $H_0 : \theta \geq \theta_0$ versus $H_A : \theta < \theta_0$, then the achieved significance level is the percentage of $T^* < \theta_0$.

interaction between parameters should be taken into consideration, among other things. In short, bootstrap estimation is a very large subject, and requires more study and investigation into the details for each application than can possibly be discussed in this document.

7 Exercises

1. Write an R function that illustrates, via Monte Carlo simulation, the Empirical Rule for $\bar{X}$ (Theorem 3.2).

2. Use bootstrap resampling to illustrate the Empirical Rule for $\bar{X}$ (Theorem 3.2).

3. Write an R function for investigating the behavior of confidence intervals for a population mean that uses Monte Carlo simulation. The function should allow the user to specify (1) the population distribution and its parameters (2) the sample size, (3) the confidence level, and (4) the number of replications. The numerical results of the function should include at least (a) the percentage of simulated confidence intervals that contain the population mean (b) the population mean, and (c) the population standard deviation. The function should also create meaningful graph that shows (graphically, and perhaps with text), individual confidence intervals and their relationship to the population mean. *Note:* Consider the implications of the number of replications on the appearance of the graph.

4. Write an R function for investigating the behavior of test statistics for testing $H_0 : \mu = \mu_0$ versus $H_A : mu < \mu_0$ using Monte Carlo simulation. The function should allow the user to specify (1) the population distribution and its parameters, (2) the sample size, (3) the hypothesized value μ_0, (4) the significance level, and (5) the number of replications. The numerical results of the function should include at least (a) the percentage of simulated confidence intervals that result in a Type I error (b) the population mean, and (c) the population standard deviation. The function should also create a graph that contains a histogram of the simulated values of the test statistic, with the normal theory curve superimposed. Incorporated into the graph should be some meaningful graphical (and perhaps textual) information that shows the test statistics that resulted in a Type I error.

5. Write an R function for investigating the behavior of confidence intervals for a population mean that uses bootstrap resampling. The function should allow the user to specify (1) the population distribution, (2) the sample size, and (3) the confidence level. The results of the function should include at least the percentage of simulated confidence intervals that contain the population mean and a meaningful graph (in the context of confidence intervals) that shows individual confidence intervals and their relationship to the population mean.

6. Write an R function for investigating the behavior of test statistics for testing $H_0 : \mu = \mu_0$ versus $H_A : mu < \mu_0$ using bootstrap resampling. The function should allow the

user to specify (1) the population distribution and its parameters, (2) the sample size, (3) the hypothesized value μ_0, (4) the significance level, and (5) the number of replications. The numerical results of the function should include at least (a) the percentage of simulated confidence intervals that result in a Type I error (b) the population mean, and (c) the population standard deviation. The function should also create a graph that contains a histogram of the simulated values of the test statistic, with the normal theory curve superimposed. Incorporated into the graph should be some meaningful graphical (and perhaps textual) information that shows the test statistics that resulted in a Type I error.

7. Write an R function for investigating the behavior of test statistics for testing $H_0 : \mu = \mu_0$ versus $H_A : \mu \neq \mu_0$ using bootstrap resampling. The function should allow the user to specify (1) the population distribution and its parameters, (2) the sample size, (3) the hypothesized value μ_0, (4) the significance level, and (5) the number of replications. The numerical results of the function should include at least (a) the percentage of simulated confidence intervals that result in a Type I error (b) the population mean, and (c) the population standard deviation. The function should also create a graph that contains a histogram of the simulated values of the test statistic, with the normal theory curve superimposed. Incorporated into the graph should be some meaningful graphical (and perhaps textual) information that shows the test statistics that resulted in a Type I error.

8. The discussion in this monograph dealt exclusively with distributional properties of $\bar{X}$. Many of the same general, logical arguments can be made about the sampling distribution of the sample variance S^2, as given in Definition 3.3, although the details change. In particular

   Theorem 7.1 *If $X_1, \ldots, X_n$ is a random sample of size n from a normally distributed population with mean μ and standard deviation σ, then the sampling distribution of $\chi^2 = (n-1)S^2/\sigma^2$ is a $\chi^2(\nu)$ distribution, where $\nu = n - 1$ are called "degrees of freedom."*

   If the population distribution is not normal, but the sample size is sufficiently large, then the $\chi^2(\nu)$ distribution still acts well as an approximation to the sampling distribution of χ^2.

