9

Monte Carlo Simulation

Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.

John von Neumann[1]

The one thing about Monte Carlo is that it never gives an exact answer.

Stanislaw Ulam[2]

9.1. Introduction

The chapters in the first part of this book make clear that regression analysis can be used to describe data. The remainder of this book is dedicated to understanding regression as a tool for drawing inferences about how variables are related to each other. The central idea in inferential statistics is that the data we observe are just one sample from a larger population. The goal of inference is to determine what evidence the sample provides about the relationship between variables in the population.

This chapter explains how we will use the computer to draw random samples to evaluate the performance of a variety of sample-based statistics. We will review basic theory behind random number generation with computers, offer a simple example of Monte Carlo simulation, and introduce a Monte Carlo simulation Excel add-in.

Like regression analysis, Monte Carlo simulation is a general term that has many meanings. The word "simulation" signifies that we build an artificial model of a real system to study and understand the system. The "Monte Carlo" part of the name alludes to the randomness inherent in the analysis:

The name "Monte Carlo" was coined by [physicist Nicholas] Metropolis (inspired by [Stanislaw] Ulam's interest in poker) during the Manhattan Project of World

[1] von Neumann (1951).
[2] Ulam (1991, p. 199).

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

War II, because of the similarity of statistical simulation to games of chance, and because the capital of Monaco was a center for gambling and similar pursuits. Monte Carlo is now used routinely in many diverse fields, from the simulation of complex physical phenomena such as radiation transport in the earth's atmosphere and the simulation of the esoteric subnuclear processes in high energy physics experiments, to the mundane, such as the simulation of a Bingo game or the outcome of Monty Hall's vexing offer to the contestant in "Let's Make a Deal."

(Drakos, 1995)

Monte Carlo simulation is a method of analysis based on artificially recreating a chance process (usually with a computer), running it many times, and directly observing the results.

We will use Monte Carlo simulation to understand the properties of different statistics computed from sample data. In other words, we will test-drive estimators, figuring out how different recipes perform under different circumstances. Our procedure is quite simple: In each case we will set up an artificial environment in which the values of important parameters and the nature of the chance process are specified; then the computer will run the chance process over and over; finally the computer will display the results of the experiment.

The next section explains the fundamental principles behind random number generation, which is the engine that drives a Monte Carlo simulation. Section 9.3 is a practical guide to generating random numbers in Excel. Section 9.4 demonstrates Monte Carlo via a simple example, and the last section introduces an Excel add-in that can be used to run a Monte Carlo simulation in any Excel workbook.

9.2. Random Number Generation Theory

Workbook: RNGTheory.xls

Because Monte Carlo simulation is based on repeatedly sampling from a chance process, it stands to reason that random numbers are a crucial part of the procedure. This section will briefly explain the theoretical principles behind random number generation.

We begin with a simple but important claim: Excel, like all other computer software, cannot draw a true sequence of random numbers. At best, Excel's random draws can mimic the behavior of truly random draws, but true randomness is unattainable. The inability of computer software to generate truly random numbers results from a computer program's having to follow a deterministic algorithm to produce its output. If the previous number and the algorithm are known, so is the next number. Because the essence of randomness is that you do not know what is going to happen next, numbers produced by computer software are not genuinely random. Thus, Monte Carlo simulation

	A	B	C	D	E	F	G
1	**An LCG in action. Set the A, B, and m values to control the LCG**						
2	A	100					
3	B	3					
4	m	5					
5							
6	seed	0.5	NextNumber = MOD(B*PreviousNumber+A,m)				
7					1.5		
8					4.5		
9					3.5		
10					0.5		
11					1.5		
12					4.5		
13					3.5		
14					0.5		

Figure 9.2.1. An LCG demonstration.
Source: [RNGTheory.xls]LCG.

with Excel is based on pseudorandom number generation. Throughout this book, when we say random number, we actually mean pseudorandom number.

The random number recipe used by all versions of Excel before Excel 2003 is called a linear congruential generator (LCG).[3] Starting from an initial value, called the seed, the LCG simply puts a number through a formula

$$\text{NextNumber} = (B \cdot \text{PreviousNumber} + A) \,\text{Mod}\, m,$$

to generate the next number. In the formula above Mod means Modulus. The expression x Mod y yields the remainder when a number x is divided by another number y.

To see the simple logic behind this algorithm, go to the LCG sheet in RNGTheory.xls. Figure 9.2.1 is a picture of a portion of the LCG sheet. Starting from a seed of 0.5 and $A = 100$, $B = 3$, and $m = 5$, the next number is 1.5 (cell E7). The steps in the calculation are (1) $3 \times 0.5 = 1.5$, (2) $1.5 + 100 = 101.5$, (3) 101.5 Mod $5 = 1.5$. The output of the Excel function MOD(x, y) is x Mod y.

The LCG $(3 \cdot \text{PreviousNumber} + 100)$ Mod 5 is an unsatisfactory random number generator (RNG). After all, we will see 1.5 followed by 4.5, 3.5, 0.5 (the first number), and then the numbers simply repeat themselves. One way to judge a random number generator is by its period or the number of values generated before returning to the first value and recycling through the list.

[3] As part of a massive revision of statistical functions, Excel 2003 uses a new algorithm to generate Uniform(0,1) random numbers. Unfortunately, as of this writing, the new algorithm has a problem and can give negative numbers. A patch is available from Microsoft at <office.microsoft.com>. As this section will explain, we recommend using our own built-in random number generator. Execute Help: About Microsoft Excel to see what version of Excel you are using. See the Basic Tools/RandomNumber folder for more information about Excel 2003's random number generator.

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

By changing the parameters, A, B, and m, you change the performance of the generator. For example, set $m = 7$ (in cell B4). The generated sequence of numbers changes and the period lengthens to 6. The period, however, is a simple, and potentially misleading attribute of a random number generator. There are many other desirable attributes in a random number generator, and many different tests have been devised to judge randomness.

We are now ready to examine Excel's random number function, RAND as implemented in versions prior to Excel 2003. Click the ShowRAND button to see the Excel LCG. For the LCG used by RAND, Microsoft programmers chose $A = 0.211327$, $B = 9821$, and $m = 1$. The numbers generated are always between 0 and 1. Excel's RAND function simulates a uniform distribution on the interval from 0 to 1 (known as the Uniform(0,1) distribution). The idea is that we are drawing random numbers from the interval 0 to 1 with every number equally likely to be chosen.[4] This is not as limiting as you might think. For example, we can obtain numbers uniformly distributed between 0 and 10 by multiplying the original numbers by 10. In addition, we can add 50 to make them range from 50 to 60. In fact, starting from numbers drawn from the Uniform(0,1) distribution, it is possible to generate numbers that are random draws from almost any desired statistical distribution.

