

L

A

B

# 9 SIMULATION OUTPUT ANALYSIS

*Nothing has such power to broaden the mind as the ability to investigate systematically and truly all that comes under thy observation in life.*

—Marcus Aurelius

In this lab we review the differences between terminating and nonterminating simulations. You will learn how to collect observations of a performance metric from both terminating and nonterminating simulations using ProModel. Key problems of identifying the warm-up period and selecting the run length of a nonterminating simulation are addressed in the lab. We also illustrate how supporting software can be used to help determine if the observations reported by ProModel for a simulation are independent and normally distributed.

## L9.1 Terminating versus Nonterminating Simulations

A terminating simulation has a fixed starting condition and a fixed ending condition (terminating point). For example, a simulation model of a store that opens at 9:00 A.M. begins with a starting condition of no customers in the system and terminates after the store closes at 7:00 P.M. and the last customer is processed. Many service-oriented systems (banks, doctors' offices, post offices, schools, amusement parks, restaurants, and department stores) are analyzed with a terminating simulation. The system returns to a fixed starting condition after reaching a natural terminating event.

A nonterminating simulation has neither a fixed starting condition nor a natural ending condition. The simulation cannot run indefinitely, so we end it after its output reflects the long-term average behavior of the system. A hospital or a 24-hour gas station could be analyzed with a nonterminating simulation depending on the objective of the study. Many systems appear to be terminating but in

reality are nonterminating. Systems that close/terminate but do not go back to a fixed starting condition (empty for example) are usually analyzed with nonterminating simulations. For example, a production system that uses the current day's unfinished work as the next day's starting work is really a nonterminating system. If we compress time by removing the intervals during which the system is closed, it behaves as a continuously operating system with no point at which the system starts anew. In general, most manufacturing systems are nonterminating. The work-in-process inventory from one day is typically carried forward to the next day.

When we report the simulation model's response based on the output from a terminating simulation or a nonterminating simulation, it is important to use interval estimates rather than simple point estimates of the parameter being measured. The advantage of the interval estimate is that it provides a measure of the variability in output produced by the simulation. Interval estimates or confidence intervals provide a range of values around the point estimate that have a specified likelihood of containing the parameter's true but unknown value. For a given confidence level, a smaller confidence interval is considered better than a larger one. Similarly, for a given confidence interval, a higher confidence level indicates a better estimate. To construct the confidence interval, the observations produced by replications or batch intervals need to be independent and identically distributed. Furthermore, the observations should be normally distributed when using the Student's  $t$  distribution to calculate the confidence interval. We can fudge a little on the normality assumption but not on the independence and identically distribution requirements. These topics are elaborated on in Chapter 9.

## L9.2 Terminating Simulation

### Problem Statement

**Spuds-n-More by the Seashore** is a favorite among vacationers at the seaside resort. Spuds-n-More, a takeout fast-food restaurant, specializes in potato foods, serving many different varieties of french fries, waffle fries, and potato logs along with hamburgers and drinks. Customers walking along the boardwalk access the restaurant via a single ordering window, where they place their orders, pay, and receive their food. There is no seating in the restaurant. The interarrival time of customers to Spuds-n-More is exponentially distributed with a mean of six minutes. The activity times are given in Table L9.1. The waiting area in front of the walkup window can accommodate only five customers. If the waiting area is filled

**TABLE L9.1** Process Times at Spuds-n-More by the Seashore

<i>Activity</i>	<i>Process Time (minutes)</i>
Order food	E(2)
Make payment	E(1.5)
Pick up food	E(5)

to capacity when a customer arrives, the customer bypasses Spuds-n-More. Mr. Taylor closes the entryway to the waiting area in front of his restaurant after eight hours of operation. However, any customers already in the waiting area by the closing time are served before the kitchen shuts down for the day.

Mr. Taylor is thinking about expanding into the small newspaper stand next to Spuds-n-More that has been abandoned. This would allow him to add a second window to serve his customers. The first window would be used for taking orders and collecting payments, and the second window would be used for filling orders. Before investing in the new space, however, Mr. Taylor needs to convince the major stockholder of Spuds-n-More, Pritchard Enterprises, that the investment would allow the restaurant to serve more customers per day. The first step is to build a baseline (as-is) model of the current restaurant configuration and then to embellish the model with the new window and measure the effect it has on the number of customers processed per day. Mr. Taylor is interested in seeing a 90 percent confidence interval for the number of customers processed per day in the baseline simulation.

