Applied Probability and Stochastic Processes

RICHARD M. FELDMAN
Industrial Engineering Department
Texas A & M University

CIRIACO VALDEZ-FLORES
Sielken, Incorporated

7-03-96
twe

PWS PUBLISHING COMPANY

I(T)P ▪ An International Thomson Publishing Company

Boston ▪ Albany ▪ Bonn ▪ Cincinnati ▪ Detroit ▪ London
Madrid ▪ Melbourne ▪ Mexico City ▪ New York ▪ Paris
San Francisco ▪ Singapore ▪ Tokyo ▪ Toronto ▪ Washington

3

Simulation

Simulation is one of the most widely used probabilistic modeling tools in industry. It is used for the analysis of existing systems and for the selection of hypothetical systems. For example, suppose a bank has been receiving complaints from customers regarding the length of time customers spend waiting in line at the drive-in window. Management has decided to add some extra windows; they now need to decide how many to add. Simulation models can be used to help management determine the number of windows to add. Even though the main focus of this textbook is on building analytical (as opposed to simulation) models, there will be times when the physical system is too complicated for analytical modeling; in such a case, simulation would be an appropriate tool. The idea behind simulation, applied to this banking problem, is that a computer program would be written to generate randomly arriving customers, and then process each customer through the drive-in facility. In such a manner, the effect of having different windows could be determined before the expense of building them is incurred. Or, to continue this example, it may be possible (after covering Chapter 5) to build an analytical model of the banking problem; however, the analyst may want to furnish additional evidence for the validity of the analytical model. (When a model is to be used for decision making that involves large capital expenditures, validation efforts are always time-consuming and essential.) A simulation could be built to model the same system as the analytical model describes. Then, if the two models agree, the analyst would have confidence in their use.

A final reason for simulation is that it can be used to increase understanding of the process being modeled. In fact, this is why Chapter 3 is devoted to simulation. Now that the theory of Markov chains has been taught, simulation can be used to help the intuitive understanding of the dynamics of the process. (We could have used it similarly after Chapter 1, but we felt that it is more interesting to begin work on stochastic processes as soon as possible.) It is impossible to build a simulation of something not understood, so just the process of developing a simulation of a Markov chain will force an understanding of Markov chains.

In this chapter, only a cursory introduction to simulation is given.[1] Our goal is to give enough of an introduction to enable the student to simulate simple probabilistic events, Markov chains, and Markov processes. A more detailed treatment is postponed until Chapter 6. In particular, the use of future event lists and the statistical analyses of simulation output are discussed in the second simulation chapter.

One important topic that is not discussed in this book is the use of a special simulation language. To facilitate the building of simulations, many special-purpose simulation languages have been developed. Some of these languages are very easy to learn, and we recommend that students interested in simulation investigate the various simulation languages.

3.1 EXAMPLES

In this section, we introduce the basic concepts of a simulation through two simple examples. The examples use simulation to determine properties of simple probabilistic events. Although the examples can be solved using analytical equations, they should serve to illustrate the concept of simulation.

EXAMPLE 3.1 Consider a microwave oven salesperson who works in a busy retail store. Often people come into the department merely to browse, while others are interested in purchasing an oven. Of all the people who take up the salesperson's time, 50% end up buying one of the three models available and 50% do not make a purchase. Of those customers who actually buy an oven, 25% purchase the plain model, 50% purchase the standard model, and 25% purchase the deluxe model. The plain model yields a profit of \$30, the standard model yields a profit of \$60, and the deluxe model yields a profit of \$75.

The salesperson wishes to determine the average profit per customer. (We assume that the salesperson does not have a mathematical background and therefore is unable to calculate the exact expected profit per customer. Hence, an alternative to a mathematical computation for estimating the profit must be used.) One approach for estimating the average profit would be to keep records of all the customers who talk to the salesperson and, based on the data, calculate an estimate for the expected profit per customer. However, there is an easier and less time-consuming method: The process can be simulated.

---

[1] Some of the material presented in Chapters 3 and 6 is taken from G. L. Curry, B. L. Deuermeyer, and R. M. Feldman, *Discrete Simulation: Fundamentals and Microcomputer Support* (Oakland, Calif.: Holden-Day, Inc., 1989).

The basic concept in simulation is to generate random outcomes (for example, we might toss a coin or roll dice to generate outcomes) and then associate appropriate physical behavior with the resultant (random) outcomes. To simulate the microwave oven buying decision, a (fair) coin is tossed with a head representing a customer and a tail representing a browser; thus, there is a fifty-fifty chance that an individual entering the department is a customer. To simulate the type of microwave that is bought by an interested customer, two fair coins are tossed. Two tails (which occur 25% of the time) represent buying the plain model, a head and a tail (which occur 50% of the time) represent buying the standard model, and two heads (which occur 25% of the time) represent buying the deluxe model. Table 3.1 summarizes the relationship between the random coin tossing results and the physical outcomes being modeled. Table 3.2 shows the results of repeating this process 20 times to simulate 20 customers and their buying decisions.

If we take the final cumulative profit and divide it by the number of customers, the estimate for the expected profit per customer is calculated to be $34.50. Two facts are immediately obvious. First, the number of interested customers was 12 out of 20, but because there is a fifty-fifty chance that any given customer will be interested, we expect only 10 out of 20. Furthermore, utilizing some basic probability rules, a person knowledgeable in probability would determine that the theoretical expected profit per customer is only $28.125. Thus, it is seen that the simulation does not provide the exact theoretical values sought.

The simulation is, in fact, just a statistical experiment. This point cannot be overemphasized. *The results of a simulation involving random numbers must be interpreted statistically.* In Chapter 6, some basic statistical concepts will be given to aid in the proper analysis of simulation results. For now it is important to realize that the simulation is simply a statistical experiment performed so that the expense and time needed to perform and/or to observe the actual process can be avoided. For instance, this experimental evaluation might require less than an hour to accomplish, whereas obtaining an estimate of the theoretical profit by observing the actual process for 20 customers might take days.

TABLE 3.1 Procedure for determining the interest of a potential customer and the microwave model an interested customer buys.

Random Value	*Simulated Outcome*
Head	Customer
Tail	Browser

Random Value	*Simulated Outcome*
Tail-tail	Buy plain
Tail-head	Buy standard
Head-tail	Buy standard
Head-head	Buy deluxe

TABLE 3.2 Simulated behavior of 20 customers.

