- Use Random Numbers to Generate Arrival and Service Time with an example

Example 2: Random Service Times An automated telephone information service spends either 3, 6, or 10 minutes with each caller. The proportion of calls for each service length is 30%, 45%, and 25%, respectively. We want to simulate these service times in a spreadsheet. Our actual purpose is to learn how to generate random samples from a discrete distribution, in preparation for the queueing, inventory, and other examples to follow. This also is an example of a Monte Carlo simulation. Table 2 shows the input specification and a portion of the simulation table (the first 4 callers), taken from the spreadsheet model, “Example2.2ServiceTimes.xls”. The input specification defines the probabilities for the service times, and also includes the cumulative probabilities. Cumulative probabilities always increase to 1.0; as we shall see, the generation method uses the cumulative probabilities. The method always starts with a random number and ends with the desired value, in

Table 2 Input Specification and Simulation Table for Service Times

	A	B	C	D
4		Service Time	Probability	Cumulative Probability
5				
6				
7		3	0.30	0.30
8		6	0.45	0.75
9		10	0.25	1.00
10				
11		Number of Callers=		25
12		Simulation Table		
13		Step	Activity	
14				
15		Caller	Service Time	
16		1	6	
17		2	6	
18		3	10	
19		4	6	

this case, a random service time with the specified distribution. The simulation table in the spreadsheet shows the resulting frequency distribution (or histogram) of the generated service times for 25 callers.

Figure 1 illustrates how to transform a random number to a service time. Think of it as throwing a special dart at a special dart board; the dart will always land somewhere on the unit interval, with equal probability of landing at any of the points on the line between 0 and 1. The subinterval where the dart lands determines the value generated.

In Figure 1, the probabilities (at the top) are represented by non-overlapping segments on a unit interval, and must add up to 1. The cumulative probabilities (just below the line) are represented by points on the line. The arrows represent the transformation. The algorithm works as follows: First, choose a random number, say R. If R is in the first interval, that is, if R is less than or equal to 0.30, the procedure generates a sample of 3 minutes for service time. Similarly, if R is between 0.30 and 0.75, the service time is 6 minutes, and finally, if R is greater than 0.75, then the service time is 10 minutes. This algorithm easily generalizes to any discrete distribution.

To illustrate the procedure, we first generate 5 samples of random numbers, either in Excel (using `RAND` or `Rnd01` or any other random number generator), or take 5 samples from a table of random numbers such as Table A.1 in the Appendix. Suppose the 5 samples are:

```
0.9871    0.0226    0.0008    0.2128    0.8586
```

After the transformation illustrated in Figure 1, the resulting service times are:

```
10        3         3         3         10
```

In a small sample (here, size 5), we cannot expect the observed frequencies of occurrence of each value to be close to the probabilities; however, with a large enough sample, the sample

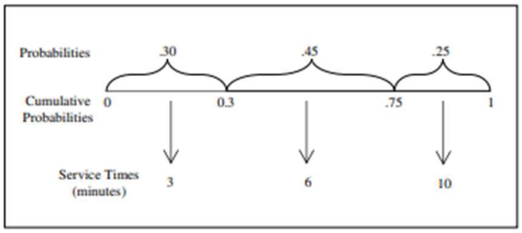

Figure 1 "Random Dart" Transformation from Random Number to Service Time.

frequencies should be fairly close to the probabilities. You can experiment with the spreadsheet example, "Example2.2ServiceTimes.xls", to see how close or far small samples can be from the assumed probabilities.

In Excel, we might think we could implement this transformation, at least for distributions with a small number of values (here, 3), with a nested IF worksheet function, as in:

```
=IF(Rnd01()<=0.30,3,IF(Rnd01()<=0.75,6),10)     (WRONG !!)
```

This attempt at a solution introduces a grave error. Each reference to the `Rnd01()` function generates a new random number; the transformation procedure depends on using just one random number. (You may want to modify the spreadsheet solution in "Example2.2ServiceTimes.xls" with this incorrect solution to see the resulting probabilities.) This approach can be modified to use two cells, the first to hold the result from `Rnd01`, the second with a modified `IF` statement.