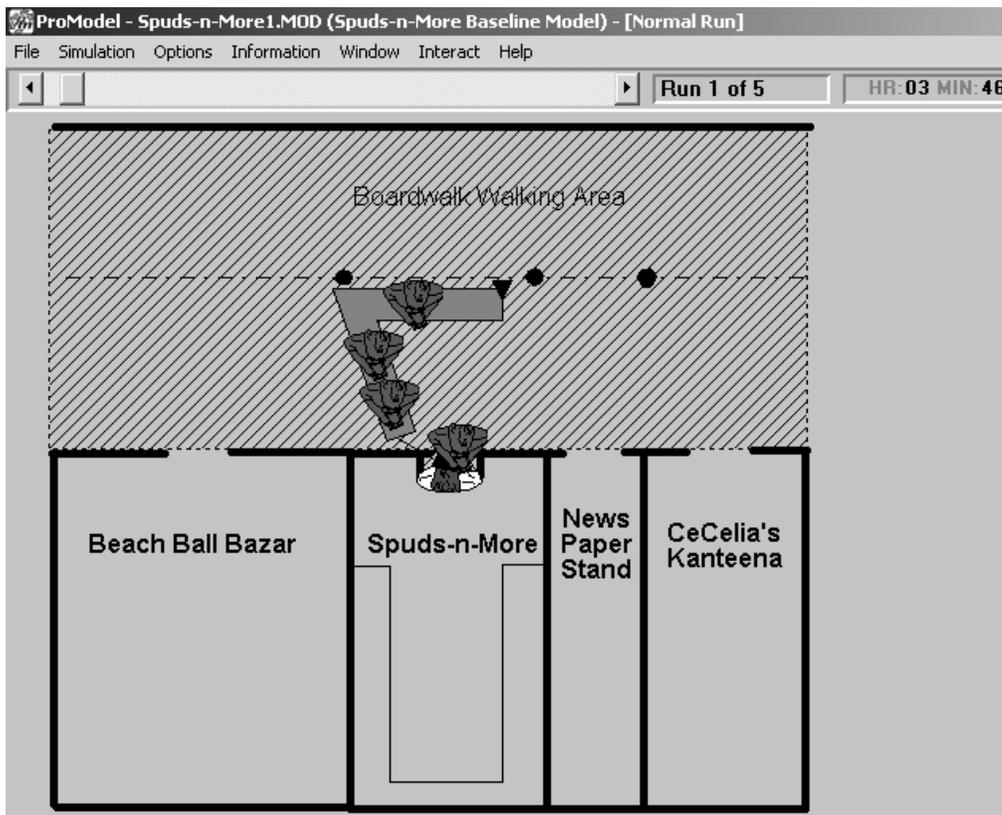
### L9.2.1 Starting and Terminating Conditions (Run Length)

Build the baseline model of the as-is restaurant and conduct a terminating simulation to estimate the expected number of customers served each day. The starting condition for the simulation is that the system begins in the morning empty of customers with the first customer arriving at time zero. Customers are admitted into the restaurant's queue over the eight hours of operation. The simulation terminates after reaching eight hours and having processed the last customer admitted into the queue. The layout of the simulation model is shown in Figure L9.1, and a printout of the model is given in Figure L9.2. Note that the length of the Order\_Q location is 25 feet and the Customer entity travel speed is 150 feet per minute. These values affect the entity's travel time in the queue (time for a customer to walk to the end of the queue) and are required to match the results presented in this lab. The model is included on the CD accompanying the book under file name Lab 9\_2\_1 Spuds-n-More1.MOD.

Let's take a look at the model listing in Figure L9.2 to see how the termination criterion was implemented. The Entry location serves as a gate for allowing customers into the queue until the simulation clock time reaches eight hours. Entities arriving after the simulation clock reaches eight hours are turned away (routed to Exit) by the routing rule User Condition (IF clock(hr) > 8) at the Entry location. Entities arriving within eight hours are routed to the Order\_Q location (see routing rule User Condition IF clock(hr) <= 8). Two global variables are declared to count the number of entities arriving at the Order\_Q location and the number of entities processed at the Order\_Clerk location. See the statement Inc Arrivals at the Order\_Q location and the statement Inc Processed at the Order\_Clerk location. The Inc statement increments the variable's value by one each time it is executed. As the entities that arrive after eight hours of simulation time are turned away to the Exit, the number of entities processed by the Order\_Clerk is compared to the number of entities allowed into the Order\_Q (see Move Logic). If Processed = Arrivals, then all entities that were allowed into the

**FIGURE L9.1**

*Layout of the Spuds-n-More simulation model.*



Order\_Q have been processed and the simulation is terminated by the `stop` statement. Notice that the combined capacity of the Entry and Order\_Q locations is five in order to satisfy the requirement that the waiting area in front of the restaurant accommodates up to five customers.

We have been viewing the output from our simulations with ProModel's new Output Viewer 3DR. Let's conduct this lab using ProModel's traditional Output Viewer, which serves up the same information as the 3DR viewer but does so a little faster. To switch viewers, select Tools from the ProModel main menu bar and then select Options. Select the Output Viewer as shown in Figure L9.3.

### **L9.2.2 Replications**

Run the simulation for five replications to record the number of customers served each day for five successive days. To run the five replications, select Options from under the Simulation main menu. Figure L9.4 illustrates the simulation options set to run five replications of the simulation. Notice that no run hours are specified.

FIGURE L9.2

The simulation model of Spuds-n-More.

```

Time Units: Minutes
Distance Units: Feet
*****
*                               Locations                               *
*****
Name      Cap Units Stats      Rules      Cost
-----
Entry     1    1    None      Oldest, ,
Order_Q   4    1    None      Oldest, FIFO,
Order_Clerk 1    1    None      Oldest, ,
*****
*                               Entities                               *
*****
Name      Speed (fpm) Stats      Cost
-----
Customer  150      None
*****
*                               Processing                              *
*****
Process                                Routing
-----
Entity  Location  Operation      Blk Output  Destination Rule      Move Logic
-----
Customer Entry      1    Customer Order_Q    IF clock(hr) <= 8, 1
Customer Order_Q   1    Customer EXIT      IF clock(hr) > 8      If Processed = Arrivals
Customer Order_Clerk // Order      Then Stop
Customer Order_Clerk // Pay
Customer Order_Clerk Wait e(2) min
Customer Order_Clerk // Pay
Customer Order_Clerk Wait e(1.5) min
Customer Order_Clerk // Pickup
Customer Order_Clerk Wait e(5) min
Customer Order_Clerk Inc Processed      1    Customer EXIT      FIRST 1
*****
*                               Arrivals                               *
*****
Entity  Location Qty Each  First Time Occurrences Frequency Logic
-----
Customer Entry  1    0    inf      E(6) min
*****
*                               Variables (global)                       *
*****
ID      Type      Initial value Stats
-----
Arrivals Integer  0      None
Processed Integer  0      Time Series ← Note that the observation-based
suboption of Time Series was selected.