Customer Number	*Coin Toss*	*Interested?*	*Two-Coin Toss*	*Profit*	*Cumulative Profit*	*Average Profit*
1	T	No	—	0	0	0.00
2	H	Yes	TH	60	60	30.00
3	H	Yes	HH	75	135	45.00
4	H	Yes	TH	60	195	48.75
5	T	No	—	0	195	39.00
6	T	No	—	0	195	32.50
7	H	Yes	TT	30	225	32.14
8	T	No	—	0	225	28.13
9	H	Yes	HT	60	285	31.67
10	H	Yes	HH	75	360	36.00
11	H	Yes	TT	30	390	35.45
12	H	Yes	TT	30	420	35.00
13	T	No	—	0	420	32.31
14	T	No	—	0	420	30.00
15	T	No	—	0	420	28.00
16	H	Yes	TH	60	480	30.00
17	H	Yes	HT	60	540	31.76
18	T	No	—	0	540	30.00
19	H	Yes	HH	75	615	32.37
20	H	Yes	HH	75	690	34.50

EXAMPLE 3.2 Let us consider a game taken from a popular television game show. The contestant is shown three doors, labeled *A*, *B*, and *C*, and told that behind one of the three doors is a pot of gold worth $120,000; there is nothing behind the other two doors. The contestant is to pick a door. After the door is selected, the host will pick one of the other two doors, one with an empty room. The contestant will then be given the choice of switching doors or keeping the originally selected door. After the contestant decides whether or not to switch, the game is over and the contestant gets whatever is behind the door finally chosen. The question of interest is "Should the contestant switch when given the option?" We simulate here the "no-switch" option and refer to Exercise 3.2 for the "switch" option.

To simulate the "no-switch" policy, it is necessary to simulate the random event of placing the gold behind a door, and simulate the event of selecting a door. Both events will be simulated by rolling a single die and interpreting the outcome according to Table 3.3. Since a switch is never made, it is not necessary to simulate the host selecting a door after the initial selection.

The results from 20 simulated games are shown in Table 3.4. Based on the simulation run of Table 3.4, the estimate for the expected yield is $36,000 per game. (Of course, it is easy to calculate that the actual expected yield is $40,000 per game, which illustrates that simulations only yield statistical estimates.) As can be seen from these two examples, a difficulty in simulation is deciding what information to maintain, and then keeping track

TABLE 3.3 Random events for the "pot-of-gold" game.

Random Value	*Simulated Outcome*
1 or 2	*A*
3 or 4	*B*
5 or 6	*C*

of that information. Once the decision is made as to what information is relevant, the simulation itself is straightforward.

▶ *Suggestion: Do Exercises 3.1 and 3.2.*

3.2 GENERATION OF RANDOM VARIATES

System simulation depends heavily on random number generation to model the stochastic nature of the systems being studied. The examples of the previous section simulated random outcomes by having the modeler physically toss a coin or a die. A computer, by contrast, must simulate random

TABLE 3.4 Simulated results from the "pot-of-gold" game using a no-switch policy. Yield is in terms of thousands of dollars.

Game Number	*Die Toss*	*Door for Gold*	*Die Toss*	*Door Chosen*	*Yield (thousands)*	*Cumulative Yield*	*Average Yield*
1	5	*C*	2	*A*	0	0	0.00
2	3	*B*	5	*C*	0	0	0.00
3	5	*C*	4	*B*	0	0	0.00
4	3	*B*	2	*A*	0	0	0.00
5	2	*A*	1	*A*	120	120	24.00
6	1	*A*	6	*C*	0	120	20.00
7	6	*C*	4	*B*	0	120	17.14
8	4	*B*	4	*B*	120	240	30.00
9	5	*C*	5	*C*	120	360	40.00
10	5	*C*	3	*B*	0	360	36.00
11	4	*B*	2	*A*	0	360	32.73
12	3	*B*	4	*B*	120	480	40.00
13	2	*A*	4	*B*	0	480	36.92
14	1	*A*	3	*B*	0	480	34.29
15	1	*A*	3	*B*	0	480	32.00
16	4	*B*	6	*C*	0	480	30.00
17	5	*C*	6	*C*	120	600	35.29
18	4	*B*	4	*B*	120	720	40.00
19	1	*A*	6	*C*	0	720	37.94
20	2	*A*	4	*B*	0	720	36.00

outcomes by generating numbers to give the appearance of randomness. This section deals with the problems involved in programming a computer to simulate randomness. The goal is to introduce the tools necessary for statistically describing a physical situation for modeling purposes.

3.2.1 Uniform Random Numbers

Most mathematical table books contain several pages of random numbers. Such *random numbers* have three properties: (1) The numbers are between zero and one, (2) the probability of selecting a number in the interval (a, b) is equal to $(b - a)$, where $0 \leq a < b \leq 1$, and (3) the numbers are statistically independent. The concept of statistical independence simply means that if several numbers are chosen, knowledge of the value of one number does not provide information that will help in predicting the value of another number. Independence, therefore, rules out the possibility of trends or cyclic patterns occurring within a sequence of random numbers. In terms of probability, these random numbers are sequences of independent, identically distributed random variates having a continuous uniform distribution between zero and one.

Technically, random numbers can refer to the observations of random variables having any arbitrary probability distribution; however, usually the term *random numbers* refers to the observations of uniform zero-one random variables. For observations of random variables governed by a distribution other than the uniform zero-one, the more general term *random variates* is used. Thus, the term *random numbers* as used in this text will always refer to numbers that statistically reproduce observations of zero-one uniform random variables, and the term *random variates* will refer to numbers that statistically reproduce observations of arbitrarily distributed random variables.

As shown later, random numbers are used to generate random variates. Thus, the first task we must be able to do with a computer is generate random numbers by way of a simple and fast algorithm. Because a computer can follow only very specific steps, random numbers generated by a computer are not truly random but are more properly called "pseudo-random numbers" (although the "pseudo" prefix is usually dropped). Numbers generated by a computer program or subprogram are, therefore, called "random" if statistical tests cannot determine the difference between computer-generated and truly random sequences of numbers.

All methods of random number generation in computers begin with an initial value called the *initial random number seed*. Each time a random number is desired, the random number seed is used to produce the next random number and the seed is transformed into another number that becomes the new seed. Thus, the random number seed is continually changed as the random numbers are generated. A simulation program must always furnish an initial random number seed. If the same initial random number seed is used whenever the simulation program is run, the exact same string of random numbers will be generated. If a different string of random numbers is desired, then a different initial random number seed must be furnished.

One of the most popular methods of generating random numbers by a computer is the *congruential method.* In this method the random number seed is always a positive integer, and the random number associated with the seed is the value of the seed divided by the largest possible integer that can be stored in the computer. To obtain the new seed, the old seed is multiplied by a large constant, and a second constant is added to the product. The resulting integer might then be larger than the computer word size (i.e., larger than the largest possible integer for the computer size). If it is too large, it is divided by the largest possible integer, and the remainder is the new seed. If it is not too large for the computer word size, it becomes the new seed. To express this mathematically, let a and b be two fixed integers, and let L denote the largest possible (signed) integer that the computer can store. Let S be a random seed and let S_{next} be the next seed to be determined. The random number associated with the seed is

$$R = \frac{S}{L},$$

and the next seed is

$$S_{\text{next}} = (aS + b) \,\mathbf{mod}\, L.$$

For example, for a 16-bit microcomputer, $L = 32767 = (2^{15} - 1)$ and we might set $a = 1217$, set $b = 0$, and let the initial random number seed be $S_0 = 23$. For this situation, the random number sequence is generated by the following calculations:

$$\begin{aligned}
S_1 &= (1217 \times 23) \,\mathbf{mod}\, 32767 = 27991 \\
R_1 &= \frac{27991}{32767} = 0.85424 \qquad \text{(first random number)}, \\
S_2 &= (1217 \times 27991) \,\mathbf{mod}\, 32767 \\
&= 34065047 \,\mathbf{mod}\, 32767 = 20134 \\
R_2 &= \frac{20134}{32767} = 0.61446 \qquad \text{(second random number)}, \\
&\vdots
\end{aligned}$$

Several rules of thumb are used to determine values for the constants a and b that will produce good pseudo-random number strings. However, most high-level languages contain their own random number generators; although you may not need to be concerned with programming[2] these methods, a conceptual understanding of random number generation is beneficial. It is

---

[2] See Section B.3 of Appendix B if you need a code for random number generation.

especially important to remember the role of the initial random number seed. If the initial seed is the same for every run of a given simulation program, the output will be the same.