   If the sample size is extremely large, then the sampling distribution of $Z = (\chi^2 - \nu)/\sqrt{2\nu}$ can be approximated by a normal distribution with mean zero and standard deviation one.

   Write a function similar to the function `CLT.sim` that uses Monte Carlo simulation to investigate the properties of the sampling distribution of χ^2.

9. Repeat Exercise 3 for $(1-\alpha)100\%$ confidence intervals for σ^2, given by

$$\left(\frac{(n-1)s^2}{\chi^2_{1-\alpha/2}(n-1)}, \frac{(n-1)s^2}{\chi^2_{\alpha/2}(n-1)} \right),$$

where $\chi^2_p(\nu)$ is the p-th quantile of a $\chi^2(\nu)$ distribution. Use Theorem 7.1 to help decide upon a complete set of properties that need to be included in the investigation.

10. Repeat Exercise 9 for $(1-\alpha)100\%$ confidence intervals for σ^2 using bootstrap resampling instead of Monte Carlo simulation.

11. The test statistic for testing $H_0 : \sigma^2 = \sigma_0^2$ versus $H_A : \sigma^2 > \sigma_0^2$ is

$$\chi^2 = \frac{(n-1)s^2}{\sigma_0^2} \tag{4}$$

which has a $\chi^2(n-1)$ distribution if H_0 is true. Write a function that uses Monte Carlo simulation to investigate the behavior of the test statistic when H_0 is true. Use Theorem 7.1 to help decide upon a complete set of properties that need to be included in the investigation.

12. Repeat Exercise 11 using bootstrap resampling instead of Monte Carlo simulation.

13. Use a nonparametric Monte Carlo bootstrap algorithm to estimate the bias and variance of S and an estimator of σ.

8 References

Arons, Abigail, (2011). "Childhood Obesity in Texas: The Costs, the Policies, and a Framework for the Future." Published by Children's Hospital Association of Texas: Austin, TX. `http://www.childhealthtx.org/pdfs/`.

Efron, Bradley, (1979). "Bootstrap methods: Another look at the jackknife," *Annals of Statistics*, **7**, pp. 1-26.

Efron, Bradley and Tibshirani, Robert J., (1993) *An Introduction to the Bootstrap*. Published by Chapman & Hall: New York.

Gentle, James E., (2009) *Computational Statistics*. Published by Springer: New York.

Gentle, James E., (2003) *Random Number Generation and Monte Carlo Methods*, Second edition. Published by Springer: New York.

Gentle, James E., (2002) *Elements of Computational Statistics*. Published by Springer: New York.

A Documentation for `CLT.sim` Function

`CLT.sim` Function

Description

This function investigates the behavior of the sampling distribution of the sample mean via Monte Carlo simulation.

Usage

```
CLT(n = 30, dist = "normal", params = c(0, 1), reps = 10000, seed = 0)
```

Arguments

`n` a positive integer scalar containing the size of the samples. The default value is `n = 30`.

`dist` a character variable indicating the distribution from which to generate samples. Possible values are `"normal"`, `"uniform"`, and `"gamma"`. The default is `dist = "normal"`.

`params` a real vector of length two containing the values of the parameters for the distribution specified in `dist`. The default value is `params = c(0, 1)`, in accordance with the standard normal distribution (mean of zero, and standard deviation one). If `dist = "uniform"`, the default value remains `c(0, 1)`, corresponding to the lower and upper bounds of a uniform(0, 1) distribution. If `dist = "gamma"`, the default value will be changed (within the program) to `c(1, 1)`, corresponding to a gamma distribution with shape parameter = 1 and scale parameter = 1 (an exponential(1) distribution).

`reps` a positive integer scalar indicating the number of samples of size `n` from distribution specified in `dist`. The default value is `reps = 10000`.

`seed` an single integer specifying the seed for the random number generator. If `seed = 0`, then no seed is set.