```

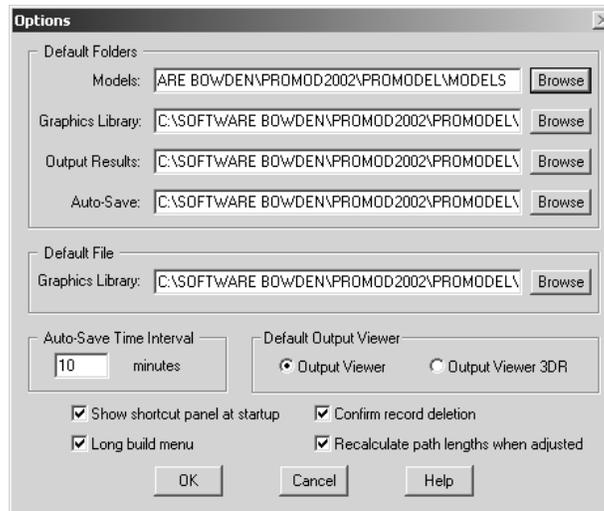
After ProModel runs the five replications, it displays a message asking if you wish to see the results. Answer yes.

Next ProModel displays the General Report Type window (Figure L9.5). Here you specify your desire to see the output results for “<All>” replications and then click on the Options . . . button to specify that you wish the output report to also include the sample mean (average), sample standard deviation, and 90 percent confidence interval of the five observations collected via the five replications.

The ProModel output report is shown in Figure L9.6. The number of customers processed each day can be found under the Current Value column of the output report in the VARIABLES section. The simulation results indicate that the number of customers processed each day fluctuates randomly. The fluctuation may be

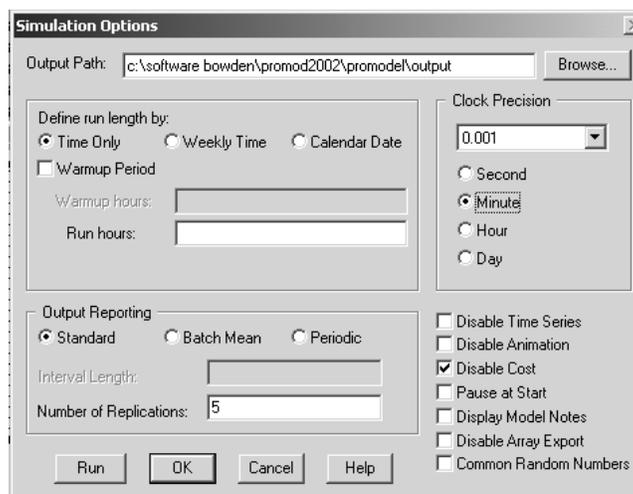
**FIGURE L9.3**

*ProModel's Default Output Viewer set to Output Viewer.*



**FIGURE L9.4**

*ProModel's Simulation Options window set to run five replications of the simulation.*



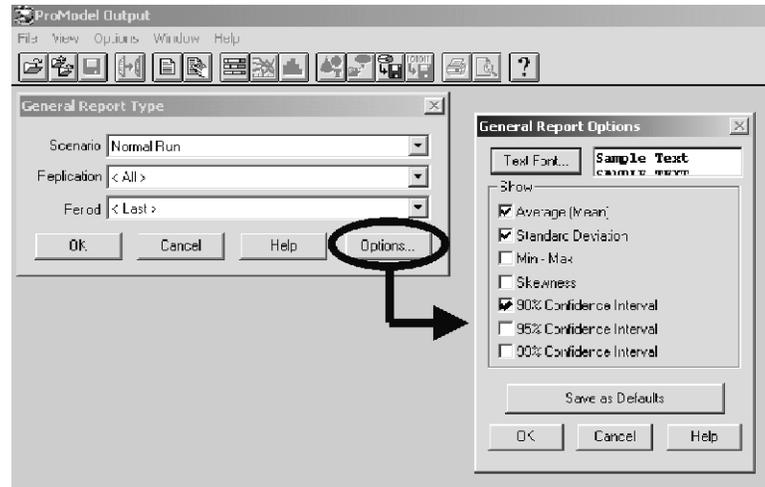
attributed to variations in processing times or customer arrivals. It is obvious that we cannot reliably use the observation from a single day's simulation to estimate the expected number of customers served each day. The results from a single day could produce an estimate as small as 59 or as large as 67. Either value could be a very poor estimate of the true expected number of customers served each day.

The average number of customers processed over the five days is computed as  $\bar{x} = 62.20$  (Figure L9.6). This provides a much better point estimate of the expected daily service rate (number of customers processed each day), but it fails to capture any information on the variability of our estimate. The sample deviation of  $s = 3.27$  customers per day provides that information. The most valuable piece of information is the 90 percent confidence interval computed by ProModel using

Lab 9 Simulation Output Analysis

**FIGURE L9.5**

*ProModel General Report Options set to display the results from all replications as well as the average, standard deviation, and 90 percent confidence interval.*



**FIGURE L9.6**

*ProModel output report for five replications of the Spuds-n-More simulation.*

FAILED ARRIVALS

Entity Name	Location Name	Total Failed	
Customer	Entry	21	(Rep 1)
Customer	Entry	18	(Rep 2)
Customer	Entry	17	(Rep 3)
Customer	Entry	14	(Rep 4)
Customer	Entry	30	(Rep 5)
Customer	Entry	20	(Average)
Customer	Entry	6.123	(Std. Dev.)
Customer	Entry	14.161	(90% C.I. Low)
Customer	Entry	25.838	(90% C.I. High)