A better approach is to use the VBA function called `DiscreteEmp()` (for discrete empirical, meaning defined by data) supplied with all the examples. To see how this function is used, see the spreadsheet solution, "Example2.2ServiceTimes.xls"; a typical cell for generating a random service time appears as follows:

```
=DiscreteEmp($D$7:$D$9, $B$7:$B$9)
```

where $D$7:$D$9 is the range of cells containing the cumulative probabilities and $B$7:$B$9 is the range of cells containing the desired values for service times. Discrete means that only the values listed will be generated. Each of the two worksheet ranges must be a column vector (meaning all data in one column), with each having the same length, that is, the same number of cells. The "$" in each address in front of each row and column indicates an absolute address rather than a relative address; this makes it easier to copy and paste the formula into all the cells in the "Service Time" column after typing it into just one cell.)

The method used here with the supplied VBA function, `DiscreteEmp()`, easily generalizes to most of the examples (and exercises) in the chapter that use simple discrete distributions for model inputs. To use `DiscreteEmp()`, supply two ranges in the worksheet, one for cumulative probabilities and the second for the desired values. It works with any number of values in the distribution.

In future examples, keep Figure 1 in mind; it illustrates exactly how `DiscreteEmp()` works. Although its VBA implementation involves a loop and may appear complex to a non-programmer, its basic logic is the simple logic of Figure 1 — the simple logic of throwing a dart at a special dart board, the unit interval, and a transformation.

- Monte Carlo Simulation single server queuing model having Inter-arrival and Service times Constant

Example 3: Random Arrival Times Telephone calls to the telephone information service, where service times are defined in Example 2, occur at random times defined by a discrete distribution for which the interarrival times have values 1, 2, 3, or 4 minutes, all with equal probability. Our purpose here is to show how to generate both interarrival times and arrival times. Since this example has one event, namely, the arrival event, it is our first example of a dynamic, event-based model (despite its simplicity). Dynamic simply means time-based, with system state changing over time; dynamic, event-based means that the model tracks the progression of event occurrences over time. The interarrival times can be generated randomly

Example 3: Random Arrival Times

Telephone calls to the telephone information service, where service times are defined in Example 2, occur at random times defined by a discrete distribution for which the interarrival times have values 1, 2, 3, or 4 minutes, all with equal probability. Our purpose here is to show how to generate both interarrival times and arrival times.

Since this example has one event, namely, the arrival event, it is our first example of a dynamic, event-based model (despite its simplicity). *Dynamic* simply means time-based, with system state changing over time; dynamic, event-based means that the model tracks the progression of event occurrences over time.

The interarrival times can be generated randomly in exactly the same manner as done in Example 2 for service times. The spreadsheets, however, supply a second VBA function to simplify random generation from a discrete uniform distribution. *Uniform* means that all the values have equal probabilities; in this example, all the interarrival times may have one of the values 1, 2, 3, or 4 minutes, each occurring with probability 0.25. For such discrete uniform distributions, the spreadsheets supply the VBA function, `DiscreteUniform()`, which takes two arguments: either the cell containing the minimum value followed by the cell containing the maximum value of the desired range of integers, or the actual low and high values, as in:

```
= DiscreteUniform($G$5,$G$6)
```

or

```
= DiscreteUniform(1,4)
```

The `DiscreteUniform(low, high)` function is implemented by a single line of VBA code, as follows:

```
DiscreteUniform = low + Int((high - low + 1)*Rnd())
```

which becomes, for this example:

```
DiscreteUniform = 1 + Int(4 * Rnd())
```

where `Rnd()` generates a random number between 0 and 1 (in VBA, it is always less than one but may equal zero), and Int() is a VBA function that truncates (rounds down) its argument. (Note that 4 * `Rnd()` generates real values between 0.0 and 3.99999, so that truncating and then adding 1 yields the desired range 1 to 4. Why do we expect equal probabilities?)

Table 3 shows the input specification for interarrival times and a portion of the simulation table, taken from the spreadsheet model in "Example2.3ArrivalTimes.xls". As shown in Step 1 (row 14), we assume that the first arrival occurs at simulation time 0. (The assumed arrival time for the first customer is model-dependent; it could be any other specified time, constant or random.) The interarrival times are randomly and independently generated using the `DiscreteUniform(1, 4)` function. The arrival times are computed by a simple formula, by adding the current customer's interarrival time to the previous customer's arrival time. (You should verify the spreadsheet formula for arrival times.)