3.2.2 Discrete Random Variates

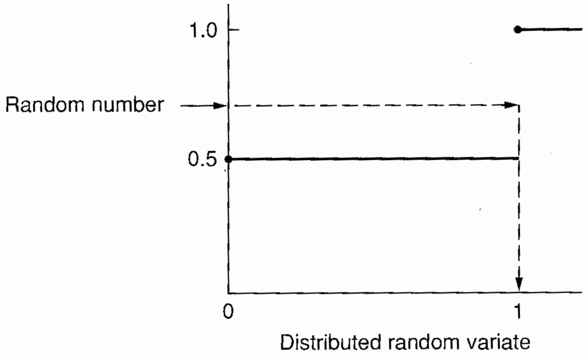
Let us assume that we want to write a computer simulation model of the microwave salesperson situation of Example 3.1. An immediate problem arises because the example involves two random variables, neither of which is a continuous uniform zero-one random variable. The first random variable represents the decision as to whether or not a customer will end up buying a microwave. Mathematically, we can denote the customer's interest by the random variable C and give its probability mass function as

$$\Pr\{C = 0\} = 0.5,$$
$$\Pr\{C = 1\} = 0.5,$$

where the random variable being zero ($C = 0$) represents a customer who is not interested in purchasing a microwave, and the random variable being one ($C = 1$) represents the customer who purchases a microwave.

The cumulative probability distribution function (whose values are between zero and one) permits an easy transformation of a random number into a random variate arising from another probability law. The transformation is obtained simply by letting the random number occur along the ordinate (y-axis) and using the cumulative function to form the inverse mapping back to the abscissa (x-axis). For the "customer interest" random variable C, the following rule is obtained: If the random number is less than or equal to 0.5, then $C = 0$, i.e., the customer is not interested in a purchase; otherwise, if the random number is greater than 0.5, then $C = 1$, i.e., the customer is interested. This transformation is represented schematically in Figure 3.1 for one possible value of the random number.

FIGURE 3.1 Transformation using the cumulative distribution function to obtain the random variate C.



The second random variable that must be simulated in the microwave salesperson example is the profit resulting from the sale of a microwave. Let the "model" random variable be denoted by M with probability mass function given by

$$\Pr\{M = 30\} = 0.25,$$
$$\Pr\{M = 60\} = 0.50,$$
$$\Pr\{M = 75\} = 0.25.$$

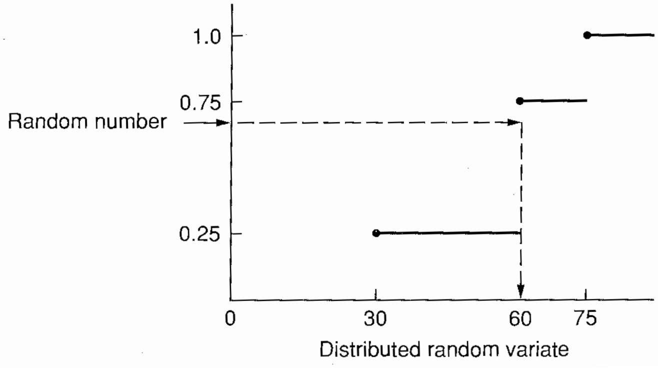
In this case, $M = 30$ is the profit from the sale of a basic model, $M = 60$ the profit from a standard model, and $M = 75$ the profit from a deluxe model. Again, it is easiest to simulate this random variable if its cumulative probability distribution function is first determined. The jump points of the cumulative function are given by

$$\Pr\{M \leq 30\} = 0.25,$$
$$\Pr\{M \leq 60\} = 0.75,$$
$$\Pr\{M \leq 75\} = 1.0.$$

The inverse mapping that gives the transformation from the random number to the random variate representing the profit obtained from the sale of a microwave is illustrated in Figure 3.2. The results are as follows: If the random number is less than or equal to 0.25, the plain model is sold; if it is greater than 0.25 and less than or equal to 0.75, the standard model is sold; and if it is greater than 0.75, the deluxe model is sold.

The rather intuitive approach for obtaining the random variates C and M as shown in Figures 3.1 and 3.2 needs to be formalized for use with continuous random variables. Specifically, Figures 3.1 and 3.2 indicate that the random variates were generated by using the inverse of the cumulative probability distribution function (CDF). A mathematical justification for this is now given.

FIGURE 3.2 Transformation using the cumulative distribution function to obtain the random variate M.



The following property (proven in most introductory mathematical probability and statistics books) leads to the so-called *inverse transformation method* of generating arbitrarily distributed random variates.

Property 3.1 Let R be a random variable with a continuous uniform distribution between zero and one, and let F be an arbitrary cumulative distribution function. If the inverse of the function F exists, denote it by F^{-1}; otherwise, let $F^{-1}(a) = \min\{t \mid F(t) \geq a\}$. Then the random variable X defined by

$$X = F^{-1}(R)$$

has a distribution function given by F; that is,

$$P\{X \leq a\} = F(a) \quad \text{for } -\infty < a < \infty.$$

From Property 3.1, we can see that the function that relates the continuous uniform zero-one random variable with another random variable is simply the inverse of the cumulative probability distribution function. We utilize this property in the next section. First, however, we give some examples simulating discrete random variables.

EXAMPLE 3.3 A manufacturing process produces items with a defect rate of 15%; in other words, the probability of an individual item being defective is 0.15 and the probability that any one item is defective is independent of whether or not any other item is defective. The items are packaged in boxes of four. An inspector tests each box by randomly selecting one item in that box. If the selected item is good, the box is passed; if the item is defective, the box is rejected. The inspector has just started for the day. What is the expected number of boxes that the inspector will look at before the first rejection?

Let the random variable M denote the number of defective items per box. The distribution for M is binomial [Eq. (1.10)]; therefore, its pmf is given by

$$\begin{aligned}
\Pr\{M = 0\} &= 0.5220,\\
\Pr\{M = 1\} &= 0.3685,\\
\Pr\{M = 2\} &= 0.0975,\\
\Pr\{M = 3\} &= 0.0115,\\
\Pr\{M = 4\} &= 0.0005.
\end{aligned}$$

The transformation from random numbers to random variates is based on the CDF, so the above probabilities are summed to obtain Table 3.5. (There are more efficient techniques than the one demonstrated here for generating some of the common random variates;[3] however, our purpose here is to

---

[3] An excellent reference is A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis* (New York: McGraw-Hill, Inc., 1991).