VARIABLES ( \* indicates observation based variables)

Variable Name	Total Changes	Average Minutes Per Change	Minimum Value	Maximum Value	Current Value	Average Value	
Processed*	64	8.136	1	64	64	32.5	(Rep 1)
Processed*	67	8.000	1	67	67	34	(Rep 2)
Processed*	61	8.529	1	61	61	31	(Rep 3)
Processed*	59	9.002	1	59	59	30	(Rep 4)
Processed*	60	8.753	1	60	60	30.5	(Rep 5)
Processed*	62.2	8.488	1	62.2	62.2	31.6	(Average)
Processed*	3.271	0.418	0	3.271	3.271	1.635	(Std. Dev.)
Processed*	59.081	8.089	1	59.081	59.081	30.040	(90% C.I. Low)
Processed*	65.318	8.887	1	65.318	65.318	33.159	(90% C.I. High)

the sample mean and sample standard deviation. With approximately 90 percent confidence, the true but unknown mean number of customers processed per day falls between 59.08 and 65.32 customers. These results convince Mr. Taylor that the model is a valid representation of the actual restaurant, but he wants to get a better estimate of the number of customers served per day.

### L9.2.3 Required Number of Replications

The advantage of the confidence interval estimate is that it quantifies the error in the point estimate. A wide confidence interval implies that the point estimate is not very accurate and should be given little credence. The width of the confidence interval is a function of both the variability of the system and the amount of data collected for estimation. We can make the confidence interval smaller by collecting more data (running more replications). However, for highly variable systems, one may need a lot of data to make extremely precise statements (smaller confidence intervals). Collecting the data may be very time consuming if the simulation takes long to run. Hence, we sometimes have to make a judicious choice about the amount of precision required.

The half-width of the 90 percent confidence interval based on the five replications is 3.12 customers (65.32 C.I. High – 62.20 Average). How many additional replications are needed if Mr. Taylor desires a confidence interval half-width of 2.0 customers? Using the technique described in Section 9.2.3 of Chapter 9 we have

$$e = hw = 2.0 \text{ customers}$$

$$\alpha = \text{significance level} = 0.10 \text{ for a 0.90 confidence level}$$

$$n' = \left[ \frac{(Z_{\alpha/2})s}{e} \right]^2 = \left[ \frac{(t_{\infty, \alpha/2})s}{e} \right]^2 = \left[ \frac{(1.645)3.27}{2.0} \right]^2 = 7.23 \text{ replications}$$

The value of 7.23 is appropriately rounded up to 8 replications. The  $n' = 8$  provides a rough approximation of the number of replications that are needed to achieve the requested error amount  $e = 2.0$  customers at the 0.10 significance level. Running an additional three replications of the Spuds-n-More baseline simulation and combining the output with the output from the original five replications produces a sample mean of 61.50 customers and an approximate 90 percent confidence interval of 58.99 to 64.00 customers. You should verify this. Although the confidence interval half-width of 2.51 customers is larger than the target value of 2.0 customers, it is smaller than the half-width of the confidence interval based on five replications. We could keep running additional replications to get closer to the target half-width of 2.0 customers if we like.

### L9.2.4 Simulation Output Assumptions

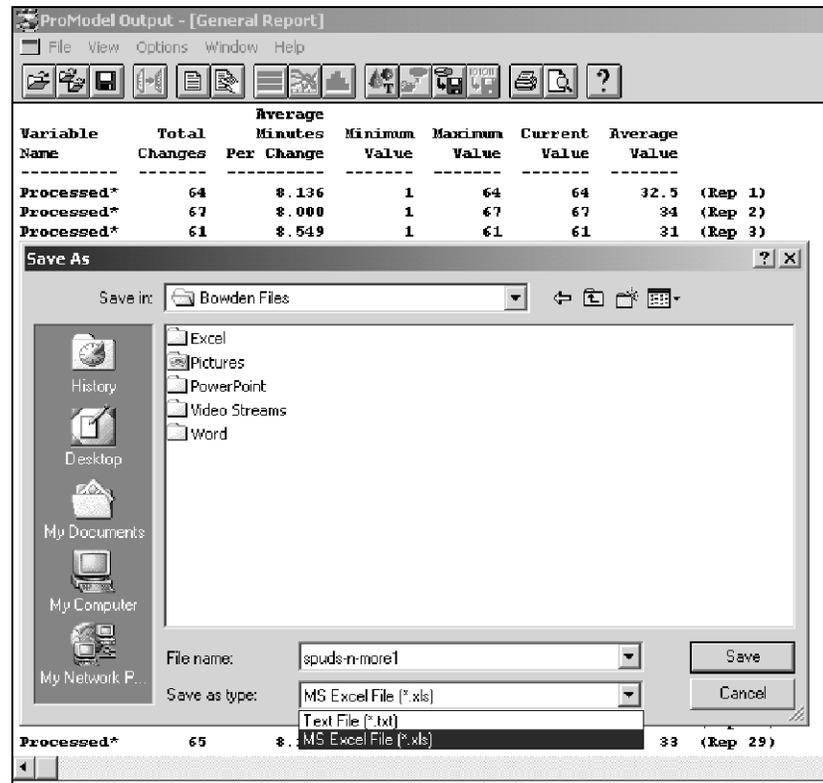
To compute the confidence interval, we assumed that the observations of the number of customers processed each day were independent and normally

distributed. The observations are independent because ProModel used a different stream of random numbers to simulate each replication. We do not know that the observations are normally distributed, however. Therefore, we should specify that the confidence level of the interval is approximately 0.90. Nevertheless, we followed an acceptable procedure to get our estimate for the service rate, and we can stand by our approximate confidence interval. To illustrate a point, however, let's take a moment to use the Stat::Fit software provided with ProModel to test if the simulation output is normally distributed. To do so, we need a lot more than eight observations. Let's pick 100 just because it is a round number and we can quickly generate them by configuring ProModel to run 100 replications of the simulation. Try this now. Notice that you get a very small confidence interval with this many observations.