Table 3 Input Specification and Simulation Table for Arrival Times.

	A	B	C	D
4		Interarrival Times		
5		(minutes)		
6		Minimum	1	
7		Maximum	4	
8				
9		Number of Callers=		25
10		Simulation Table		
11		Step	Activity	Clock
12			Interarrival	
13		Caller	Time	Arrival Time
14		1		0
15		2	2	2
16		3	1	3
17		4	3	6

Example 3 is the first that contains an event, namely, the arrival event, and the event time associated with that event, namely, the CLOCK time for the time of arrival. Note that Table 3 distinguishes the activity times from the CLOCK times.

Example 5: The Grocery Checkout, a Single-Server Queue A small grocery store has one checkout counter. Customers arrive at the checkout counter at random times that range from 1 to 8 minutes apart. We assume that interarrival times are integer-valued with each of the 8 values having equal probability; this is a discrete uniform distribution, as shown in Table 9. The service times vary from 1 to 6 minutes (also integer-valued), with the probabilities shown in Table 10. Our objective is to analyze the system by simulating the arrival and service of 100 customers and to compute a variety of typical measures of performance for queueing models.

Example 5: The Grocery Checkout, a Single-Server Queue
A small grocery store has one checkout counter. Customers arrive at the checkout counter at random times that range from 1 to 8 minutes apart. We assume that interarrival times are integer-valued with each of the 8 values having equal probability; this is a discrete uniform distribution, as shown in Table 9. The service times vary from 1 to 6 minutes (also integer-valued), with the probabilities shown in Table 10. Our objective is to analyze the system by simulating the arrival and service of 100 customers and to compute a variety of typical measures of performance for queueing models.

In actuality, 100 customers may be too small a sample size to draw reliable conclusions. Depending on our objectives, the accuracy of the results may be enhanced by increasing the sample size (number of customers), or by running multiple trials (or replications). A second issue is that of initial conditions. A simulation of a grocery store that starts with an empty system may or may not be realistic unless the intention is to model the system from startup or to model until steady-state

Table 9 Distribution of Time Between Arrivals

	G	H
4	Interarrival Times	
5	(minutes)	
6	Minimum	1
7	Maximum	8

Table 10 Distribution of Service Times

	A	B	C	D
4		Service		Cumulative
5		Times	Probability	Probability
6		(Minutes)		
7		1	0.10	0.10
8		2	0.20	0.30
9		3	0.30	0.60
10		4	0.25	0.85
11		5	0.10	0.95
12		6	0.05	1.00

operation is reached. Here, to keep calculations simple, the starting conditions are an empty grocery, and any concerns are overlooked.

How to generate service times and arrival times from the service time and interarrival time distributions has been covered in Sections 1.5 and 1.6, respectively; the same methods apply here and in subsequent examples. For manual simulations, use the random numbers in Table A.1 in the Appendix. It is good practice to start at a random position in Table A.1 and proceed in a systematic direction, never reusing the same stream of numbers in a given problem. If the same pattern is used repeatedly, statistical bias or other odd effects could affect the results.

Table 11 shows a portion of the simulation table for the single-channel queue, taken from "Example2.5SingleServer.xls"; it also shows totals and averages for selected columns (above the simulation table). The first step is to initialize the table by filling in cells for the first customer, who is assumed to arrive at time 0 with service beginning immediately. The customer finishes at time 2 minutes (the first random service time). After the first customer, subsequent rows in the table are based on the randomly generated values for interarrival time and service time of the current customer, the completion time of the previous customer, and, for the spreadsheet solution, the simulation logic incorporated in the formulas in the cells of the simulation table.

Continuing, the second customer arrives at time 5 minutes and service begins immediately. Therefore, the second customer has no wait in the queue and the server has 3 minutes of idle time.