Details

If improper values are supplied for any of the arguments, then the function will return an error message explaining why the value is improper, and the function will stop.
The function sets graphical parameters via `par(mfrow = c(1, 2), bg = "aliceblue", col = "darkblue", col.sub = "darkblue", col.lab = "darkblue", col.axis = "darkblue")`.

Value

Each call to the function `CLT.sim` creates a graphics window with two figures. The figure on the left is a graph of the density function of the population from which samples are generated as specified by the arguments `dist` and `params`. The figure on the left contains a histogram of the `reps` sample means from samples of size `n` generated from the specified population.

The histogram is a representation of the true sampling distribution of $\bar{X}$. The left graph also contains two curves superimposed on the histogram. The dark blue curve is the kernel density estimate of the sampling distribution of $\bar{X}$. The red curve is the normal normal distribution curve used to approximate the true sampling distribution of the sample means. Finally, the function returns a list containing the following real, scalar numerical results.

`pop.mean` the mean μ of the population from which the samples were generated. For

`dist = "normal"`, $\mu =$ `params[1]`,

`dist = "gamma"`, $\mu =$ `params[1]*params[2]`, and

`dist = "uniform"`, $\mu =$ `(params[1] + params[2])/2`.

`pop.stdev` the standard deviation σ of the population from which the samples were generated. For

`dist = "normal"`, $\sigma =$ `params[2]`,

`dist = "gamma"`, $\sigma =$ `sqrt(params[1])*params[2]`, and

`dist = "uniform"`, $\sigma =$ `(params[2] - params[1])/sqrt(12)`.

`sterr.xbar` the standard deviation of the sampling distribution of $\bar{X}$, computed using $\sigma/\sqrt{n}$, where σ is defined as described in `pop.stdev`.

`xbar.mean` the mean of the `reps` sample means.

`stdev.xbar` the standard deviation of the `reps` sample means.

See Also

`set.seed` for technical explanation of the various random number generator kinds in R and the use of the `set.seed` function.

`rnorm`, `rgamma`, and `runif` for generating pseudo-random variates from a normal, gamma, or uniform distribution, respectively.

`dnorm`, `dgamma`, and `dunif` for computing the probability density function of a normal, gamma, or uniform family, respectively.

`plotmath` and `bquote` for including mathematical symbols within graphics.

`apply` for applying functions to margins of an array or matrix.

`hist` for creating a histogram, or saving its elements for plotting later.

`density` for computing a kernel density estimate of the probability density function from a data set.

`par` to return graphical parameters to their defaults.

Examples

```
CLT.sim()
CLT.sim(n = 15, dist = "uniform", params = c(-1,1))
CLT.sim(n = 10)
CLT.sim(n = 100, dist = "gamma", reps = 25000)
```