It would take a while to enter the 100 observations into Stat::Fit one at a time. A faster method is to save the ProModel output report with the results from the 100 replications into a Microsoft Excel file format that can then be copied into the Stat::Fit data table. To do this from the ProModel output report viewer, select File from the main menu and click on the Save As option, then enter a name for the file, and save the file as type MS Excel (Figure L9.7). Open the Excel file that you just

FIGURE L9.7

*Saving a ProModel output report into a Microsoft Excel file format.*



**FIGURE L9.8**

The ProModel output report displayed within a Microsoft Excel spreadsheet.

Variable_Name	Total_Changes	Average_Mini	Minimum	Maximum	Current_Value	Average_Value
Processed*	64	8.136	1	64	64	32.5
Processed*	67	8	1	67	67	34
Processed*	61	8.549	1	61	61	31
Processed*	59	9.002	1	59	59	30
Processed*	60	8.753	1	60	60	30.5
Processed*	66	8.164	1	66	66	33.5
Processed*	57	9.244	1	57	57	29
Processed*	58	8.904	1	58	58	29.5
Processed*	56	9.054	1	56	56	28.5
Processed*	55	9.406	1	55	55	28
Processed*	61	8.777	1	61	61	31
Processed*	59	8.766	1	59	59	30
Processed*	57	9.011	1	57	57	29
Processed*	63	8.171	1	63	63	32
Processed*	65	7.858	1	65	65	33
Processed*	57	8.994	1	57	57	29
Processed*	53	9.366	1	53	53	27
Processed*	58	8.709	1	58	58	29.5
Processed*	65	7.874	1	65	65	33
Processed*	58	8.919	1	58	58	29.5
Processed*	55	9.331	1	55	55	28
Processed*	66	7.901	1	66	66	33.5
Processed*	63	8.693	1	63	63	32
Processed*	53	9.52	1	53	53	27

saved the ProModel output to, click on the VARIABLES sheet tab at the bottom of the spreadsheet (Figure L9.8), highlight the 100 Current Value observations of the Processed variable—making sure not to include the average, standard deviation, 90% C.I. Low, and 90% C.I. High values (the last four values in the column)—and paste the 100 observations into the Stat::Fit data table (Figure L9.9).

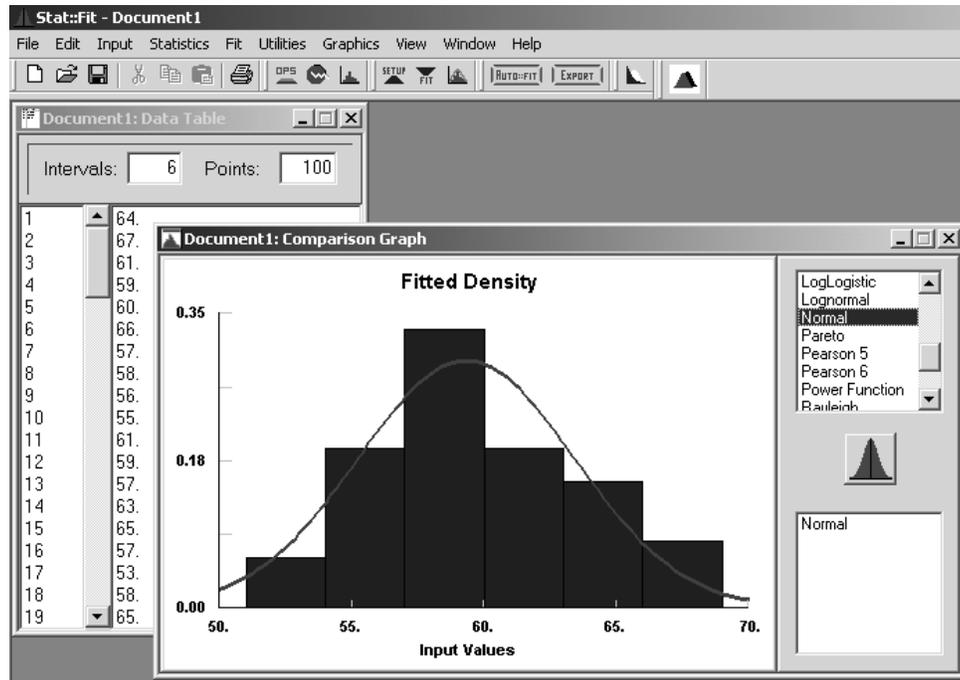
After the 100 observations are pasted into the Stat::Fit Data Table, display a histogram of the data. Based on the histogram, the observations appear somewhat normally distributed (Figure L9.9). Furthermore, Stat::Fit estimates that the normal distribution with mean  $\mu = 59.30$  and standard deviation  $\sigma = 4.08$  provides an acceptable fit to the data. Therefore, we probably could have dropped the word “approximate” when we presented our confidence interval to Mr. Taylor. Be sure to verify all of this for yourself using the software.

Note that if you were to repeat this procedure in practice because the problem requires you to be as precise and as sure as possible in your conclusions, then you would report your confidence interval based on the larger number of observations. Never discard precious data.