Skipping down, we see that the fifth customer arrives at time 16 minutes and finds the server busy. At time 18, the previous customer leaves and the fifth customer begins service, having waited 2 minutes in the queue. This process continues for all 100 customers. The spreadsheet solution in "Example2.5SingleServer.xls" contains the formulas that implement the steps described here. To best understand these formulas, you should first attempt to do a manual simulation of a few steps, figure out the logic, and then study how the formulas implement the logic.

Columns G, I, and J have been added to collect three model outputs, namely, each customer's time in queue and in system, and the server's idle time (if any) between the previous customer's departure and this customer's arrival time. These model outputs are, in turn, used to compute several measures of performance, namely, the customer's average time in queue and in system, and the proportion of time the server is idle.

In order to compute the system performance measures, we can use the calculated responses in Table 11, which shows totals and averages for interarrival times, service times, the time customers

Table 11 Model Responses and Simulation Table (first 11 customers) for the Grocery Store Simulation

	A	B	C	D	E	F	G	H	I	J
15		TOTALS	420		320		163		483	106
16		AVERAGES	4.24		3.20		1.63		4.83	1.07
17			Number of Customers=		100					
18		Simulation Table								
19		Step	Activity	Clock	Activity	Clock	Output	Clock	Output	Output
20–22		Customer	Interarrival Time (Minutes)	Arrival Time	Service Time (Minutes)	Time Service Begins	Waiting Time in Queue (Minutes)	Time Service Ends	Time Customer Spends in System (Minutes)	Idle Time of Server (Minutes)
23		1	0	0	2	0	0	2	2	
24		2	5	5	2	5	0	7	2	3
25		3	5	10	4	10	0	14	4	3
26		4	4	14	4	14	0	18	4	0
27		5	2	16	3	18	2	21	5	0
28		6	8	24	2	24	0	26	2	3
29		7	7	31	3	31	0	34	3	5
30		8	8	39	5	39	0	44	5	5
31		9	5	44	1	44	0	45	1	0
32		10	2	46	6	46	0	52	6	1
33		11	1	47	4	52	5	56	9	0

wait in the queue, the time customers spend in the system, and idle time of the server. These results are taken from the first trial run on the 'One Trial' worksheet of the spreadsheet model when using the default RNG seed (12,345). With these results, we can illustrate a number of typical performance measures for waiting line models:

1. The average waiting time for a customer is 1.63 minutes, computed as follows:

$$\begin{aligned}\text{Average waiting time (minutes)} &= \frac{\text{total time customers wait in queue (minutes)}}{\text{total numbers of customers}}\\ &= \frac{163}{100} = 1.63 \text{ minutes}\end{aligned}$$

2. The probability that a customer has to wait in the queue is 0.46, computed as follows:

$$\begin{aligned}\text{Probability (wait)} &= \frac{\text{numbers of customers who wait}}{\text{total number of customers}}\\ &= \frac{46}{100} = 0.46\end{aligned}$$

3. The server is idle about 25% of the time, computed as follows:

$$\begin{aligned}\text{Probability of idle server} &= \frac{\text{total idle time of server (minutes)}}{\text{total run time of simulation (minutes)}}\\ &= \frac{106}{426} = 0.25\end{aligned}$$

   Therefore, the server is busy about 75% of the time.

4. The average service time is 3.20 minutes, computed as follows:

$$\begin{aligned}\text{Average service time (minutes)} &= \frac{\text{total service time (minutes)}}{\text{total number of customers}}\\ &= \frac{320}{100} = 3.20 \text{ minutes}\end{aligned}$$

   This result can be compared with the expected service time by finding the mean of the service-time distribution, using the equation

$$E(S) = \sum_{s=0}^{\infty} sp(s)$$

   Applying the expected-value equation to the distribution in Table 10 gives

$$\begin{aligned}&\text{Expected service time} =\\ &1(0.10) + 2(0.20) + 3(0.30) + 4(0.25) + 5(0.10) + 6(0.05) = 3.2 \text{ minutes}\end{aligned}$$

   The expected service time is exactly the same as the average service time from this trial of the simulation. In general, this will not be true; here it is a pure coincidence.