TABLE 3.5 Procedure for determining the number of defective items per box.

Random Number Range	*Number of Defective Items*
(0.0000, 0.5220]	0
(0.5220, 0.8905]	1
(0.8905, 0.9880]	2
(0.9880, 0.9995]	3
(0.9995, 1.0000]	4

demonstrate the concept of simulation, not to derive the most efficient procedures for large-scale programs.) Notice that the ranges for the random numbers contained in the table are open on the left and closed on the right. That choice is arbitrary; it is only important to be consistent. In other words, if the random number was equal to 0.8905, the random variate representing the number of defective items in the box would equal 1.

The simulation proceeds as follows: First, a box containing a random number of defective items is generated according to the binomial distribution (Table 3.5). Then, an item is selected using a Bernoulli random variable, where the probability of selecting a defective item equals the number of defective items divided by four (i.e., the total number of items in a box). Table 3.6 shows the results of the simulation. Notice that column 4 contains a critical number, which equals the number of defective items divided by four. If the next random number generated is less than the critical number, the box is rejected; otherwise, the box is accepted. If the critical number is 0, no random number is generated since the box will always be accepted no matter which item the inspector chooses. The simulation is over when the first box containing defective items is selected. Since the question is "How many boxes will be accepted?" it is necessary to make several simulation runs. The final estimate would then be the average length of the runs. To generate random numbers, we use Table C.6 and start in (the randomly selected) row 16, picking numbers sequentially going across the row.

Obviously, only three trials is not enough to draw any reasonable conclusion, but the idea should be sufficiently illustrated. Based on the results of Table 3.6, we would estimate that the inspector samples an average of 6.67 boxes until finding a box to be rejected. ▨

Before leaving discrete random variables, a word should be said about generating bivariate random variables. One procedure is to use conditional distributions. For example, if the two dependent random variables X and Y are to be generated, we first generate X according to its marginal distribution, then generate Y according to its conditional distribution given the value of X.

TABLE 3.6 Three simulation runs involving boxes containing defective items.

Box Number	Random Number	Number of Defects	Critical Number	Random Number	Box Accepted?
Trial #1					
1	0.2358	0	0	—	Yes
2	0.5907	1	0.25	0.3483	Yes
3	0.6489	1	0.25	0.9204	Yes
4	0.6083	1	0.25	0.2709	Yes
5	0.7610	1	0.25	0.5374	Yes
6	0.1730	0	0	—	Yes
7	0.5044	0	0	—	Yes
8	0.6206	1	0.25	0.0474	No
Trial #2					
1	0.8403	1	0.25	0.9076	Yes
2	0.3143	0	0	—	Yes
3	0.0383	0	0	—	Yes
4	0.0513	0	0	—	Yes
5	0.9537	2	0.50	0.6614	Yes
6	0.1637	0	0	—	Yes
7	0.4939	0	0	—	Yes
8	0.6086	1	0.25	0.1542	No
Trial #3					
1	0.2330	0	0	—	Yes
2	0.8310	1	0.25	0.8647	Yes
3	0.2143	0	0	—	Yes
4	0.5414	1	0.25	0.0214	No

EXAMPLE 3.4 **Bivariate Discrete Random Variates.** Consider the boxes of phones discussed in Chapter 1. Each box contains two phones, both phones being radio phones or plain phones. The joint pmf describing the probabilities of the phone type and the number of defective phones is given in Example 1.12. Our goal is to estimate, through simulation, the expected number of defective phones per box. (Again, the question we are trying to answer is trivial, but we need to simulate easy systems before attempting complex systems.) Although we could use the joint pmf to generate the bivariate random variables directly, it is often easier to use conditional probabilities when dealing with more complex probability laws.

We first determine whether the box contains plain or radio phones, and then determine the number of defective phones in the box. Table 3.7 shows the transformations needed to obtain the boxes of phones; namely, the first transformation is from the CDF of the random variable indicating phone type, and the second two transformations come from the *conditional* CDFs for the number of defective phones in a box given the type of phone. For example, the conditional probability that there are no defective phones given that the

TABLE 3.7 Transformations for determining the contents of phone boxes.

Type of Phone	
Random Number Range	*Type Phone*
(0.00, 0.47]	Plain
(0.47, 1.0]	Radio

For Plain Phones	
Random Number Range	*Number Defective*
(0.000, 0.787]	0
(0.787, 0.957]	1
(0.957, 1.0]	2

For Radio Phones	
Random Number Range	*Number Defective*
(0.000, 0.849]	0
(0.849, 0.981]	1
(0.981, 1.0]	2

box contains plain phones is $0.37/0.47 = 0.787$ (see Example 1.12); thus, if a box contains plain phones, a random number in the interval $(0, 0.787]$ will result in no defective phones for that box.

The results obtained after simulating ten boxes is contained in Table 3.8. The random numbers begin in (the randomly selected) row 3 of Table C.6. The second random number used for each box (column 4 of Table 3.8) must be interpreted based on the result of the first random number (column 3 of Table 3.8). In particular, notice the results for box 4. The first random number was 0.1454, yielding a box of plain phones. The second random number was 0.9683, yielding two defective phones. However, if the box had been radio phones, the random number of 0.9683 would have yielded only one defective phone. Based on the results of Table 3.8, the expected number of defective phones per box is estimated to be 0.5. ▨

▨ **EXAMPLE 3.5** **Markov Chains.** Our final example for discrete random variables will be illustrating the simulation of Markov chains, which again use conditional dis-

TABLE 3.8 Simulation for boxes of phones.

Box Number	*Random Number*	*Phone Type*	*Random Number*	*Number Defective*	*Cumulative Defectives*
1	0.6594	Radio	0.7259	0	0
2	0.6301	Radio	0.1797	0	0
3	0.3775	Plain	0.5157	0	0
4	0.1454	Plain	0.9683	2	2
5	0.7156	Radio	0.5140	0	2
6	0.9734	Radio	0.9375	1	3
7	0.7269	Radio	0.2228	0	3
8	0.1020	Plain	0.9991	2	5
9	0.9537	Radio	0.6292	0	5
10	0.4340	Plain	0.2416	0	5

tributions for the random number to state space transformation. Consider a Markov chain with state space $\{a, b, c\}$ and with the following transition matrix:

$$\boldsymbol{P} = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.4 & 0.6 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}.$$

From the previous chapter, we know that the expected number of steps taken until the chain is absorbed into state c given that it starts in state a is $R(a, a) + R(a, b) = 12.5$. However, let us assume that as a new student of Markov chains, our confidence level is low in asserting that the mean absorption time starting from state a is 12.5. (Or equivalently, $E[T^c \mid X_0 = a] = 12.5$.) One of the uses of simulation is as a confidence builder; therefore, we wish to simulate this process to help give some evidence that our calculations are correct. In other words, our goal is to simulate this Markov chain starting from state a and recording the number of steps taken until reaching state c.