You will not see a repetition in the 15 numbers generated in column E – the Excel RAND function has an extremely long period.[5] However, it is shown below that, its long period notwithstanding, Excel's RAND is not a good random number generator. Visual Basic, the programming language behind Excel, has its own LCG random number algorithm called Rnd. It is preferable to Excel's RAND. Rnd uses $B = 1{,}140{,}671{,}485$, $A = 12{,}820{,}163$, and $m = 2^{24}$. Its period is 16,777,216 (2^{24}), but, like Excel's RAND, it is still a fairly crude RNG.

The problem with both RAND and Rnd is that the sequences of numbers they produce have too much structure, meaning they are not "random enough" when seen from certain perspectives. Figure 9.2.2 offers a simple example of the undesirable structure embedded in RAND and Rnd. The three graphs in Figure 9.2.2 were created by trapping the next number in the sequence whenever the previous number fell between 0.7 and 0.7001. Each graph has 1,000 data points. Excel's RAND function graph is the least random of the three. Whenever a number between 0.7 and 0.7001 is visited, the next number is guaranteed to lie on one of the two lines in the graph. That is not very random. Visual Basic's Rnd fills the graph, but the points are still too

[4] Well, not every number. Because it uses binary arithmetic and has finite memory, Excel can only recognize 2^{54} points on the number line between 0 and 1.

[5] In fact, because of complicated floating-point precision issues, Excel's RAND function does not exactly repeat itself after 1,000,000, the period for the LCG $9821*x_{-1} + 0.211327$. For more on this issue, see the information in the BasicTools/RandomNumber folder.

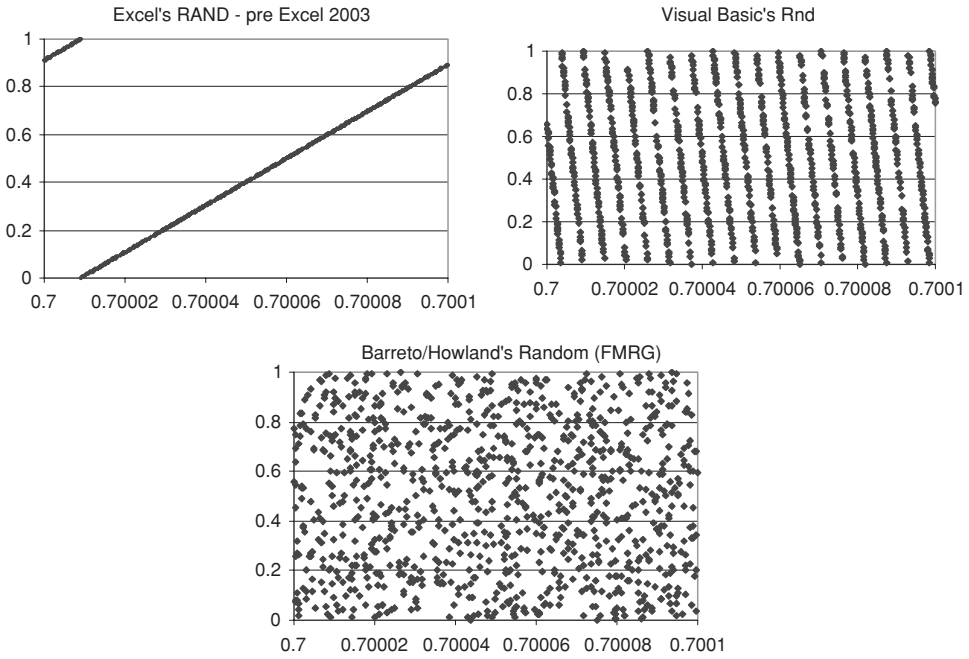

Figure 9.2.2. Comparing three random number generators.
Source: [RNGTheory.xls]Graphs.

systematic – they fall on straight lines. The bottom graph looks like the best of the three: there does not appear to be a systematic relationship between successive numbers in the sequence.

The bottom graph in Figure 9.2.2 is derived from an implementation of a random number algorithm based on a fast multiple recursive generator (FMRG) (Deng and Lin 2000). Multiple recursive generators are like LCGs in that they use the previous output to generate the next number, but instead of using just the previous number like an LCG, an MRG uses a linear combination of the past k random numbers generated.

$$x_i = (a_1 x_{i-1} + \ldots + a_k x_{i-k}) \bmod m$$

The formula above says that the ith number in the sequence is a linear combination of the previous k numbers. As with an LCG, the parameter choices (the a's and m) in an MRG are critical components of the quality of the random numbers generated. Our implementation of the MRG is the simplest one available (hence the F as in fast in FMRG) based on using only the last two random numbers generated ($k = 2$) and choosing the a_1 and a_2 coefficients from a special list of numbers. Deng and Lin, the developers of FMRG, report the period as 4,611,686,014,132,420,608. You will probably not revisit the same number.

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

Figure 9.2.2 might appear to paint FMRG as a perfect random number generator. This is not true. Although FMRG is better than RAND and Rnd, it too will exhibit structure when examined under higher magnification. The details of our implementation of Deng and Lin's FMRG are beyond the scope of this book, but additional information and complete documentation are available in the Basic Tools/RandomNumber folder.

Summary

This section has provided a basic review of the principles of random number generation and highlighted an important fact: Not all random number generators are the same. A Monte Carlo simulation based on a poor random number generator is a poor Monte Carlo simulation. The hidden structure in the pseudorandom sequence may completely invalidate the simulation.

The linear congruential random number generators employed by Excel (the RAND function in versions before Excel 2003) and Visual Basic (Rnd) are relatively unsophisticated and exhibit too much structure when successive pairs are plotted. This book will use a more sophisticated random number generator that has an extremely long period and possesses other desirable properties.

This is not to say that the FMRG generator in our RANDOM function is ideal or perfect. It turns out that random number generation is a complex, difficult task. There are many other generators out there (with such colorful names as the Mersenne Twister) and a great deal of debate in the computational science community about the best ones. If you are interested in the details of our random number generator or want references for a more in-depth study of random number theory, please see the Basic Tools/RandomNumber folder.

You should never trust a Monte Carlo simulation without knowing the random number generator used. You should always report the random number generator used in a simulation. We recommend avoiding Excel's RAND unless the application is rudimentary or a simple demonstration. The next section explains how to use RANDOM, the random number generator supplied with this book.