5. The average time between arrivals is 4.24 minutes, computed as follows:

$$\begin{array}{c}\text{Average time between}\\ \text{arrivals (minutes)}\end{array} = \frac{\begin{array}{c}\text{sum of all times}\\ \text{between arrivals (minutes)}\end{array}}{\text{number of arrivals} - 1}$$

$$= \frac{420}{99} = 4.24 \text{ minutes}$$

Since the first arrival occurs at time 0, we divide by 99 rather than 100. This result can be compared to the expected time between arrivals by finding the mean of the discrete uniform distribution whose endpoints are $a = 1$ and $b = 8$. The mean is given by

$$E(A) = \frac{a+b}{2} = \frac{1+8}{2} = 4.5 \text{ minutes}$$

The expected time between arrivals is slightly higher than the average. However, as the simulation becomes longer, the average value of the time between arrivals tends to become closer to the theoretical mean, $E(A)$.

6. The average waiting time of those who wait is 3.54 minutes, computed as follows:

$$\begin{array}{c}\text{Average waiting time of}\\ \text{those who wait}\\ \text{(minutes)}\end{array} = \frac{\text{total time customers wait in queue (minutes)}}{\text{total number of customers who wait}}$$

$$= \frac{163}{46} = 3.54 \text{ minutes}$$

Note that this is different from the average waiting time of all customers, many of whom (54 in this simulation) have no wait at all before their service begins. Over all customers, the average waiting time is $\frac{163}{100} = 1.63$ minutes.The number who wait, 46, comes from the simulation table, column G under "Waiting Time in Queue," by counting the number of nonzero waiting times, or alternately from Figure 9, which displays a histogram of the 100 waiting times.

7. The average time a customer spends in the system is 4.83 minutes, computed in two ways. First:

$$\begin{array}{c}\text{Average time customer}\\ \text{spends in the system}\\ \text{(minutes)}\end{array} = \frac{\begin{array}{c}\text{total time customers spend in the}\\ \text{system (minutes)}\end{array}}{\text{total number of customers}}$$

$$= \frac{483}{100} = 4.83 \text{ minutes}$$

The second way is to realize that the following relationship must hold:

$$\begin{array}{c}\text{Average time}\\ \text{customer spends}\\ \text{in the system}\\ \text{(minutes)}\end{array} = \begin{array}{c}\text{average time}\\ \text{customer spends}\\ \text{waiting in the}\\ \text{queue (minutes)}\end{array} + \begin{array}{c}\text{average time}\\ \text{customer spends}\\ \text{in service}\\ \text{(minutes)}\end{array}$$

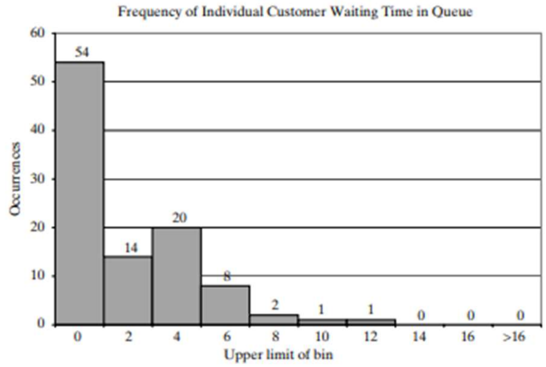

Figure 9 Frequency of waiting time in queue.

From findings 1 and 4, this results in

$$\text{Average time customer spends in the system } = 1.63 + 3.20 = 4.83 \text{ minutes}$$

We are now interested in running an experiment to address the question of how the average time in queue for the first 100 customers varies from day to day. Running the 'One Trial' worksheet once with 100 customers corresponds to one day. Running it 50 times corresponds to using the 'Experiment' worksheet with "Number of Trials" set to 50, where each trial represents one day.

For Example 5, the frequency of waiting time in queue for the first trial of 100 customers is shown in Figure 9. (Note: In all histograms in the remainder of this chapter, the upper limit of the bin is indicated on the legend on the x-axis, even if the legend is shown centered within the bin.) As shown in the histogram, 54% of the customers did not have to wait, and of those who did wait, 34% waited less than four minutes (but more than zero minutes). These results and the histogram in Figure 9 represent the results of one trial, and by themselves do not constitute an answer to our question.