In this simulation, we use the random numbers listed in Appendix C (Table C.6) to produce the necessary randomness. Therefore, the first step is to record the transformations from the random numbers to the distributions defined by the matrix $\boldsymbol{P}$. These transformations are given in Table 3.9.

We begin the simulation by rolling two dice and coming up with the number 7; therefore, the first random number will be at the start of row 7 in Table C.6. (We choose the starting point randomly so that all the simulations will not be identical.) Table 3.10 contains the results of the three realizations of the experiment, where each experiment records the steps of the Markov chain until it reaches the absorbing state.

The estimate for the expected number of steps until the chain reaches state c based on the above three trials is $(5 + 7 + 1)/3 = 4\frac{1}{3}$. If we did not know how to determine the theoretical answer and if we were satisfied with only three trials, our simulation would give very misleading results. This should emphasize the extremely important fact that a simulation is a statistical experiment whose output must be interpreted statistically. Therefore, our suggestion is that a minimum of 25 to 30 trials should be used when estimating means of random variables. (Reasonable variations from the 25- to 30-trial minimum will make sense after Chapter 6.) ▨

▶ *Suggestion: Do Exercises 3.3–3.9.*

TABLE 3.9 Procedure for determining the transitions for the Markov chain.

From State a	
Random Number Range	*Next State*
(0.0, 0.8]	a
(0.8, 0.9]	b
(0.9, 1.0]	c

From State b	
Random Number Range	*Next State*
(0.0, 0.4]	a
(0.4, 1.0]	b

TABLE 3.10 Three simulation runs for the Markov chain.

Step Number	Current State	Random Number	Next State
Trial #1			
0	a	0.2419	a
1	a	0.1215	a
2	a	0.6602	a
3	a	0.7957	a
4	a	0.9518	c
5	c		
Trial #2			
0	a	0.1589	a
1	a	0.5695	a
2	a	0.8823	b
3	b	0.4605	b
4	b	0.5212	b
5	b	0.1180	a
6	a	0.9862	c
7	c		
Trial #3			
0	a	0.9965	c
1	c		

3.2.3 Continuous Random Variates

Although many physical systems can be modeled using discrete random variables, many systems are more appropriately modeled using continuous random variables. For example, the times between arrivals of customers to a teller window in a bank would be best described by a continuous value instead of a discrete value.

One of the most commonly encountered continuous random variables is the exponentially distributed random variable. Specifically, if T is an exponentially distributed random variable, then its cumulative distribution function is given by

$$F(a) = 1 - e^{-a/\theta} \quad \text{for } a \geq 0, \tag{3.1}$$

where θ is a positive scalar with $E[T] = \theta$.

The *inverse transformation method* (Property 3.1) can be used to generate exponentially distributed random variates from random numbers. (See Figure 3.3 for a graph illustrating the transformation from R to T.) To see this, let R be a random number and T be an exponentially distributed random variate. By Property 3:1, we have the following:

$$R = 1 - e^{-T/\theta}$$
$$e^{-T/\theta} = 1 - R$$

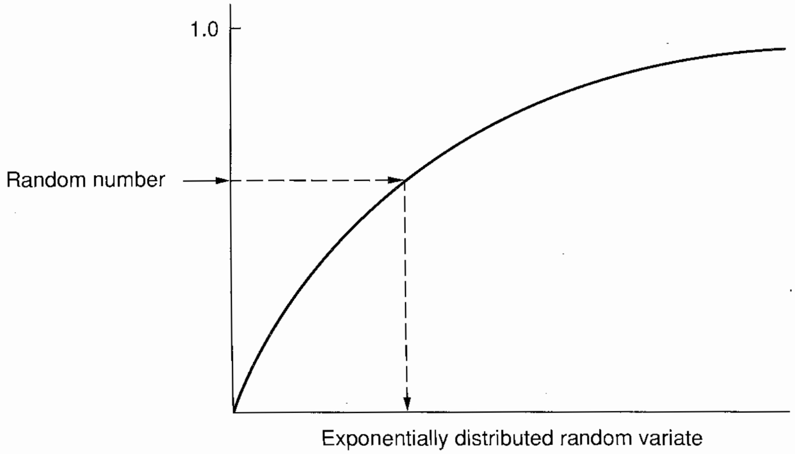


FIGURE 3.3 Inverse transformation for the exponentially distributed random variable T.

$$\ln(e^{-T/\theta}) = \ln(1 - R)$$
$$-\frac{T}{\theta} = \ln(1 - R)$$
$$T = -\theta \ln(1 - R).$$

One final simplification can be made by observing that if R is uniformly distributed between 0 and 1, then $(1 - R)$ must also be; therefore,

$$T = -\theta \ln(R).$$

Since the Erlang distribution [Eq. (1.16)] is the sum of exponentials, the preceding transformation is easily extended for Erlang random variates. Specifically, if X is an Erlang random variate with parameters m and λ, then

$$X = -\frac{1}{m\lambda}\sum_{k=1}^{m}\ln(R_k) = -\frac{1}{m\lambda}\ln\left(\prod_{k=1}^{m} R_k\right),$$

where R_k represents independent random numbers.

The exponential distribution is convenient to use because the expression for its cumulative distribution function is easy to invert. However, many distributions are not so "well behaved." A normally distributed random variable (its probability density function has the familiar bell-shaped curve) is an example of a distribution whose cumulative probability distribution function cannot be written in closed form, much less inverted. An efficient method of generating normally distributed random variables is based on a derivation by Box and Muller.[4] They established that if R_1 and R_2 are

---

[4] G. E. P. Box and M. E. Muller, "A Note on the Generation of Random Normal Deviates," *Annals of Mathematical Statistics* **29** (1958) pp. 610–611.

two independent random numbers, then Z_1 and Z_2 are two independent normally distributed random variates with mean zero and standard deviation of one, where

$$Z_1 = \sqrt{-2\ln(R_1)}\cos(2\pi R_2)$$

$$Z_2 = \sqrt{-2\ln(R_1)}\sin(2\pi R_2).$$

Several other techniques are used for generating nonuniform random variates when the inverse of the cumulative distribution function cannot be obtained in closed form. However, it is not necessary to know these techniques, since either the advanced techniques are buried in the simulation language being used (and thus are transparent to the user) or else an approximation method can be used quite adequately.

EXAMPLE 3.6 **Compound Poisson Process.** A monitoring device has been placed on a highway to record the number of vehicles using the highway. Actually, what is recorded are axles, so we would like to simulate the number of axles passing a particular point. The time between vehicle arrivals at the monitoring device is exponentially distributed with a mean of 1.2 minutes between vehicles. The probability that a given vehicle has 2, 3, 4, or 5 axles is 0.75, 0.10, 0.07, or 0.08, respectively. We are interested in simulating this process to obtain an estimate for the expected number of axles that pass over the monitoring device during any fixed 5-minute interval. There are two sources of randomness: (1) the interarrival times and (2) the number of axles per vehicle. Table 3.11 shows the mapping from a random number to the number of axles per vehicle; the transformation yielding the random interarrival times is given by

$$T = -1.2\ln(R),$$

where R is a random number and T is the interarrival time.