Before leaving this section, look back at Figure L9.6. The Total Failed column in the Failed Arrivals section of the output report indicates the number of customers that arrived to eat at Spuds-n-More but left because the waiting line

**FIGURE L9.9**

*Stat::Fit analysis of the 100 observations of the number of customers processed per day by Spuds-n-More.*



was full. Mr. Taylor thinks that his proposed expansion plan will allow him to capture some of the customers he is currently losing. In Lab Chapter 10, we will add embellishments to the as-is simulation model to reflect Mr. Taylor's expansion plan to see if he is right.

### L9.3 Nonterminating Simulation

We run a nonterminating simulation of a system if we are interested in understanding the steady-state behavior of the system—that is, its long term average behavior. Because a nonterminating system does not return to a fixed initial condition and theoretically never shuts down, it is not easy to select either the starting condition for the simulation or the amount of time to run the simulation.

When a simulation has just begun running, its state (number of customers in the system for example) will be greatly affected by the initial state of the simulation and the elapsed time of the run. In this case, the simulation is said to be in a transient condition. However, after sufficient time has elapsed, the state of the simulation becomes essentially independent of the initial state and the elapsed time. When this happens, the simulation has reached a steady-state condition. Although

the simulation’s output will continue to be random, the statistical distribution of the simulation’s output does not change after the simulation reaches steady state.

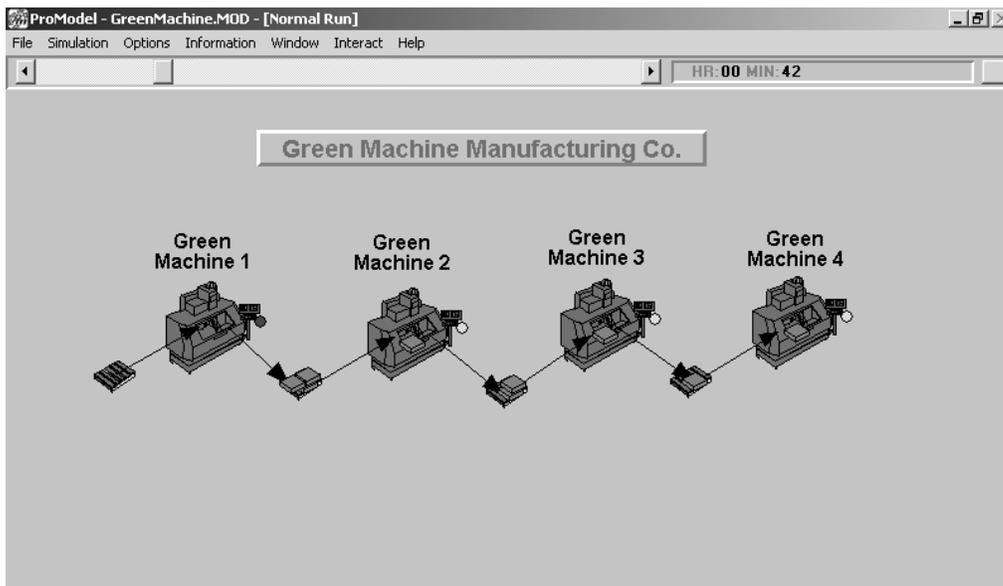
The period during which the simulation is in the transient phase is known as the warm-up period. This is the amount of time to let the simulation run before gathering statistics. Data collection for statistics begins at the end of the warm-up and continues until the simulation has run long enough to allow all simulation events (even rare ones) to occur many times (hundred to thousands of times if practical).

**Problem Statement**

A simulation model of the **Green Machine Manufacturing Company (GMMC)** owned and operated by Mr. Robert Vaughn is shown in Figure L9.10. The interarrival time of jobs to the GMMC is constant at 1.175 minutes. Jobs require processing by each of the four machines. The processing time for a job at each green machine is given in Table L9.2.

**FIGURE L9.10**

*The layout of the Green Machine Manufacturing Company simulation model.*



**TABLE L9.2 Process Times at Green Machine Manufacturing Company**

Activity	Process Time (minutes)	ProModel Format
Machine 1	Exponential with mean = 1.0	Wait E(1.0) min
Machine 2	Weibull with minimum = 0.10, $\alpha = 5, \beta = 1$	Wait (0.10 + W(5,1)) min
Machine 3	Weibull with minimum = 0.25, $\alpha = 5, \beta = 1$	Wait (0.25 + W(5,1)) min
Machine 4	Weibull with minimum = 0.25, $\alpha = 5, \beta = 1$	Wait (0.25 + W(5,1)) min

A plentiful amount of storage is provided for waiting jobs in front of each of the green machines. Our task is to conduct a nonterminating simulation to estimate the time-average amount of work-in-process (WIP) inventory (the steady-state expected number of jobs in the system). A printout of the model is given in Figure L9.11. The model is included on the CD accompanying the book under the file name Lab 9\_3 GreenMachine.MOD.

### L9.3.1 Warm-up Time and Run Length

In this section we estimate the amount of time needed for the GMMC model to warm up, and the amount of time to run the model past its warm-up period. We first declare a model variable called WIP to keep track of the work-in-process inventory in the simulated manufacturing system. Notice we are collecting statistics only for the WIP variable and the entity. You should configure your model the same and run the simulation 250 hours to produce a time-series plot of the WIP inventory levels (Figure L9.12(a)).

The simplest method for selecting the end of the warm-up phase is visual determination. In this method, we plot the simulation model's output over time, as illustrated in Figure L9.12(a), and then visually determine the point at which the initial transient phase is over. In this example, it seems that the transient phase created by the initial condition of zero WIP inventory is over at around 20 hours. Usually you will want to select a value beyond this to guard against underestimating the end of the warm-up phase. So we will pick 100 hours as the end of the warm-up. Actually, you should not make this decision based on the output from one replication of the simulation. We are doing this here to quickly illustrate the need for allowing the model to warm up. We will use the preferred method to estimate the end of the warm-up phase in a moment.