9.3. Random Number Generation in Practice

Workbook: RNGPractice.xls

Although previous section focused on the theory behind random number generation, this section will provide a guide to the practical issues of how to actually get Excel to provide random numbers. In addition to reviewing

	RAND Uniform	NORMINV(RAND(),0,1) Normal	
Average	0.50019275828668	0.00288489306875	
SD	0.29027807011725	1.00048089727527	
Max	0.99953004104712	3.30795409112160	Note how draws near 0 or 1 are
Min	0.00059867432756	−3.23954051612582	translated into numbers like −3.7 or +4.2

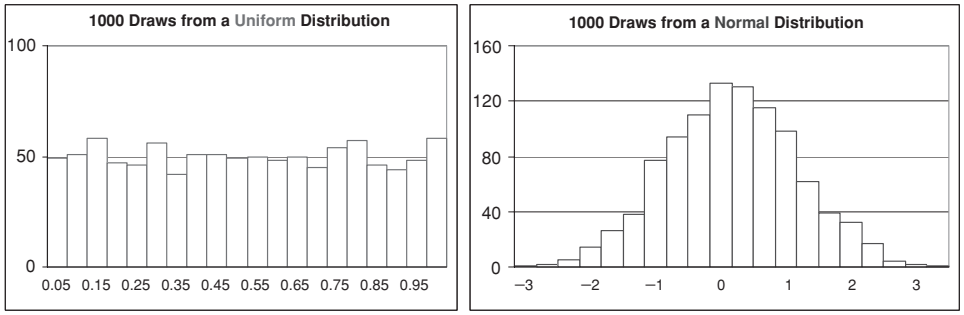

Figure 9.3.1. Output from RAND and NORMINV(RAND(),0,1).
Source: [RNGPractice.xls]NormalRand.

the different formulas available for uniform and normal distributions, this section also reviews how Excel calculates cells.

Before explaining the options available, we warn against generating random numbers with the Data Analysis add-in provided with many versions of Microsoft Excel. Regrettably, not only does the add-in simply provide "dead" values that do not change when the sheet is calculated, but the properties of the random number generator are bad. The Data Analysis add-in should never be used to generate random numbers.

To generate uniformly distributed random numbers with Excel, use either Excel's RAND function or, if the functions packaged with this book are available, use the RANDOM function. Both functions require formulas that use parentheses without any arguments: = RAND() and = RANDOM().

Open the RNGPractice.xls workbook and go to the *Uniform* sheet. Examine the formulas and results in columns A and E. Hit F9 to draw more random numbers. (If it takes a long time for the sheet to draw new random numbers, hit the Recalculate this Sheet Only button instead. There are thousands of cells containing random numbers in the workbook, and every time you hit F9, the workbook must compute formulas to replace every one of them.)

As with the uniform case, there are two ways to obtain normally distributed random numbers. The first approach uses intrinsic Excel functions RAND and NORMINV. The *NormalRand* sheet uses the formula "= NORMINV(RAND(), 0, 1)" to draw 1,000 random numbers from a normal distribution with mean zero and standard deviation one. Hit F9 to draw another 1,000 numbers. The summary statistics and histogram (see Figure 9.3.1) show that NORMINV is working as advertised.

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

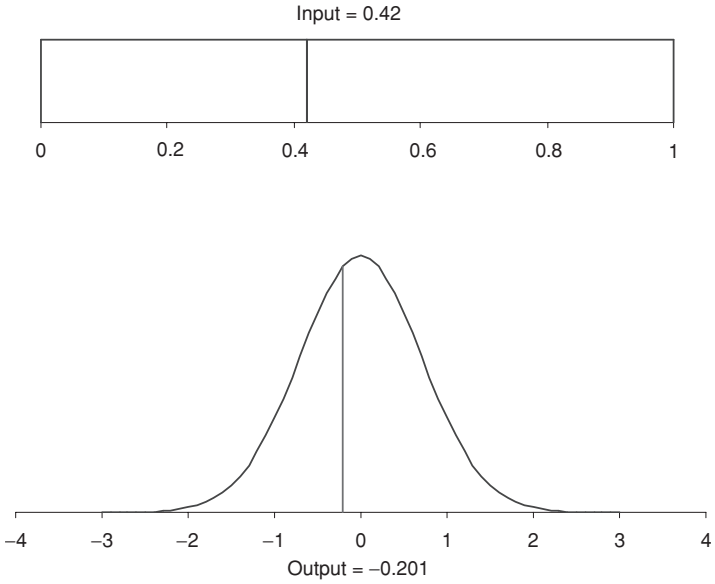

Figure 9.3.2. Converting from uniformly distributed random variables to normally distributed random variables.
Source: [RNGPractice.xls]UniformToNormal.

The *UniformToNormal* sheet explains how the NORMINV function maps numbers that are uniformly distributed into normally distributed numbers. It begins by taking a random number from the Uniform(0,1) distribution – for example, 0.42. Figure 9.3.2 shows that when we graph 0.42 on the Uniform(0,1) distribution, we see that 42 percent of the area under the entire curve lies between 0 and 0.42. We want to translate that number into a normally distributed random number. This is done as follows. Moving down to the Standard Normal curve (with mean 0 and SD 1), we find that value of x such that 42 percent of the area under the standard normal distribution lies between negative infinity and x. This turns out to be -0.201. That is what NORMINV does: given inputs, 0.42 for the area under the curve, 0 for the mean of the normal distribution, and 1 for the SD, NORMINV(0.42,0,1) yields a value of -0.201. Each time you hit F9, the *UniformtoNormal* sheet will draw another uniformly distributed random number and show how that number is converted into a normally distributed random number. The sheet also explains how to obtain numbers that follow a normal distribution other than the Standard Normal distribution.

The *NormalRand* sheet uses the NORMINV and RAND functions to generate normally distributed random variables. Although all seems well, our review of the theory behind random number generation in the previous

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

section explained that RAND is not a great random number generator. In addition, it turns out that NORMINV in versions before Excel 2002 has a rare but serious problem. It can return the nonsensical value of minus 500,000 (or 500,000) whenever the first argument, *y,* is too close to 0 (or 1). NORMINV fails to report the error value #NUM (an indication that the computation is invalid) for values very close to 0 or 1. More modern versions of Excel have partially corrected the badly erroneous results, but testing has shown NORMINV still has problems in Excel 2002 (and XP).[6]

We therefore recommend using the second approach to generating normally distributed random numbers: the NORMALRANDOM function included with this book. The sheet *NormalRandom* shows how to use the formula, "=NORMALRANDOM(mean, SD)" to draw 1,000 numbers quickly and correctly from a normal distribution with given mean and SD.[7]

Although the results of the two sheets are superficially quite similar, remember that RANDOM and NORMALRANDOM are superior to Excel's intrinsic, analogous functions. Of course, you must have these functions properly installed on the computer you are using. Our workbooks come fully prepared with these functions, but you cannot simply type =RANDOM() on a blank spreadsheet because Excel may not have access to the function. You must either open a workbook with the function available or install one of the add-ins packaged with this book (such as the Monte Carlo Simulation add-in described later in this chapter). If =RANDOM() is entered in a cell and Excel displays #NAME?, then the function is not available. Finally, because RAND is a core Excel function, it is somewhat faster than RANDOM and NORMALRANDOM.[8] We believe the trade-off of lower speed for computational superiority is worth it.

You should be aware that if RANDOM or NORMALRANDOM is used on a computer with our software properly installed and you then try to open the workbook from a different computer, an update links notification will be received, as shown in Figure 9.3.3.

If you click the Don't Update button, when the workbook calculates, cells using the RANDOM or NORMALRANDOM functions will display a #NAME? error. If the Update button is clicked, it is possible to change the source to an add-in on the computer you are currently using that has the functions available.