We now have the necessary information to begin the simulation. The random numbers in Table C.6 will again be used to simulate the random

TABLE 3.11 Procedure for determining the number of axles per vehicle.

AXLES PER VEHICLE	
Random Number Range	*Number of Axles*
(0.0, 0.75]	2
(0.75, 0.85]	3
(0.85, 0.92]	4
(0.92, 1.0]	5

phenomenon for this situation. This time we roll four dice to determine the starting row. The roll gives 19, so we turn to the nineteenth row of Table C.6 and build Table 3.12.

The result of this simulation produces an estimate of $\frac{38}{3} = 12\frac{2}{3}$ axles per 5-minute interval. Since only three 5-minute intervals were simulated, it should be obvious that this estimate should be taken with a great deal of skepticism; however, for purposes of illustration, the 15-minute time span should be sufficient.

Before closing this example, it may be helpful to discuss the type of process that was just simulated. As mentioned briefly in Chapter 1, a Poisson process satisfies the following two conditions: (1) The times between arrivals form a sequence of independent, identically distributed exponential random variables and (2) the arrival times refer to the arrival of single units. Thus, the arrival of vehicles forms a Poisson process. A relaxation of condition (2) yields a compound Poisson process. That is, a compound Poisson process satisfies (1) the times between arrivals form a sequence of independent, identically distributed exponential random variables and (2) the number of units arriving at each arrival time forms a sequence of independent, identically distributed random variables. Thus, the arrival process of axles is a compound Poisson process.

EXAMPLE 3.7 **Nonhomogenous Poisson Process.** A bank has decided to increase its drive-in window capacity, and therefore desires to model the facility. A first step in modeling the drive-in bank facility is to simulate the arrival process. The drive-in windows open at 7:30 A.M. during the week. The arrival process is a Poisson process with a varying arrival rate. For the first 15 minutes (from

TABLE 3.12 Simulation of the axle arrival process.

Random Number	*Interarrival Time*	*Arrival Time*	*Random Number*	*Number of Axles*	*Cumulative Number*
0.2330	1.75	1.75	0.8310	3	3
0.8647	0.17	1.92	0.2143	2	5
0.5414	0.74	2.66	0.0214	2	7
0.3664	1.20	3.86	0.6326	2	9
0.7129	0.41	4.27	0.1671	2	11
0.2222	1.81	6.08	0.9869	5	16
0.9278	0.09	6.17	0.1854	2	18
0.3735	1.18	7.35	0.8792	4	22
0.9019	0.12	7.47	0.4779	2	24
0.1472	2.30	9.77	0.3953	2	26
0.5929	0.63	10.40	0.6838	2	28
0.4449	0.97	11.37	0.2235	2	30
0.3689	1.20	12.57	0.5881	2	32
0.9092	0.11	12.68	0.1110	2	34
0.3456	1.27	13.95	0.2000	2	36
0.7063	0.42	14.37	0.3995	2	38
0.2013	1.92	16.29			

7:30 A.M. until 7:45 A.M.), the mean arrival rate is 10 cars per hour; for the next 15 minutes (from 7:45 A.M. until 8:00 A.M.), the mean arrival rate is 15 cars per hour, and then the mean arrival rate returns to 10 cars per hour for the next 30 minutes (from 8:00 A.M. until 8:30 A.M.). As the previous example illustrates, simulating this process involves simulating the exponential interarrival times. During the first interval of time, the interarrival times average 6 minutes; from 7:45 A.M. until 8:00 A.M., the interarrival times average 4 minutes; then starting again at 8:00 A.M., the average returns to 6 minutes.

The main difficulty in simulating this process is handling the interarrival time that crosses the boundary at 7:45 A.M. and at 8:00 A.M.. For example, say the current time is 7:43 A.M., and we generate an interarrival time using an exponential distribution with a mean of 6 minutes. Assume this (randomly generated) time turned out to be 6.3 minutes. This would imply that most of the time interval occurred in the 7:45 to 8:00 time period, in which case an average of 4 minutes should have been used instead of the 6 minutes that was used. To handle an interarrival time that crosses a boundary we do the following: Let λ_1 be the mean arrival rate during the time interval $(0, t_1)$, and let λ_2 be the mean arrival rate during the time interval (t_1, t_2). Assume the current time is in the first time interval and is denoted by t. Furthermore let $T = (-1/\lambda_1)\ln(R)$; that is, T is an exponentially generated interarrival time. Now assume that $t + T > t_1$; that is, although T was generated in the first interval, it causes the time to cross into the second interval. The idea is to take the portion of time that is within the second interval (namely, $t + T - t_1$) and factor it up or down according to the new rate. Thus the new time would be

$$t_{\text{new}} = t_1 + \frac{\lambda_1}{\lambda_2}(t + T - t_1).$$

To continue our illustration using the numbers at the beginning of this paragraph, we first observe that the 6.3 minutes includes 4.3 minutes of time into the second interval. The 4.3 minutes is multiplied by the ratio $\frac{4}{6}$ to obtain 2.9, yielding the next arrival time at 7:47.9 (i.e., 2.9 minutes past 7:45).

Table 3.13 presents a simulation of the arrival process for the first 30 minutes. (Notice that the fourth column of the table gives the time in terms of cumulative amount since startup; whereas the first column gives the time in terms of the "time of day.") The simulation crosses two boundaries, and these points are indicated by the superscript "*" on the interarrival times. In particular notice the calculations at time 7:40.72. The random number 0.2143 was generated, its natural logarithm was taken and multiplied by 6, yielding an interarrival time of 9.24. The current time is 10.72 minutes from the start, and we add 9.24 minutes to that time and obtain 19.96 minutes. Notice that the 19.96 minutes is recorded under the column with a heading of "Potential Arrival Time." Since it crosses a boundary, the time must be modified before it can be recorded as the actual arrival time. In particular, the 9.24 minutes includes 4.96 minutes into the next time interval; therefore, the modified

TABLE 3.13 Simulation of nonhomogenous Poisson process.

Current Time	*Random Number*	*Interarrival Time*	*Potential Arrival Time*	*Amount into Next Interval*	*Arrival Time*
7:30.00 A.M.	0.2330	8.74	8.74	—	8.74
7:38.74	0.8310	1.11	9.85	—	9.85
7:39.85	0.8647	0.87	10.72	—	10.72
7:40.72	0.2143	9.24*	19.96	4.96	18.31
7:48.31	0.5415	2.45	20.76	—	20.76
7:50.76	0.0214	15.38*	36.14	6.14	39.21
8:09.21					

interarrival time will be $4.96 \times \frac{4}{6} = 3.31$ minutes into the second interval yielding the next time being 18.31 minutes after startup, or 7:48.31 A.M.. The second "crossover" occurs when the current time is 7:50.76. In this case, the amount of time into the next interval is 6.14 minutes; therefore, the modified time will be $6.14 \times \frac{6}{4} = 9.21$ minutes into the next time interval, yielding the next arrival time of 8:09.21.