To answer our question, we use the 'Experiment' sheet, setting "Number of Trials" to 50 and click the button labeled 'Reset Seed & Run'. This results in 50 estimates of average time in queue (each averaged over 100 customers). The overall average waiting time over 50 trials was 1.32 minutes. Figure 10 shows a histogram of the 50 average waiting times for the 50 trials. This shows how average waiting time varies day-by-day over a sample of 50 days.

By experimenting with the spreadsheet solution and doing some of the related exercises, you can discover the effects of randomness and of the assumed input data. For example, what if you run 400 trials, or 10 trials, instead of 50? How much does the shape of the distribution in Figure 10 change the number of trials? Why is the range (0 to 12 minutes) in Figure 9 so much wider than the range (0.5 to 3 minutes) in Figure 10?

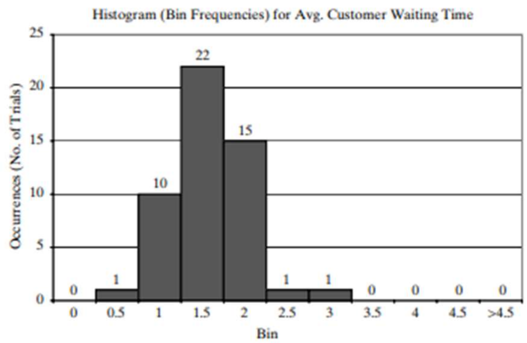

Figure 10 Frequency distribution of average waiting times for 50 trials.

- Monte Carlo simulation technique, determine the average profit of a company

Q10, Chapter 11. Estimation of absolute performance, page 415

A store selling Mother's Day cards must decide 6 months in advance on the number of cards to stock. Reordering is not allowed. Cards cost $0.45 and sell for $1.25. Any cards not sold by Mother's Day go on sale for $0.50 for 2 weeks. However, sales of the remaining cards is probabilistic in nature according to the following distribution: 32% of the time, all cards remaining get sold. 40% of the time, 80% of all cards remaining are sold. 28% of the time, 60% of all cards remaining are sold. Any cards left after 2 weeks are sold for $0.25. The card-shop owner is not sure how many cards can be sold, but thinks it is somewhere (i.e., uniformly distributed) between 200 and 400. Suppose that the card-shop owner decides to order 300 cards. Estimate the expected total profit with an error of at most $5.00. [Hint: Make ten initial replications. Use these data to estimate the total sample size needed. Each replication consists of one Mother's Day.]

10. A store selling Mother's Day cards must decide 6 months in advance on the number of cards to stock. Reordering is not allowed. Cards cost $0.45 and sell for $1.25. Any cards not sold by Mother's Day go on sale for $0.50 for 2 weeks. However, sales of the remaining cards is probabilistic in nature according to the following distribution:

    32% of the time, all cards remaining get sold.

    40% of the time, 80% of all cards remaining are sold.

    28% of the time, 60% of all cards remaining are sold.

    Any cards left after 2 weeks are sold for $0.25. The card-shop owner is not sure how many cards can be sold, but thinks it is somewhere (i.e., uniformly distributed) between 200 and 400. Suppose that the card-shop owner decides to order 300 cards. Estimate the expected total profit with an error of at most $5.00. [*Hint*: Make ten initial replications. Use these data to estimate the total sample size needed. Each replication consists of one Mother's Day.]

Example 7: The News Dealer's Problem A news dealer buys papers for 33 cents each and sells them for 50 cents each. Newspapers not sold at the end of the day are sold as scrap for 5 cents each. Newspapers can be purchased in bundles of 10. Thus, the newsstand can buy 50 or 60 or 70 papers, and so on. The order quantity, Q, is the only policy decision. Unlike some inventory problems, the order quantity Q is fixed since ending inventory is always zero due to scrapping leftover papers.