B Listing of `CLT.sim` Function

```
CLT.sim <- function(n = 30, dist = "normal", params = c(0, 1),
                    reps = 10000, seed = 0) {
#-------------------------------------------------------------------------
#  R function to investigate the behavior of the sampling distribution
#  of the sample mean via Monte Carlo simulation.
#
#
#  ARGUMENTS:
#    n = a positive integer scalar containing the size of the samples.
#        The default value is n = 30.
#    dist = a character variable indicating the distribution of the
#           population from which the samples are to be generated.
#           Allowed values of dist are "normal", "uniform", and
#           "gamma".  The default is dist = "normal".
#    params = a real vector of length two containing the values of the
#             parameters for the distribution specified in dist.  The
#             default value is params = c(0, 1), in accordance with the
#             standard normal distribution (mean of zero, and standard
#             deviation one).  If dist = "uniform", the default value
#             remains c(0, 1), corresponding to the lower and upper
#             bounds of a uniform(0, 1) distribution.  If dist = "gamma",
#             the default value will be changed (within the program)
#             to c(1, 1), corresponding to a gamma distribution with
#             shape = 1 and scale = 1 (an exponential(1) distribution).
#    reps = a positive integer scalar indicating the number of samples
#           of size n from distribution specified in dist.  The default
#           value is reps = 10000.
#    seed = an single integer specifying the seed for the random number
#           generator.  If seed = 0, then no seed is set.
#
#  DETAILS: Regarding the parameter values for the three distributions,
#           refer to the details in the help files for the respective
#           distributions.
#
#  VALUE: The function CLT.sim opens a graphics window that contains
#         two plots.  The plot on the left is a graph of the shape of
#         the distribution of the population from which the samples are
#         generated.  The plot of the right is of a histogram of the
#         reps values of the sample mean with a kernel density estimate
#         of the true sampling distribution superimposed in darkblue and
#         the normal theory curve superimposed in red.  The function
```

```
#         returns a list containing the following scalar objects:
#         pop.mean = the mean of the population,
#         pop.stdev = the standard deviation of the population,
#         sterr.xbar = the theoretical standard error of the sample
#                      mean (= pop.stdev/sqrt(n)),
#         xbar.mean = the mean of the reps sample means, and
#         stdev.xbar = the standard deviation of the reps sample means.
#
#  SEE ALSO: set.seed, rnorm, runif, rgamma, dnorm, dunif, dgamma,
#            density, hist, apply, plotmath, bquote
#
#  EXAMPLES:
#     CLT.sim()
#     CLT.sim(n = 15, dist = "uniform", params = c(-1,1))
#     CLT.sim(n = 10)
#     CLT.sim(n = 100, dist = "gamma", reps = 25000)
#
#  Written: 07/28/2012 Jane L. Harvill
#-------------------------------------------------------------------------
  eps <- 1e-08
#
#  Check validity of input:
#
#  Sample size, reps, and seed first:
#
  if(abs(n - as.integer(n)) > 0) stop("\n n must be an integer.\n")
  if(n < 1) stop("\n n must be an integer greater than 0.\n")
  if(abs(reps - as.integer(reps)) > 0) stop("\n reps must be an integer.\n")
  if(reps < 1) stop("\n reps must be an integer greater than 0.\n")
  if(seed < 0) stop("\n seed must be a non-negative integer.\n")
  if(abs(seed - as.integer(seed)) > 0) stop("\n seed must be an integer.\n")
  if(seed > eps) set.seed(as.integer(seed))
#
#  Value of dist:
#
  if(dist != "normal" && dist != "uniform" && dist != "gamma")
    stop("\n Please specify an allowable name for dist.\n")