EXAMPLE 3.8 **Nonhomogenous Poisson Process—Revisited.** Let us revisit the drive-in bank example with a slightly different arrival process. Assume again that the drive-in windows open at time zero and cars arrive according to a nonhomogenous Poisson process. Whereas in the previous example, we assumed that the mean arrival rate was constant over different intervals; we now assume that the mean arrival rate is described by a continuous function. In particular, assume that for the first hour, the mean arrival rate increases linearly from 10 per hour to 15 per hour. Thus, we have a rate function defined, for $t \leq 60$, by

$$\lambda(t) = 10 + t/12,$$

where t is in minutes and $\lambda(\cdot)$ is in cars per hour.

The concept for simulating this process is to generate arrivals at the maximum rate and then "thin" the process by removing some of the arrivals. In particular, let λ^* denote the maximum arrival rate. If an arrival (generated using λ^*) occurs at time t, then it is accepted with a probability equal to $[\lambda(t)/\lambda^*]$ and rejected with a probability equal to $1 - [\lambda(t)/\lambda^*]$. When an arrival is rejected, the clock is still updated, but an arrival event is not recorded. Table 3.14 shows the results of a simulation that generated arrivals for the first 32.56 minutes. Notice that for each time, two random numbers are generated: a random number that will produce an interarrival time and a random number that will be used to test the interarrival time. Table 3.14 starts at time 0 and the first random number results in a "potential" arrival time of 5.83. At that arrival time, we have $\lambda(5.83) = 10.4858$; therefore, the first arrival is accepted with probability $10.4858/15 = 0.699$. Because the second

TABLE 3.14 Simulation of nonhomogenous Poisson process using a continuous rate function.

Current Time	*Random Number*	*Interarrival Time*	*Arrival Time*	*Critical Number*	*Random Number*	*Event Occur?*
0.00	0.2330	5.83	5.83	0.699	0.8310	No
5.83	0.8647	0.58	6.41	0.702	0.2143	Yes
6.41	0.5415	2.45	8.86	0.716	0.0214	Yes
8.86	0.3664	4.02	12.88	0.738	0.6326	Yes
12.88	0.7129	1.35	14.23	0.746	0.1671	Yes
14.23	0.2222	6.02	20.25	0.779	0.9869	No
20.25	0.9278	0.30	20.55	0.781	0.1854	Yes
20.55	0.3735	3.94	24.49	0.803	0.8792	No
24.49	0.9019	0.41	24.90	0.805	0.4779	Yes
24.90	0.1472	7.66	32.56	0.848	0.3953	Yes
32.56						

random number is greater than 0.699, the arrival event occurring at time 5.83 is rejected. Notice that in the first 32.56 minutes, a total of 10 arrivals were generated, but the simulation only has 7 arrival events actually occurring, with the first interarrival time being 6.41 minutes. ■

3.2.4 Random Variates from Empirical Distributions

Sometimes data are available to approximate a distribution function and so a form of an empirical function is needed. For example, suppose we need to reproduce a continuous random variable we know is always between 1 and 4. Furthermore, some data have been collected, and we know that 25% of the data are between 1 and 1.5, 50% of the data are between 1.5 and 3.5, and 25% are between 3.5 and 4. In such a situation we use a piecewise-linear function containing three segments to approximate the (unknown) cumulative distribution function. The application of the inverse transformation method is illustrated in Figure 3.4. Notice from the figure that the cumulative distribution function is graphed, then a random number is created on the y-axis, and finally the random variate of interest is obtained by an inverse mapping from the y-axis to the x-axis.

As another example, suppose we need to simulate a continuous random variable and all we have are the following ten data points: 5.39, 1.9, 4.62, 2.71, 4.25, 1.11, 2.92, 2.83, 1.88, 2.93. All we know other than the ten points is that the random variable has some (unknown) upper and lower limits that are positive. Our procedure will be the same as that illustrated in the previous example; that is, a cumulative distribution function will be drawn using a piecewise-linear approximation and then the inverse transformation method will be used to go from a random number to the random variate. The distribution function should have the property that 10% of the generated points are centered around 5.39, 10% of the generated points are centered

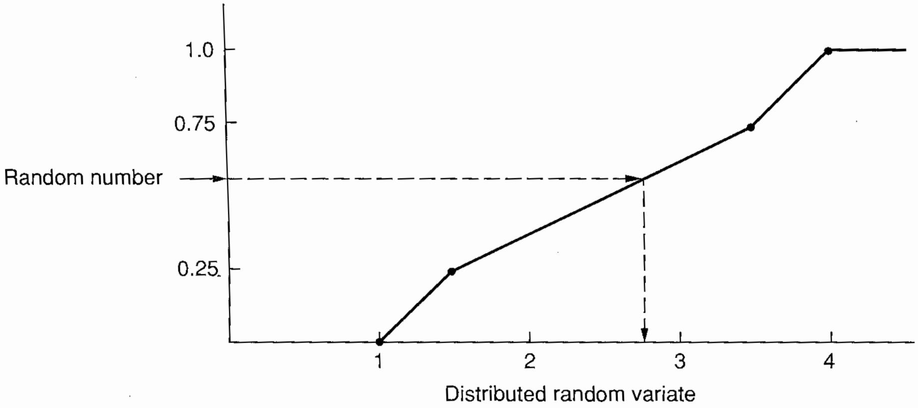


FIGURE 3.4 Mapping used to generate a random variate from a distribution based on experimental data.

around 1.9, etc. To accomplish this, the first step is to order the data, and then calculate the midpoints between adjacent values as shown in the following list.

Data Point	*Midpoint*
1.11	
1.88	1.495
1.90	1.890
2.71	2.305
2.83	2.770
2.92	2.875
2.93	2.925
4.25	3.590
4.62	4.435
5.39	5.005

In developing a distribution function based on these data points, we refer to Figure 3.5. Because there were ten data points, there are ten "cells" with each cell having as its midpoint one of the data points. The major difficulty is the size of the first and last cell; that is, establishing the upper and lower limits. The best way to establish these limits would be through an understanding of the physical limits of whatever it is we are simulating; however, if that cannot be done then one procedure is simply to assume that the lower and upper data points (i.e., 1.11 and 5.39, respectively) are the midpoints of their cells. In other words, the lower and upper limits are established to make this true. Thus, the lower limit is 0.725, and the upper limit is 5.775. [Note that $1.11 = (0.725 + 1.495)/2$ and $5.39 = (5.005 + 5.775)/2$.]

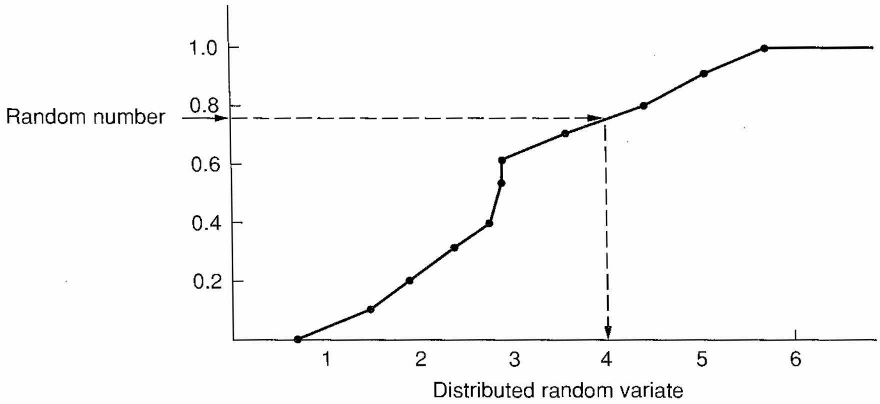


FIGURE 3.5 Mapping used to generate a random variate from a distribution based on ten data points.