[6] See articles listed in Section 9.8 for more details.

[7] NORMALRANDOM does not make use of the inverse cumulative function. See the Basic Tools/RandomNumber folder for more details on the Box–Muller algorithm used by our function.

[8] Testing has shown that NORMALRANDOM is quite a bit faster than NORMINV(RAND()).

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

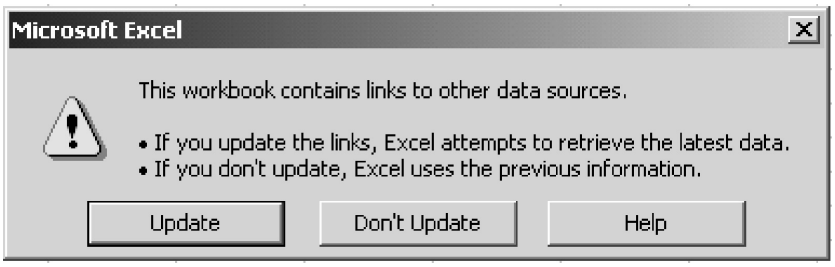

Figure 9.3.3. Update links notification.

We end this section with a brief review of calculation issues in Excel. You may have noticed, as determined by the speed of your computer, that Excel pauses for a few seconds when you hit F9 in the RNGPractice.xls workbook. This is because thousands of cells are being recalculated.

Excel's default calculation setting is to recalculate every cell with a formula in every open workbook automatically whenever any cell is modified. Excel reports its progress in the status bar at the bottom left-hand corner of the screen. Automatic recalculation can be quite cumbersome and tedious when you have a large spreadsheet with many formulas because it is necessary to wait for Excel to finish recalculating after every new entry or change in a cell.

In many of our workbooks, we change the calculation setting to manual by executing Tools: Options and clicking on the Calculation tab (displayed in Figure 9.3.4). Try this now in the RNGPractice.xls workbook. After changing

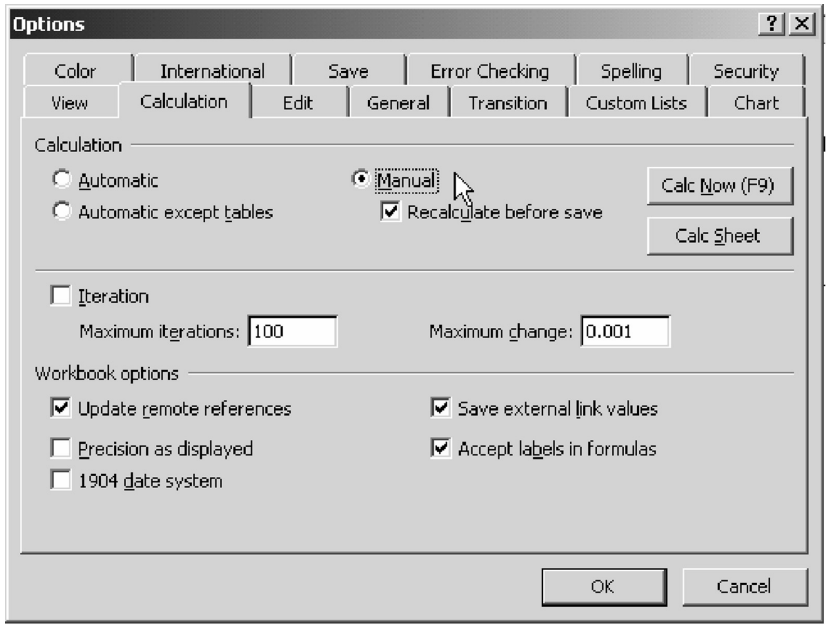

Figure 9.3.4. Controlling calculation.

the calculation setting to manual, enter a number in a blank cell and hit enter. Notice that the sheet does not recalculate and Excel displays the word "Calculate" in the status bar (on the bottom of your screen). This is the signal that the spreadsheet has been altered but the cells have not been recomputed and updated. You can continue to make changes and new entries in cells without pausing for recalculation because Excel is set to manual calculation. You can force calculation when in the manual calculation mode by hitting F9.

Manual calculation is a useful feature with large spreadsheets. Remember that the values displayed in the cells may be wrong, however, when the "Calculation" signal is displayed because the cells have yet to be recalculated.

Summary

This section has shown how to get uniformly and normally distributed random numbers from Excel. The Excel functions RAND and NORMINV can be used for this purpose. This book also provides software with our own functions, RANDOM and NORMALRANDOM, that we recommend and have used under a wide variety of applications.

When a spreadsheet is populated with many thousands of cells with formulas, automatic recalculation can really slow you down. Change the setting to manual calculation and use F9 to recalculate as needed.

Having reviewed the theory of random number generation in the previous section and covered how to generate random numbers within Excel in this section, we now turn to the heart of this chapter: Monte Carlo simulation.

9.4. Monte Carlo Simulation: An Example

Workbook: MonteCarlo.xls

This section presents a concrete example of how Monte Carlo simulation can be used. Suppose we know that Larry Bird, the legendary basketball player, is a 90-percent free-throw shooter. That is, the chance of his making any given free throw is 90 percent regardless of whether he made or missed his previous free throw.[9]

Suppose further that we want to know how well the sample percentage will perform as an estimator of Bird's free-throw accuracy if we have a sample of 100 free throws. Put another way, assume we have Bird, whom we know is truly a 90-percent free-throw shooter, take 100 free-throw attempts. What

[9] According to the Web site <www.larrybird.com/stats.html> Bird's lifetime NBA free-throw percentage was 88.6 percent in the regular season (3,960 made out of 4,471 attempts) and 89.0 percent in the playoffs (901 out of 1012).

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

percentage of the 100 attempts will he be likely to hit? We know that we should see something around 90 percent because that is his true long-run performance. However, because chance plays a role in free-throw shooting, we may well get something different from 90 percent.

Now, the possibilities are anywhere from 0 to 100 percent, but what are the likely or typical results? Is it plausible that we could see him make only 72 out of 100 attempts for a sample percentage of 72 percent? Is making every shot (100 straight free throws), giving him a 100-percent sample percentage, something that we might see every once in a while? Or, are results like 72 and 100 percent so extremely rare as not to be worth worrying about?

In statistics, "rare" and "likely" are important words, whereas "possible" is not too interesting.[10] If results like 72 percent were quite common, we would conclude that a single sample percentage of made shots out of 100 free throws would be a bad way to gauge Bird's true skill. After all, if we did not know his true percentage and had only one sample with which to guess his true, but unknown, shooting percentage, we might get a result like 72 percent and be way off. If, on the other hand, we consistently get a sample percentage within, for instance, 1 percentage point of 90 percent, then it could be argued that the sample percentage of made shots out of 100 free throws is a good gauge of Bird's true skill.