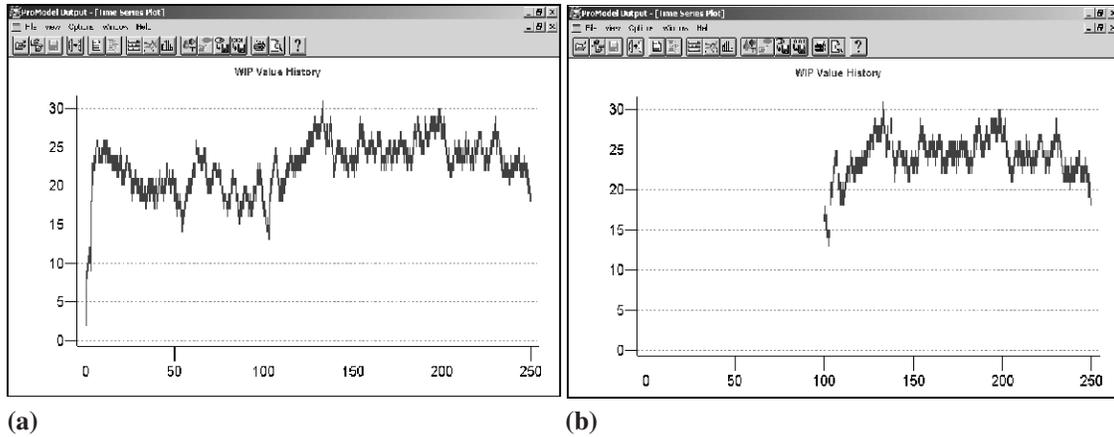
In most nonterminating simulation models, the quality of the results substantially improves if we throw away the output generated during the warm-up phase of the simulation. In Figure L9.12(a), the observations during the transient phase of the first 100 hours of the simulation are alleged to be artificially low. As such, we do not want them to bias our estimate of the time-average WIP inventory. Therefore, in our next run of the GMMC simulation, we specify a warm-up time of 100 hours (Figure L9.13). Notice how the simulation output during the 100-hour warm-up has been discarded in Figure L9.12(b). The discarded output will no longer affect the observations reported in the simulation output.

The simplified process we just used to identify the end of the warm-up phase should be avoided. The process can be substantially improved by running multiple replications of the simulation and then combining the time-series output from each replication into an average time-series plot. Averaging the time-series data from multiple replications together will help filter out the large fluctuations (high values from one replication are offset by small values from another replication) and *will give you a better estimate of the true but unknown time-series output*. In many cases, you may need to add an additional layer of filtering to smooth the “averaged” time-series plot. This can be done by computing a Welch moving average of the time-series data. Section 9.6.1 of Chapter 9 provides additional



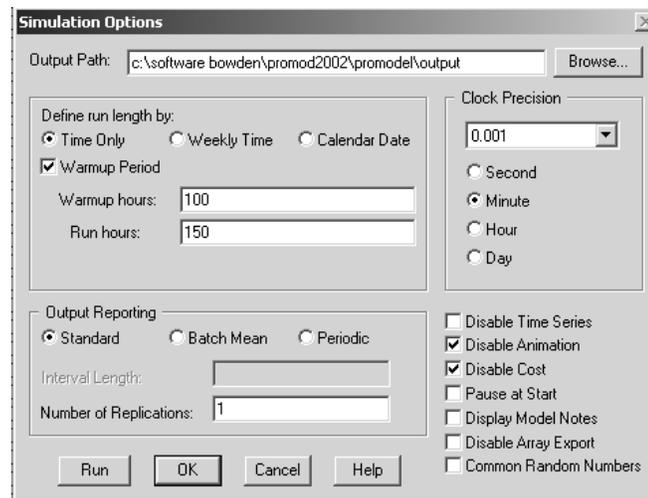
**FIGURE L9.12**

Work-in-process (WIP) inventory value history for one replication of the GMMC simulation. (a) WIP value history without warm-up phase removed. Statistics will be biased low by the transient WIP values. (b) WIP value history with 100-hour warm-up phase removed. Statistics will not be biased low.



**FIGURE L9.13**

ProModel's Simulation Options window set for a single replication with a 100-hour warm-up period followed by a 150-hour run length.