Solution:

Table 16 Distribution of Daily Newspaper Demand, by Type of Newsday

	B	C	D	E	F	G	H
4	Distribution of Newspapers Demanded						
5	Demand	Demand Probabilities			Cumulative Probabilities		
6		Good	Fair	Poor	Good	Fair	Poor
7	40	0.03	0.10	0.44	0.03	0.10	0.44
8	50	0.05	0.18	0.22	0.08	0.28	0.66
9	60	0.15	0.40	0.16	0.23	0.68	0.82
10	70	0.20	0.20	0.12	0.43	0.88	0.94
11	80	0.35	0.08	0.06	0.78	0.96	1.00
12	90	0.15	0.04	0.00	0.93	1.00	1.00
13	100	0.07	0.00	0.00	1.00	1.00	1.00

There are three types of newsdays: "good", "fair", and "poor", but the news dealer cannot predict which type will occur on any given day. Table 16 gives the distribution of newspapers demanded by type of day. Table 17 provides the distribution for type of newsday.

Our objective is to compute the optimal number of papers the newsstand should purchase (the order quantity, Q). This will be accomplished by simulating demands for 20 days (in the 'One Trial' worksheet) and recording profits from sales each day for a given order quantity. After a trial is completed, total profit (the model response or performance measure) is calculated. To complete the experiment, we use the 'Experiment' sheet to replicate some number of times for a given order quantity (for example, 70 papers). Finally, we vary the order quantity over some reasonable range (40, 50, 60, 70, and so on) and compare different order policies based on total profit over the 20-day period.

Daily profit is given by the following relationship:

$$\text{Profit} = \begin{pmatrix}\text{revenue}\\ \text{from sales}\end{pmatrix} - \begin{pmatrix}\text{cost of}\\ \text{newspapers}\end{pmatrix} - \begin{pmatrix}\text{lost profit from}\\ \text{excess demand}\end{pmatrix} + \begin{pmatrix}\text{salvage from sale}\\ \text{of scrap papers}\end{pmatrix}$$

Since the revenue from sales is 50 cents per paper, and the cost to the dealer is 33 cents per paper, the lost profit from excess demand is 17 cents for each paper demanded that could not be provided.

Table 17 Distribution of Type of Newsday

	J	K	L
4	Type of Newsday		
5	Type	Probability	Cumulative Probability
7	Good	0.35	0.35
8	Fair	0.45	0.80
9	Poor	0.20	1.00

Table 18 Simulation Table for Purchase of 70 Newspapers

	B	C	D	E	F	G	HI	I
16	Simulation Table							
17								
18		Type of		Revenue from	Lost Profit from Excess	Salvage from Sale	Daily	Daily
19	Day	Newsday	Demand	Sales	Demand	of Scrap	Cost	Profit
20	1	Fair	50	$25.00	$0.00	$1.00	$23.10	$2.90
21	2	Fair	50	$25.00	$0.00	$1.00	$23.10	$2.90
22	3	Fair	70	$35.00	$0.00	$0.00	$23.10	$11.90
23	4	Good	80	$35.00	$1.70	$0.00	$23.10	$10.20
24	5	Poor	40	$20.00	$0.00	$1.50	$23.10	-$1.60
25	6	Fair	50	$25.00	$0.00	$1.00	$23.10	$2.90
26	7	Poor	50	$25.00	$0.00	$1.00	$23.10	$2.90
27	8	Fair	70	$35.00	$0.00	$0.00	$23.10	$11.90
28	9	Good	40	$20.00	$0.00	$1.50	$23.10	-$1.60
29	10	Good	100	$35.00	$5.10	$0.00	$23.10	$6.80
30	11	Fair	70	$35.00	$0.00	$0.00	$23.10	$11.90
31	12	Poor	50	$25.00	$0.00	$1.00	$23.10	$2.90
32	13	Fair	50	$25.00	$0.00	$1.00	$23.10	$2.90
33	14	Poor	40	$20.00	$0.00	$1.50	$23.10	-$1.60
34	15	Good	80	$35.00	$1.70	$0.00	$23.10	$10.20
35	16	Good	70	$35.00	$0.00	$0.00	$23.10	$11.90
36	17	Good	80	$35.00	$1.70	$0.00	$23.10	$10.20
37	18	Poor	40	$20.00	$0.00	$1.50	$23.10	-$1.60
38	19	Fair	70	$35.00	$0.00	$0.00	$23.10	$11.90
39	20	Good	100	$35.00	$5.10	$0.00	$23.10	$6.80
40							TOTAL PROFIT =	$114.70

Such a shortage cost is somewhat controversial, but makes the problem much more interesting. The salvage value of scrap papers is 5 cents each.