What we are trying to do, of course, is to evaluate the likely size of the spread in the sample percentage of a sample of 100 free throws. Each free throw has some chance built into it, and thus the sample percentage of 100 free throws also has a chance component. We need to figure out how much variation there is in the sample percentage of 100 free throws. In other words, we need to find the SE (standard error) of the sample percentage. A small SE of the sample percentage is good – it means that the observed sample percentages are unlikely to stray far from 90 percent.

There are two routes to figuring out the variation in the sample percentage. The first is statistical theory.[11] The second route is the Monte Carlo approach, which entails producing a simulation of the data generation process, generating a series of replications of that process, and analyzing the results of the experiment. This section shows how to implement this strategy.

The *OneFreeThrow* sheet in the MonteCarlo.xls workbook explains how to use the RANDOM() and IF functions to simulate the result from a single free throw. If the random number drawn is below 0.9, the free throw is made; otherwise, it is missed. Excel registers a "1" for a hit and "0" for a miss.

[10] It is "possible" that a 90-percent free-throw shooter would miss 100 in a row. The likelihood of this outcome, 0.1^{100}, is so remote that we ignore it completely. The chances of making every shot are not so great either – $0.9^{100} = 0.00266$ percent.

[11] We review exactly how statistical theory can be used to solve this problem in the next chapter.

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

To simulate Bird's shooting 100 free throws is simple: just repeat the formula in 100 cells as we show in the sheet called *Sample*. Call the results from 100 "shots" a single *repetition* of the simulation. The key information from a single repetition would be the sample percentage of 1's. You should press F9 per the instructions in the *Sample* sheet to make sure you understand that the sample percentage of 100 attempts varies; press F9 again and again and watch how the sample percentage bounces around. Sometimes Larry does exceptionally well, maybe 94 or 95 percent, but every once in a while he does quite badly – well, never as poorly as Shaq,[12] for instance. Badly for Larry is 85 percent, and below 80 percent is really rare. You might repeatedly press F9 for 20 minutes and not see 79 percent.

Now that you understand how the success or failure of a single free throw is determined via the RANDOM function and IF statement and how we calculate the sample percentage from 100 free throws, we can turn to actually creating and interpreting Monte Carlo simulation results.

To figure out the spread of the sample percentage in the Larry Bird example, we simply conduct many repetitions and examine the resulting empirical histogram of the results. Let us say we perform 1,000 repetitions. Now we have 1,000 sample percentages. We can find the mean of these sample percentages and their SD (standard deviation). You are guaranteed to get an average close to 0.90 (90 percent). The question is, How much spread is there in the 1,000 sample percentages? The SD of the 1,000 sample percentages is a Monte Carlo–generated approximation to the true, exact SE of the sample percentage. Similarly, the empirical histogram of the 1,000 sample percentages approximates the exact probability histogram (or sampling distribution).

Monte Carlo simulation will always be an approximation to the exact truth because the exact truth in a sampling context is based on an infinite number of repetitions. One thousand repetitions will usually generate a fairly good approximation, but 10,000 would be even closer to the truth. No finite number of repetitions, no matter how large, will give the exact answer. Monte Carlo simulation cannot be used to obtain the exact right answer, but it can give an increasingly good approximation as the number of repetitions rises.

We ran a Monte Carlo analysis of the sample percentage of 100 attempts with our simulated Larry Bird shooting free throws. Figure 9.4.1 shows the results.

The bars in the histogram show how many samples of 100 free throws made a particular percentage. Of the 10,000 repetitions of 100 free throws, the lowest sample percentage was 79 percent and the highest was 99 percent. In almost 1,400 samples, the computer simulation of Larry Bird made exactly 90 out of 100 free-throw attempts. The mean of the 10,000 sample percentages

[12] Shaquille O'Neal is a tremendously gifted 7-foot-1-inch athlete in the NBA.

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

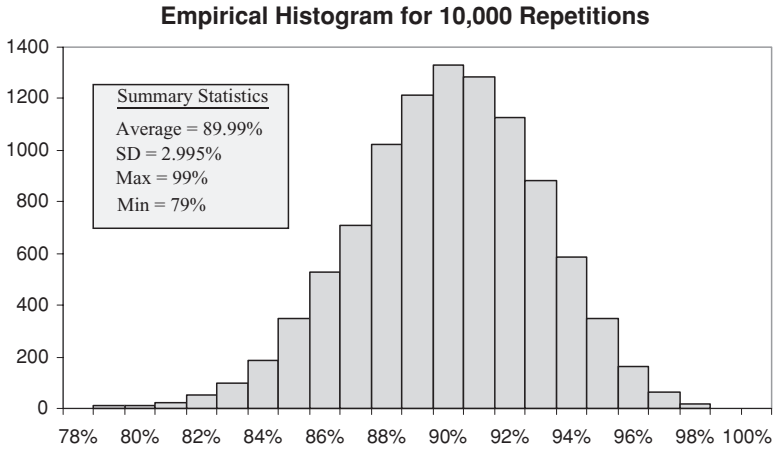

Figure 9.4.1. Monte Carlo simulation of percentage made.
Source: [MonteCarlo.xls]MCSim.

was 89.99 percent with a standard deviation of 2.995 percent. This analysis says that the likely size of chance error for the sample percentage of 100 free throws is about 3 percentage points. Thus, we should not be surprised to find that Larry Bird sinks 87 or 93 percent of his free throws when he makes 100 attempts. It would be very surprising, however, if he hit all 100, or if he hit only 80 out of 100, because these values are more than 3 standard deviations away; in most cases that means such outcomes are rare indeed.

Now it is your turn. From the *Samples* sheet, click on the Run Monte Carlo Simulation button. A new sheet appears in the workbook called *MCSim*, and you are looking at the results of a previous Monte Carlo simulation of the sample percentage of 100 free throws. There is one extremely important difference between Figure 9.4.1 and the graph on the *MCSim* sheet, for the former is "dead" and the latter is "alive." That is, the graph on the Excel sheet will change as the values in column B change. That means you can run your own Monte Carlo simulation as many times as you wish. Simply click on the Run Monte Carlo Simulation button.

A dialog box like the one in Figure 9.4.2 will appear. After clicking the OK button, you will be able to watch the progress of the simulation. So, how did your simulation turn out? Is your histogram similar to ours?