The empirically based distribution function increases 0.10 over each cell because there were ten points. In general, if a distribution function is built with n data points, then it would increase $1/n$ over each data cell.

▶ *Suggestion: Do Exercises 3.10–3.14.*

3.3 EXERCISES

3.1 A door-to-door salesman sells pots and pans. He only gets into 50% of the houses that he visits. Of the houses that he enters, $\frac{1}{6}$ of the householders are still not interested in purchasing anything, $\frac{1}{2}$ of them end up placing a \$60 order, and $\frac{1}{3}$ of them end up placing a \$100 order. Estimate the average sales receipts per house visit by simulating 25 house visits using a die. Calculate the theoretical value and compare it with the estimate obtained from your simulation.

3.2 Simulate playing the game of Example 3.2, where the contestant uses the "switch" policy.

(a) Which policy would you recommend?

(b) Can you demonstrate mathematically that your suggested policy is the best policy?

3.3 Given the sequence of three uniform random numbers, 0.15, 0.74, 0.57, generate the corresponding sequences of random variates

(a) from a discrete uniform distribution varying between 1 and 4.

(b) for a discrete random variable, with the probability that it is equal to 1, 2, 3, 4 being 0.3, 0.2, 0.1, 0.4, respectively.

3.4 A manufacturing process has a defect rate of 20% and items are placed in boxes of five. An inspector samples two items from each box. If one or both of the selected items are bad, the box is rejected. Simulate this process to answer the following question: If a customer orders ten boxes, what is the expected number of defective items the customer will receive?

3.5 A binomial random variate with parameters n and p can be obtained by adding n Bernoulli random variates with parameter p. Verify this by writing a computer code that does the following: (1) Generates random numbers (see Appendix B.3), (2) generates four independent Bernoulli random variates with parameter $p = 0.35$ (i.e., if X is Bernoulli, then $X = 1$ if $R \leq p$, and $X = 0$ if $R > p$, where R is a random number), (3) sums the four Bernoulli random variates to obtain a binomial random variate, and (4) estimates the probability mass function for a binomial distribution with $n = 4$ and $p = 0.35$ based on repeated generations of the binomial random variates.

3.6 Consider a Markov chain with state space $\{a, b, c\}$ and with the following transition matrix:

$$\boldsymbol{P} = \begin{bmatrix} 0.35 & 0.27 & 0.38 \\ 0.82 & 0.00 & 0.18 \\ 1.00 & 0.00 & 0.00 \end{bmatrix}.$$

(a) Given that the Markov chain starts in state b, estimate through simulation the expected number of steps until the first return to state b.

(b) The expected return time to a state should equal the reciprocal of the long-run probability of being in that state. Estimate through simulation and analytically the steady-state probability and compare it to your answer to part (a). Explain any differences.

3.7 Use simulation to estimate the answer for part (b) of the problem contained in Exercise 2.6.

3.8 Consider a Markov chain with state space $\{a, b, c\}$ and with the following transition matrix:

$$\boldsymbol{P} = \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.2 & 0.7 \\ 0.8 & 0.0 & 0.2 \end{bmatrix}.$$

Each visit to state a results in a profit of \$5, each visit to state b results in a profit of \$10, and each visit to state c results in a profit of \$12. Write a computer program that will simulate the Markov chain so that an estimate of the expected profit per step can be made. Assume that the chain always starts in state a. The simulation should involve accumulating the profit from each step; then the estimate per simulation run is the cumulative profit divided by the number of steps in the run.

(a) Let the estimate be based on ten replications, where each replication has 25 steps. The estimate is the average value over the ten replicates. Record both the overall average and the range of the averages.

(b) Let the estimate be based on ten replications, where each replication has 1000 steps. Compare the estimates and ranges for parts (a) and this part and explain any differences.

3.9 Consider the problem in Exercise 2.8, part (d). Assume that management wants to recover the initial investment of $100,000 in 3 weeks. Write a computer simulation to estimate the probability that it will take longer than 3 weeks to recover the $100,000.

3.10 Given the sequence of four uniform random numbers, 0.23, 0.74, 0.57, 0.07, generate the corresponding sequences of random variates

(a) from an exponential distribution with a mean of 4.

(b) from a Weibull distribution with scale parameter 0.25 and shape parameter 2.5.

(c) from a normal distribution with a mean of 4 and variance of 4.

(d) from a continuous random variable designed to reproduce the randomness observed by the following data: 4.5, 12.6, 13.8, 6.8, 10.3, 12.5, 8.3, 9.2, 15.3, 11.9, 9.3, 8.1, 16.3, 14.0, 7.3, 6.9, 10.5, 12.3, 9.9, 13.6.

3.11 A mail-order service is open 24 hours per day. Operators receive telephone calls according to a Poisson process. Simulate the process and determine the expected volume (in terms of dollars) of sales during the 15-minute interval from 12:00 noon until 12:15 pm. (Assume each day is probabilistically similar to all other days.)

(a) The Poisson arrival process is homogenous with a mean rate of 50 calls per hour. Furthermore, half of the calls do not result in a sale, 30% of the calls result in a sale of $100, and 20% of the calls result in a sale of $200.

(b) Since more people tend to call on their lunch break, the mean arrival rate slowly increases at the start of the noon hour. For the five-minute interval from 12 noon until 12:05 pm, the mean arrival rate is 40 calls per hour; from 12:05 pm until 12:10 pm, the rate is 45 calls per hour; and from 12:10 pm until 12:50 pm, the rate is 50 calls per hour. The probabilities and amounts of a sale are the same as in part (a).

(c) The arrival rate of calls is approximated by a linear function that equals 40 calls per hour at noon and equals 50 calls per hour at 12:15 pm. The rate is then constant for the next 30 minutes. The probabilities and amounts of a sale are the same as above.

(d) The arrival process is homogenous with a mean rate of 50 calls per hour; however, instead of exponential times, the interarrival times have a type-3 Erlang distribution. The probabilities and amounts of sale are the same as above.

3.12 Write a computer program to produce random variates according to a continuous uniform distribution between 3 and 8. Generate 1000 random variates and compare the sample mean and variance of the generated data with the known mean and variance of the uniform distribution. Do you trust your random variate generator?

3.13 Write a computer program to produce random variates according to a Weibull distribution where the mean and standard deviation are input constants. Generate 1000 random variates and compare the sample mean and standard deviation of the generated data with the input values. Do you trust your random variate generator? (Note, a search routine will have to be written to obtain the scale and shape parameters. See Appendix B.2 for evaluating the gamma function.)

3.14 Write a computer program that will select 100 steel rods from the shipment of rods described in Exercise 1.25. How many rods from your randomly selected sample are outside of the specifications?