The simulation table for the decision to purchase 70 newspapers, taken from the solution workbook, "Example2.7NewsDealer.xls", is shown in Table 18. On day 1, the demand is for 50 newspapers, so with 70 available, there are 20 to scrap. The revenue from the sale of the 50 newspapers is $25.00, and the scrap value is $1.00, so the first-day profit is computed as follows:

$$\text{First-day Profit } = \$25.00 - \$23.10 - 0 + \$1.00 = \$2.90$$

The profit from other days is easily computed in the same manner. The profit for the 20-day period is the sum of the daily profits, $114.70, as shown in Table 18. In general, because the results of one day are independent of previous days, inventory problems of this type are much easier than queueing problems to solve in a spreadsheet.

With the 'Experiment' worksheet from "Example2.7NewsDealer.xls", we set up an experiment with 400 trials or replications for the order policy of ordering 70 newspapers. Figure 14 shows the

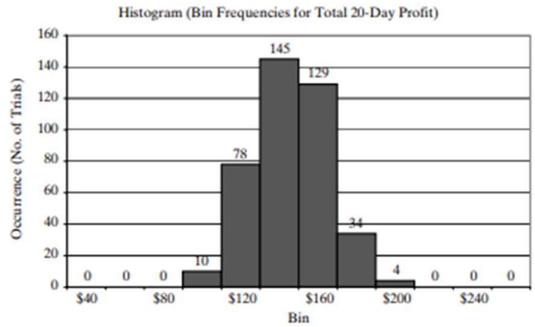

Figure 14 Frequency of total (20-day) profit when news dealer orders 70 papers per day.

result of these 400 trials, each trial being a 20-day simulation run. The average total (20-day) profit was $135.49 (averaged over the 400 trials). The minimum 20-day profit was $86.60 and the maximum was $198.00. Figure 14 shows that only 38 of the 400 trials resulted in a total 20-day profit of more than $160. (Recall that the number beneath a bin is the upper limit of the bin.)

The single trial shown in Table 18 had a profit of $114.70, which is on the low side of the distribution in Figure 14. This illustrates once again that the result for one 20-day trial may not be representative of long-term performance, and shows the necessity of simulating a number of trials as well as conducting a careful statistical analysis before drawing conclusions.

On the 'One Trial' sheet, look how Daily Profit varies when clicking the button 'Generate New Trial' multiple times. The results vary quite a bit both in the histogram entitled 'Frequency of Daily Profit' (showing what happened on each of the 20 days) and in the total profit over those 20 days. Figure 15 shows four typical histograms for daily profit from the first four trials, whose shape varies quite a bit more than the shape for the histogram for total 20-day profit.

Exercise 28 asks you to conduct an experiment to determine the optimum number of newspapers for the news dealer to order. To solve this problem by simulation requires setting a policy of buying a certain number of papers each day and running the experiment for a large number of trials. The policy (number of newspapers purchased) is changed to other values in a reasonable range (say, 40, 50, 60, and so on) and the simulation repeated until the best value is found. To estimate with confidence whether a simulated difference is likely to be real and not a random fluke, requires additional statistical methods.

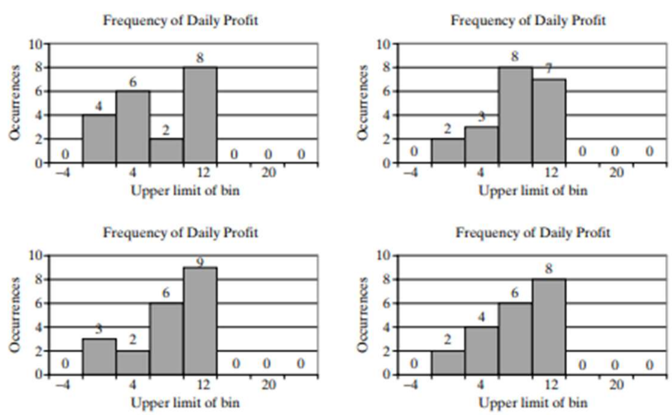

Figure 15 Four typical histograms of daily profit for the News Dealer's Problem.