A more subtle implication of the Monte Carlo analysis just performed is that the empirical histogram of the Monte Carlo simulation for Larry Bird appears slightly skewed to the left, which you can see by looking closely at Figure 9.4.1. This is not an accident of our particular run. Look at your simulation results carefully. Is the left tail a little longer than the right? Is the histogram symmetrical around the expected value of 90 percent? In other words, is the fraction of samples with 91 percent made free throws roughly

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

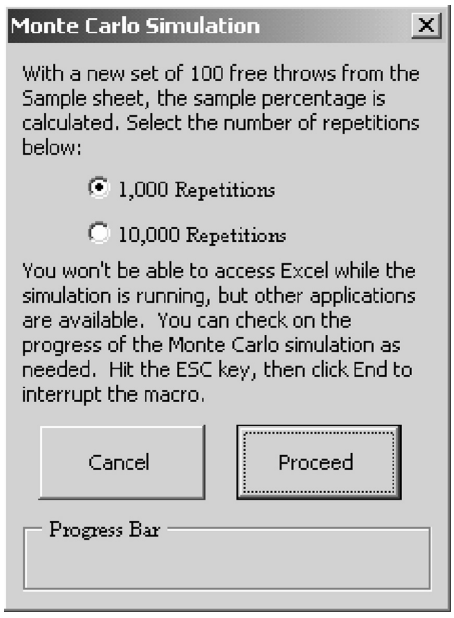

Figure 9.4.2. Running a Monte Carlo simulation.
Source: [MonteCarlo.xls]MCSim.

equal to the fraction of samples with 89 percent? How about the fraction of samples with 88 percent free throws made versus that for 92 percent? Two points can be made here. First, it is not possible to do better than 100 percent, whereas 79 percent and below are possible outcomes. Second, statistical theory tells us that, although the histogram of the sample percentage of 100 free throws ought to follow the normal distribution approximately, it will not be distributed exactly normally. This point is discussed in Chapter 10, in greater depth. For now, we remind you that the central limit theorem tells us that the sampling distribution of the sample percentage comes to resemble the normal distribution more closely as the sample size increases.

Let us summarize the Larry Bird free-throw shooting example. We wanted to know how much spread there was in the sample percentage. Instead of traditional analytical methods based on the theory of probability and statistics, we adopted the Monte Carlo simulation strategy. We resampled repeatedly and thereby obtained an approximation to the SE of the sample percentage of 100 attempts. Our run gave us a value of about 3 percent. What did you get? The formula for the SE of the sample percentage gives us precisely 3 percent.[13] It is, of course, no accident that Monte Carlo experiments yield results close to the standard formulas of statistical theory.

[13] The appropriate formula is

$$\text{SE for sample percentage} = \frac{\sqrt{\text{Probability of 1} \times \text{Probability of 0}}}{\sqrt{\text{Sample size}}}.$$

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

If Monte Carlo simulation will simply reproduce already known answers, why bother? First, it enables you to see clearly the source of chance error and variation in a problem. Formulas often make it difficult to see what is really going on. Although some people quickly understand and accept the notion of randomness and variation, we believe most people learn much better when they actually see variation. We believe many more people will really understand when they hit F9 to draw another sample and see that sample percentage bouncing around. By hitting F9, you are doing and understanding instead of passively reading or listening.

Second, Monte Carlo simulation focuses your attention on the details of the data generation process. The method requires that you set up and implement a chance process. This requires careful thought about the source of the randomness and how it is to be modeled.

Finally, Monte Carlo techniques drive home the concept of the SE, which is surely one of the most difficult ideas in statistics and econometrics for beginning students. The SE measures the spread of outcomes of chance processes. Visually, it is the spread of the probability histogram of the different outcomes of the chance process. The Monte Carlo method allows us to approximate the probability histogram and therefore the SE just by running numerous repetitions of the same data generation process.

Although our primary purpose in using Monte Carlo is to teach you econometrics, we also would like to point out that there are many random variable problems with no analytical solution. That is, traditional statistical theory cannot solve them. This happens in econometrics often when small sample sizes are under consideration. The advent of extremely fast computers has opened a new avenue for solving these problems. Thus, it is not merely a question of a neat alternative to a tried and true approach – Monte Carlo methods offer approximate solutions to previously impossible problems.

To see another example of the Monte Carlo method, click on the Streak Finder button (on the *Sample* sheet near cell D17) a few times. Each time, the longest run of consecutive free throws made in one set of 100 attempts is reported (see Figure 9.4.3).

Streaks in sports are the subject of much debate. Although no one disputes that streaks occur, there is an argument over whether observed streaks are caused by something other than chance.[14] The streaks exhibited by our virtual Larry Bird are due to chance alone because we draw random numbers to determine if a free throw is made.

There is variation in the longest streak of free throws made in each sample of 100 attempts. What is the average longest streak in 100 free throws? What is the spread in the distribution of the maximum streaks? What does the

[14] See the Hot Hand in Sports Web page: <www.hs.ttu.edu/hdfs3390/hothand.htm>.

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

Figure 9.4.3. Maximum streak report.
Source: [MonteCarlo.xls]Sample.

sampling distribution of the maximum streak look like? As before, we forego analytical solutions to these questions in favor of Monte Carlo analysis.[15]

Click on the Monte Carlo Simulation Max Streak button (on the *Sample* sheet near cell D22) to see a demonstration of how a Monte Carlo simulation can be used for approximate determination of the average and spread of the Max Streak sampling distribution. As before, a new sheet, this time named *Streak*, appears in the workbook with results from 1,000 repetitions available for your inspection. Notice that Max Streak is not normally distributed – it has a long right-hand tail.

You might want to try your own Monte Carlo analysis by clicking the Run Monte Carlo Simulation button. Once again, the dialog box will describe the simulation and the progress bar will keep you updated on where the simulation stands. The progress bar is more useful this time because the simulation takes longer (calculating the longest streak in a stretch of 100 free throws is much harder than calculating the percentage made). You can do other work while the simulation is running, but this may slow down the simulation itself (after all, your computer will be busy doing other tasks instead of grinding out the next repetition). If your screen saver comes on, this will also slow down the simulation. You can interrupt the simulation by pressing the Esc (escape) key on the upper left-hand corner of your keyboard. Excel will prompt you with a dialog box, and you can click the End button to stop the simulation. Of course, if you happen to be running on the latest-generation chip, these suggestions are moot because the simulation will fly through 10,000 repetitions.

Summary

The free-throw shooting example in this section demonstrates how Monte Carlo simulation works. We will use Monte Carlo analysis repeatedly to

[15] For an analytical approximation to the exact distribution of the maximum streak problem, see William Feller, *An Introduction to Probability Theory and Its Applications*, Vol. 1, 3rd edition, revised printing, New York: John Wiley and Sons, p.325. Our Monte Carlo results agree with Feller's approximation.

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

examine the properties of statistical estimators and to explain a variety of ideas and concepts in econometrics.

With the computer generating random numbers, it will be fast and easy to draw many random samples and then examine the resulting distribution. This will provide a visual, concrete demonstration of difficult, abstract ideas. In addition, with Excel, you will be able to run your own simulations and compare your results to ours. If a point is unclear, you can always run the simulation again.

9.5. The Monte Carlo Simulation Add-In

Workbooks: MonteCarlo.xls; MCSim.xla (Excel add-in); MCSimSolver.xla (Excel add-in)

The previous section introduced Monte Carlo simulation using a workbook that was especially designed for that purpose. This section shows how to use an Excel add-in packaged with this book that will enable you to run a Monte Carlo simulation from any Excel workbook. The add-in allows you to easily and quickly run Monte Carlos of your own models and chance processes.