#
#  Check for parameters passed if dist = "gamma".  If defaults are
#  not changed, then change them.
#
  if(dist == "gamma" && abs(params[1]) < eps && abs(params[2] - 1) < eps)
```

```
    params <- c(1, 1)
#
#  Parameter values based on value of dist.
#
  if(dist == "normal") {
    if(params[2] < eps) {
      stop("\n The standard deviation must be positive.\n")
    }
  }
  if(dist == "uniform") {
    if((params[2] - params[1]) < eps) {
      stop("\n The lower bound must exceed the upper bound.\n")
    }
  }
  if(dist == "gamma") {
    if(params[1] < eps || params[2] < eps) {
      stop("\n The parameters of a gamma distribution are positive.\n")
    }
  }
#
#  Compute population parameters and generate samples.
#
  if(dist == "normal") {
    distn <- "Normal"
    mu    <- params[1]
    sigma <- params[2]
    x     <- matrix(rnorm(n = n*reps, mean = mu, sd = sigma),
                        nrow = n, ncol = reps)
    y     <- seq(from = mu - 3.5*sigma, to = mu + 3.5*sigma, length = 101)
    fy    <- dnorm(y, mean = mu, sd = sigma)
    prtxt <- bquote(group("(",list(mu, sigma),")") ==
                    group("(",list(.(mu),.(sigma)),")") )
  }
  if(dist == "uniform") {
    distn <- "Uniform"
    mu    <- (params[1] + params[2])/2
    sigma <- (params[2] - params[1])/sqrt(12)
    x     <-  matrix(runif(n = n*reps, min = params[1], max = params[2]),
                    nrow = n, ncol = reps)
    y     <- c(params[1], params[2])
    fy    <- rep(1/(params[2] - params[1]), times = 2)
    prtxt <- bquote(group("(",list(a, b),")") ==
```

```
                group("(",list(.(params[1]),.(params[2])),")") )
  }
  if(dist == "gamma") {
    distn <- "Gamma"
    mu    <- params[1]*params[2]
    sigma <- sqrt(params[1])*params[2]
    x     <- matrix(rgamma(n = n*reps, shape = params[1], scale =
                 params[2]), nrow = n, ncol = reps)
    y     <- seq(from = eps, to = 8*sigma, length = 101)
    fy    <- dgamma(y, shape = params[1], scale = params[2])
    prtxt <- bquote(group("(",list(alpha, beta),")") ==
                  group("(",list(.(params[1]),.(params[2])),")") )
  }

  xbars  <- apply(x, 2, mean)
  mm     <- mean(xbars)
  sxbar  <- sd(xbars)
  sexbar <- sigma/sqrt(n)
#
#  Set up normal theory curve.
#
  xx     <- seq(from = min(xbars), to = max(xbars), length = 101)
  yy     <- dnorm(xx, mean = mu, sd = sexbar)
#
#  Set up plotting objects.
#
  ht     <- hist(xbars, plot = FALSE)
  to.y   <- seq(from = 0, to = max(yy, ht$density), length = 11)
  par(mfrow = c(1, 2), bg = "aliceblue", col = "darkblue",
      col.sub = "darkblue", col.lab = "darkblue", col.axis = "darkblue")
  del    <- 2*(ht$breaks[2] - ht$breaks[1])
  dens   <- density(xbars)
#
#  Create graph of distribution of population from which samples
#  are generated.
#
  plot(y, fy, xlab = "x", ylab = "f(x)", type = "l", col = "darkblue")
  mtext(paste("Distribution of",distn,"Parent Population"), side = 3,
        line = 2, cex = 1.25, col = "darkblue")
  mtext(prtxt, side = 3, line = 0.54, cex = 1.00, col = "darkblue")
#
#  Create histogram with normal theory curve superimposed.
```

```
#
  plot(ht, freq = FALSE, ylim = c(0, max(yy, ht$density)),
       xlim = c(min(ht$breaks, xx), max(ht$breaks, xx)),
       xlab = expression(bar(x)), ylab = expression(f(bar(x))),
       main = " ", col = "darkgreen", border = "gold", axes = FALSE)
  box(col = "darkblue")
  lines(xx, yy, col = "red", lwd = 2)
  lines(dens, col = "darkblue", lwd = 2)
  axis(side = 1, at = ht$breaks, labels = TRUE, tick = TRUE,
       col = "darkblue", cex = 0.75)
  axis(side = 2, labels = TRUE, tick = TRUE, col = "darkblue", cex = 0.75,
       at = round(to.y, 3))
  mtext(paste("Sampling Distribution of Sample Means, n =",n), side = 3,
        line = 2, cex = 1.25, col = "darkblue")
  mtext(bquote(group("(",list(mu[bar(x)], sigma[bar(x)]),")") ==
               group("(",list(.(mu),.(round(sexbar,4))),")") ),
        side = 3, line = 0.5)
#
#  Return list of values.
#
  return(list(pop.mean = mu, pop.stdev = sigma, sterr.xbar = sexbar,
              xbar.mean = mm, stdev.xbar = sxbar))
}
```