The first step is to install the Monte Carlo simulation add-in. The software is in the Basic Tools/ExcelAddIns/MCSim folder. Open the MCSim.doc file in that folder for instructions on how to install the add-in. Having installed the MCSim.xla file, open the MonteCarlo.xls workbook (from the previous section) to test drive the Monte Carlo Simulation add-in. Go to the *Sample* sheet (because this is where the free-throw shooting chance process is implemented in Excel) and execute Tools: MCSim . . . to get the dialog box shown in Figure 9.5.1.

Enter cell B1 (which is the sample percentage) and click the Proceed button; the MCSim add-in will then go to work. It simply recalculates the sheet for as many repetitions as requested and keeps track of the value of cell B1. When finished, it adds a worksheet to the workbook displaying the first 100 repetitions along with summary statistics and a histogram of the complete results (see Figure 9.5.2).

Comparing the results of the Monte Carlo Simulation add-in to the Monte Carlo built into the workbook shows the same substantive results, but the display in the workbook is more readable. The Monte Carlo Simulation add-in does not recognize that the sample percentage from 100 free-throws is not a continuous number (0.91 and 0.92 are possible, but 0.915 is not) because it is built for any chance process. Thus, in many of our workbooks that feature Monte Carlo simulation, we will include the simulation in the workbook and tailor it to the specific problem at hand.

The Monte Carlo Simulation add-in is ideal, however, for exploring problems in greater detail or running Monte Carlos on your own chance processes.

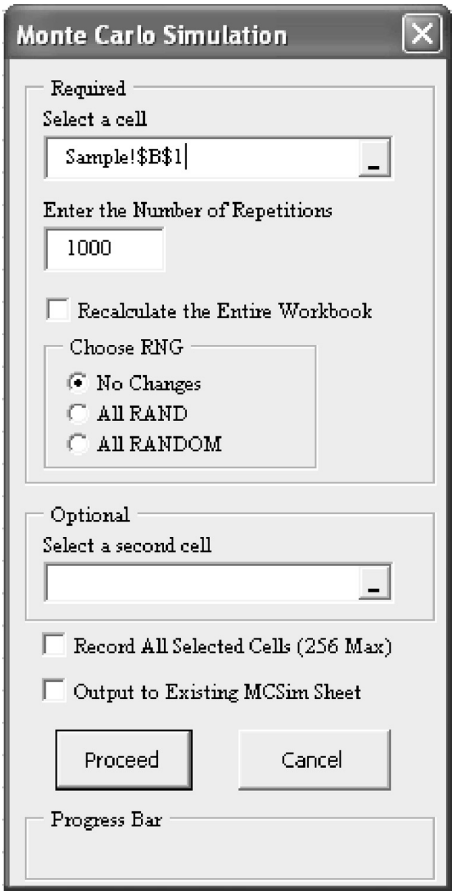

Figure 9.5.1. Preparing to run a Monte Carlo simulation.

For example, in the free-throw shooting model, you might wonder what happens to the spread in the sample percentage as the number of free throws changes. There is no way to explore this question in the MonteCarlo.xls workbook because we did not build in this option. You can easily, however, modify the sheet and use the MCSim add-in to explore this question.

To see how the SE of the sample percentage varies as the sample size changes, create a new cell in the Sample sheet that computes the sample percentage of a different number of free throws. In cell C1 of the *Sample* sheet, we entered the formula, "=AVERAGE(B4:B53)" to obtain the sample percentage of 50 free throws. Now, run the Monte Carlo Simulation add-in using cell C1. You should see that the SD of the 1,000 repetitions (which is our approximation to the true standard error) is larger and the histogram is more spread out and looks even less normally distributed.

You might worry about the bounce in the standard deviation. Remember that a Monte Carlo is never going to give the true, exact answer because that would require an infinite number of repetitions. To obtain a closer approximation to the exact SE of the sample percentage, however, you can increase

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

Summary Statistics			Notes
Average	0.900		
SD	0.0316		
Max	0.990		
Min	0.790		

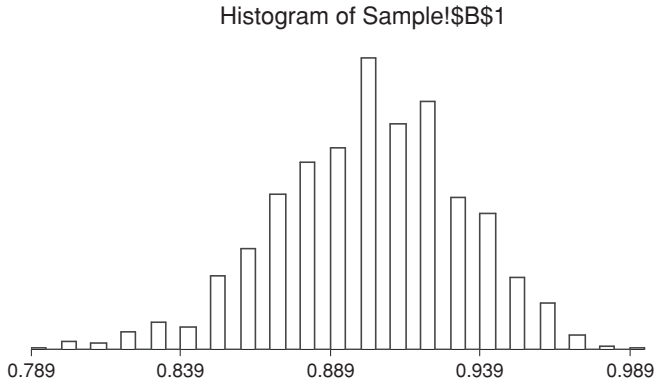

Figure 9.5.2. Results from the Monte Carlo Simulation add-in.
Source: [MonteCarlo.xls].

the number of repetitions. We also recommend that you get in the habit of running and rerunning Monte Carlos if you have doubts about the results. After all, you are just a few clicks away and the computer never gets tired.

You can modify the sheet to explore the sample percentage made of 200 free throws. Simply extend the formula in cell B103 (search for "fill down" in Excel's Help if you do not know how to do this) to cell B203; then find the average of the cells from B4 to B203. Run the Monte Carlo Simulation add-in to see the effect on the standard error. Note that you can compare two cells and the results will be displayed on the same histogram. Run a Monte Carlo that compares the sample percentage of 100 free throws to the sample percentage of 50 free throws.

The pattern is clear: As n (the number of free throws) increases, the SE of the sample percentage falls. In other words, the sampling distribution becomes more tightly concentrated around the true shooting percentage. You have demonstrated that the sample percentage is a consistent estimator of Larry Bird's true shooting percentage, which is an important property of the sample percentage in this chance process. (We study consistency in more depth in Chapter 15.)

Summary

Many of our workbooks will have Monte Carlo simulations that are configured especially for the chance process being discussed. By clicking a button,

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

you can display the results. The Monte Carlo Simulation add-in is a more flexible, general tool. It permits resampling from any chance process that has been modeled in an Excel workbook. Use it to explore advanced ideas and to analyze your own problems via the Monte Carlo method.

Once you use Monte Carlo methods on your own models, you may find a second Monte Carlo add-in that is part of this book especially helpful. The Monte Carlo Simulation with Solver add-in uses a special, non-volatile cell formula, RANDOMNV(), to draw random numbers. After loading the MCSimSolver.xla add-in, you can enter RANDOMNV() as part of a cell formula. Use RANDOMNV() instead of RAND() or RANDOM() as you implement the optimization problem on a worksheet.

The volatility of RAND() and RANDOM() works in our favor when doing conventional Monte Carlo simulation (with MCSim.xla) because we can easily recalculate the sheet, then track the results. However, a Monte Carlo based on running Solver each repetition (e.g., to find a nonlinear least squares fit as discussed in Chapter 22) cannot be implemented with volatile random number formulas because each time Solver puts down a trial solution, the sheet recalculates and gets a new random number. For more information on this advanced Monte Carlo simulation tool, please open the MCSimSolver.doc file in the Basic Tools \ ExcelAddIns \ MCSim folder.

9.6. Conclusion

Random number generation is the heart and soul of Monte Carlo simulation. This chapter has briefly reviewed the theory behind the generation of pseudorandom numbers via linear congruential generators and explained how to obtain random numbers on a spreadsheet with either Excel's own RAND function or the RANDOM function packaged with this book.

Once random numbers are generated on a sheet, it is a short jump to a full-fledged Monte Carlo simulation. By repeatedly resampling and keeping track of the results, we create a concrete, visual representation of sample-based statistics. Our workbooks in the book may have built-in Monte Carlos, or we may use the Monte Carlo Simulation add-in that was introduced in the previous section. We hope the latter will stimulate your creativity and encourage you to model chance processes in a wide a variety of applications.

Econometricians have known that Monte Carlo simulation is an effective way to teach sophisticated concepts involving chance, but actually running a resampling procedure requires writing code, including loops, and storing results. We have completely removed this barrier in our materials. This book will use Monte Carlo methods extensively to race estimators and learn sophisticated concepts that once were accessible only through advanced mathematical means.

P1: IYP
0521843197c09 CB962B/Barretto 0 521 84319 7 September 29, 2005 1:26

9.7. Exercises

1. Change the setup in the *Sample* sheet of MonteCarlo.xls to simulate the free-throw shooting behavior of Shaquille O'Neal, who shoots 50 percent from the free-throw line. Run a 1,000-repetition Monte Carlo simulation of 100 free throws by O'Neal. Of course he will make fewer free throws on average than Bird, but what happens to the spread in the number of free throws made per 100 attempts?
2. Change the setup in the *Sample* sheet of MonteCarlo.xls to simulate a more complicated process. On the very first shot that a player takes, he or she has an 80-percent chance of hitting the free throw. On every subsequent shot, the chances of hitting depend on what happened on the previous attempt. In taking a given shot, if the player missed the previous time, his or her chances of hitting are 70 percent; if the player hit, his or her chances are 90 percent. Run a 1,000-repetition Monte Carlo simulation of 100 such free-throw attempts. What are the Monte Carlo estimates of the expected percentage of free throws made and the SE of the percentage of free throws made?
3. Open the EcolCorr.xls workbook used (in Chapter 2) and run a Monte Carlo simulation (from the *Live* sheet) with 10,000 repetitions that tracks both the individual- and group-level correlation coefficients. Take a picture of your results. Copy and paste the picture in your Word document. Comment on your results.

4A. How do the average and SD reported by the Monte Carlo simulation relate to the expected value and SE?

4B. As the number of repetitions increases, what happens to the expected value and SE?

5A. Use the Record All Selected Cells option and run another 10,000-repetition Monte Carlo. In your 10,000 samples, how many times was the group-level r negative? HINT: Use an IF statement like this: =IF(D3 < 0,1,0), then add the entire column. (Do not forget to hit F9 to calculate the sheet if needed.)

If individual-level $r > 0$ and group-level $r < 0$, then you have an example of the worst form of the ecological fallacy – association reversal.

5B. It is also possible to obtain a negative individual-level r with a positive group-level r. Use your Monte Carlo results to demonstrate this. HINT: Use the IF statement method used in part A.

References

The sources below provide an introduction and serve as an excellent starting point for studying random-number generation. The BasicTools/RandomNumber/ExcelRNGDocumentation folder contains links to a series of Microsoft Knowledge Base articles available online.

Deng, Lih-Yuan and Dennis K. J. Lin (2000). "Random Number Generation for the New Century," *The American Statistician* **54**(2): 145–150.

L'Ecuyer, Pierre (2001). "Software for Uniform Random Number Generation: Distinguishing the Good from the Bad"; available at <www.iro.umontreal.ca/~lecuyer/> and <www.informs-cs.org/wsc01papers/prog01.htm>.

Marsaglia, George, *DIEHARD Battery of Tests*, available online at <stat.fsu.edu/pub/diehard>.

Matsumoto, Makoto and Takuji Nishimura, *Mersenne Twister*, available online at <www.math.sci.hiroshima-u: ac.jp/~m-mat/MT/emt.html> (free Excel implementation available online at <www.numtech.com/NtRand/>).

There is a literature critical of Microsoft Excel's statistical routines. We agree that Excel is not a full-fledged statistical program, but it has served us well in developing examples for teaching econometrics. Different versions of Excel behave differently. To learn more about Excel's statistical performance, we recommend the following:

Knusel, L. (1998). "On the Accuracy of Statistical Distributions in Microsoft Excel 97," *Computational Statistics and Data Analysis* **26**: 375–377 and available online at <www.stat.uni-muenchen.de/~knuesel/elv/excelacc.pdf>.

Knusel, L. "On the Reliability of Microsoft Excel XP for Statistical Purposes," n.d., available online at <www.stat.uni-muenchen.de/~knuesel/elv/excelxp.pdf>.

Knusel, L. (2004). "On the Accuracy of Statistical Distributions in Microsoft Excel 2003." *Computational Statistics & Data Analysis* **48**(3): 445–449.

McCullough, B. D. and Berry Wilson (1999). "On the Accuracy of Statistical Procedures in Microsoft Excel 97." *Computational Statistics & Data Analysis* **31**(1): 27–37.

McCullough, B. D. and Berry Wilson (2002). "On the Accuracy of Statistical Procedures in Microsoft Excel 2000 and Excel XP." *Computational Statistics & Data Analysis* **40**: 713–721.

McCullough, B. D. and Berry Wilson (2005). On the Accuracy of Statistical Procedures in Microsoft Excel 2003." *Computational Statistics & Data Analysis* **49**(4): 1244–1252.

Streaks in sports make for fun examples and can arouse intense debate. In addition to Alan Reifman's Web site on the hot hand in sports (<www.hs.ttu.edu/hdfs3390/hothand.htm>), we suggest the following journal articles:

Albright, S. C. (1993). "A Statistical Analysis of Hitting Streaks in Baseball," *Journal of the American Statistical Association* **88**(424): 1175–1183.

Gilovich, T., R. Vallone and A. Tversky (1985). "The Hot Hand in Basketball: On the Misperception of Random Sequences," *Cognitive Psychology* **17**(3): 295–314.

Additional sources for this chapter include the following:

Drakos, Nikos (1995). Computer Based Learning Unit, University of Leeds. *Introduction to Monte Carlo Methods*, <csep1.phy.ornl.gov/mc/node1.html>.

Ulam, Stanislaw (1991). *Adventures of a Mathematician*, originally published in 1976, University of California Press; p. 199.

von Neumann, John (1951). "Various Techniques Used in Connection with Random Digits," in U.S. Department of Commerce, National Bureau of Standards, *Applied Mathematics Series 12, Monte Carlo Method*. A 42-page booklet on number-generation methods and applications of the Monte Carlo method.