C Samples and Sample Means from Example 3.1

Listing of sample indices for samples of size $n = 3$ from a population of 10 individuals. Measurements corresponding to the indices are provided in (1).

```
      [1,]    1    2    3
      [2,]    1    2    4
      [3,]    1    2    5
      [4,]    1    2    6
      [5,]    1    2    7
      [6,]    1    2    8
      [7,]    1    2    9
      [8,]    1    2   10
      [9,]    1    3    4
     [10,]    1    3    5
     [11,]    1    3    6
     [12,]    1    3    7
     [13,]    1    3    8
     [14,]    1    3    9
     [15,]    1    3   10
     [16,]    1    4    5
     [17,]    1    4    6
     [18,]    1    4    7
     [19,]    1    4    8
     [20,]    1    4    9
     [21,]    1    4   10
     [22,]    1    5    6
     [23,]    1    5    7
     [24,]    1    5    8
     [25,]    1    5    9
     [26,]    1    5   10
     [27,]    1    6    7
     [28,]    1    6    8
     [29,]    1    6    9
     [30,]    1    6   10
     [31,]    1    7    8
     [32,]    1    7    9
     [33,]    1    7   10
     [34,]    1    8    9
     [35,]    1    8   10
     [36,]    1    9   10
     [37,]    2    3    4
     [38,]    2    3    5
```

```
[39,]    2    3    6
[40,]    2    3    7
[41,]    2    3    8
[42,]    2    3    9
[43,]    2    3   10
[44,]    2    4    5
[45,]    2    4    6
[46,]    2    4    7
[47,]    2    4    8
[48,]    2    4    9
[49,]    2    4   10
[50,]    2    5    6
[51,]    2    5    7
[52,]    2    5    8
[53,]    2    5    9
[54,]    2    5   10
[55,]    2    6    7
[56,]    2    6    8
[57,]    2    6    9
[58,]    2    6   10
[59,]    2    7    8
[60,]    2    7    9
[61,]    2    7   10
[62,]    2    8    9
[63,]    2    8   10
[64,]    2    9   10
[65,]    3    4    5
[66,]    3    4    6
[67,]    3    4    7
[68,]    3    4    8
[69,]    3    4    9
[70,]    3    4   10
[71,]    3    5    6
[72,]    3    5    7
[73,]    3    5    8
[74,]    3    5    9
[75,]    3    5   10
[76,]    3    6    7
[77,]    3    6    8
[78,]    3    6    9
[79,]    3    6   10
[80,]    3    7    8
```

```
 [81,]    3    7    9
 [82,]    3    7   10
 [83,]    3    8    9
 [84,]    3    8   10
 [85,]    3    9   10
 [86,]    4    5    6
 [87,]    4    5    7
 [88,]    4    5    8
 [89,]    4    5    9
 [90,]    4    5   10
 [91,]    4    6    7
 [92,]    4    6    8
 [93,]    4    6    9
 [94,]    4    6   10
 [95,]    4    7    8
 [96,]    4    7    9
 [97,]    4    7   10
 [98,]    4    8    9
 [99,]    4    8   10
[100,]    4    9   10
[101,]    5    6    7
[102,]    5    6    8
[103,]    5    6    9
[104,]    5    6   10
[105,]    5    7    8
[106,]    5    7    9
[107,]    5    7   10
[108,]    5    8    9
[109,]    5    8   10
[110,]    5    9   10
[111,]    6    7    8
[112,]    6    7    9
[113,]    6    7   10
[114,]    6    8    9
[115,]    6    8   10
[116,]    6    9   10
[117,]    7    8    9
[118,]    7    8   10
[119,]    7    9   10
[120,]    8    9   10
```

Listing of sample means (reading from left to right) corresponding to the 120 samples listed on the previous pages.

```
18.02454 18.44152 18.93767 17.56661 18.21655 17.72898 18.01321 17.83562
18.03559 18.53174 17.16068 17.81062 17.32305 17.60728 17.42969 18.94871
17.57766 18.22759 17.74002 18.02425 17.84666 18.07381 18.72374 18.23617
18.52040 18.34281 17.35269 16.86511 17.14935 16.97176 17.51505 17.79929
17.62170 17.31171 17.13412 17.41836 18.70571 19.20186 17.83081 18.48074
17.99317 18.27741 18.09982 19.61884 18.24778 18.89772 18.41014 18.69438
18.51679 18.74393 19.39387 18.90629 19.19053 19.01294 18.02281 17.53524
17.81947 17.64188 18.18518 18.46941 18.29182 17.98184 17.80425 18.08848
19.21291 17.84185 18.49179 18.00422 18.28845 18.11086 18.33800 18.98794
18.50037 18.78460 18.60701 17.61688 17.12931 17.41354 17.23595 17.77925
18.06348 17.88589 17.57591 17.39832 17.68255 18.75497 19.40491 18.91734
19.20157 19.02398 18.03386 17.54628 17.83052 17.65293 18.19622 18.48046
18.30287 17.99288 17.81529 18.09953 18.53001 18.04243 18.32667 18.14908
18.69237 18.97661 18.79902 18.48903 18.31144 18.59568 17.32131 17.60555
17.42796 17.11798 16.94039 17.22462 17.76791 17.59032 17.87456 17.38698
```