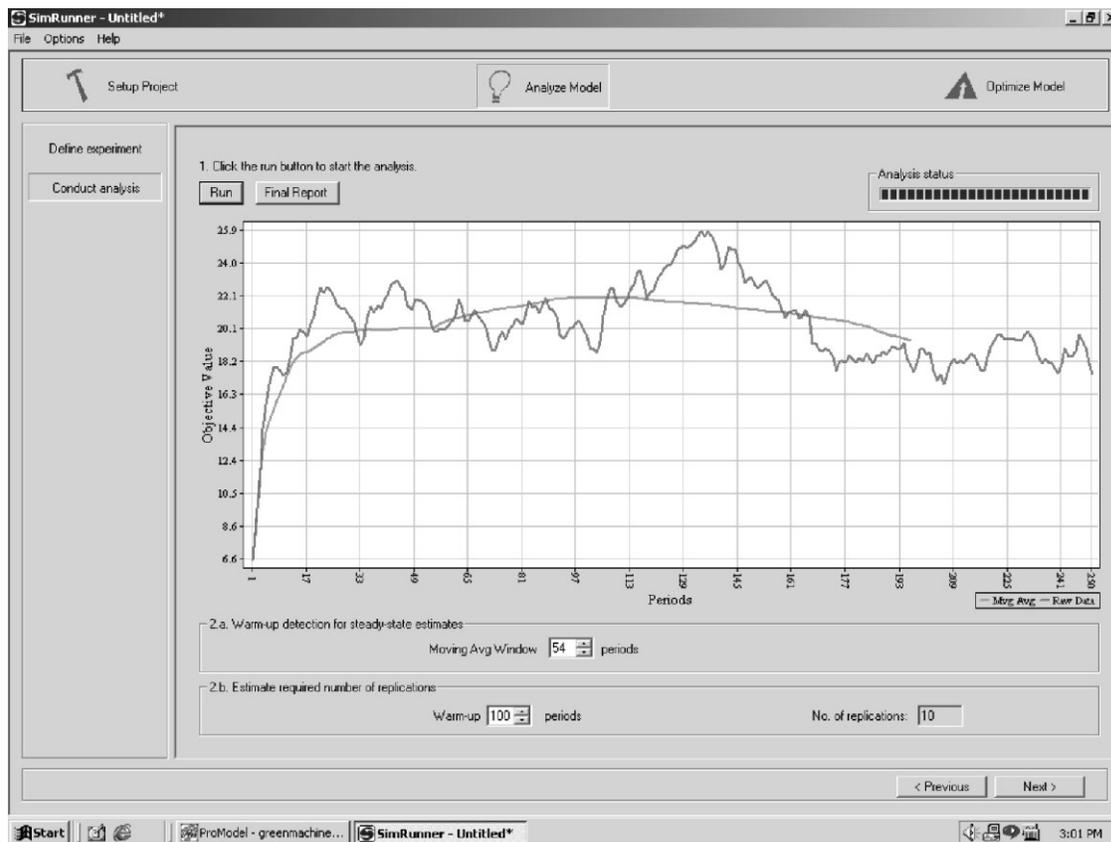


details on this subject, and exercise 4 in Lab Chapter 11 illustrates how to use the Welch method implemented in SimRunner.

Figure L9.14 was produced by SimRunner by recording the time-average WIP levels of the GMMC simulation over successive one-hour time periods. The results from each period were averaged across five replications to produce the "raw data" plot, which is the more erratic line that appears red on the computer screen. A 54-period moving average (the smooth line that appears green on the

FIGURE L9.14

SimRunner screen for estimating the end of the warm-up time.



computer) indicates that the end of the warm-up phase occurs between the 33rd and 100th periods. Given that we need to avoid underestimating the warm-up, let's declare 100 hours as the end of the warm-up time. We feel much more comfortable basing our estimate of the warm-up time on five replications. Notice that SimRunner indicates that at least 10 replications are needed to estimate the average WIP to within a 7 percent error and a confidence level of 90 percent using a warm-up of 100 periods (hours). You will see how this was done with SimRunner in Exercise 4 of Section L11.4 in Lab Chapter 11.

Why did we choose an initial run length of 250 hours to produce the time-series plot in Figure L9.14? Well, you have to start with something, and we picked 250 hours. You can rerun the experiment with a longer run length if the time-series plot, produced by averaging the output from several replications of the simulation, does not indicate a steady-state condition. Long runs will help prevent

you from wondering what the time-series plot looks like beyond the point at which you stopped it. Do you wonder what our plot does beyond 250 hours?

Let's now direct our attention to answering the question of how long to run the model past its warm-up time to estimate our steady-state statistic, mean WIP inventory. We will somewhat arbitrarily pick 100 hours not because that is equal to the warm-up time but because it will allow ample time for the simulation events to happen thousands of times. In fact, the 100-hour duration will allow approximately 5,100 jobs to be processed per replication, which should give us decently accurate results. How did we derive the estimate of 5,100 jobs processed per replication? With a 1.175-minute interarrival time of jobs to the system, 51 jobs arrive per hour (60 minutes/1.175 minutes) to the system. Running the simulation for 100 hours should result in about 5,100 jobs (100 hours  $\times$  51 jobs) exiting the system. You will want to check the number of Total Exits in the Entity Activity section of the ProModel output report that you just produced to verify this.

### L9.3.2 Replications or Batch Intervals

We have a choice between running replications or batch intervals to estimate the time-average WIP inventory. Replications are our favorite choice because we get independent observations, guaranteed. Figure L9.15 illustrates how this is done in ProModel as well as the specification of the 100 hours of run time beyond the warm-up. Running the simulation for 10 replications produces the results in Figure L9.16. We are approximately 90 percent confident that the true but unknown mean WIP inventory is between 17.64 and 21.09 jobs.

**FIGURE L9.15**

*Specification of warm-up hours and run hours in the Simulation Options menu for running replications.*

Simulation Options

Output Path: c:\software bowden\promod2002\promodel\output Browse...

Define run length by:

Time Only  Weekly Time  Calendar Date

Warmup Period

Warm up hours: 100

Run hours: 100

Output Reporting

Standard  Batch Mean  Periodic

Interval Length:

Number of Replications: 10

Clock Precision

0.001

Second  Minute  Hour  Day

Disable Time Series  Disable Animation  Disable Cost  Pause at Start  Display Model Notes  Disable Array Export

Run OK Cancel Help

**FIGURE L9.16**

*Ten replications of the Green Machine Manufacturing Company simulation using a 100-hour warm-up and a 100-hour run length.*

```

-----
General Report
Output from C:\Bowden Files\Word\McGraw 2nd Edition\GreenMachine.MOD
Date: Jul/16/2002   Time: 01:04:42 AM
-----
Scenario           : Normal Run
Replication        : All
Period            : Final Report (100 hr to 200 hr Elapsed: 100 hr)
Warmup Time       : 100
Simulation Time    : 200 hr
-----
VARIABLES

```

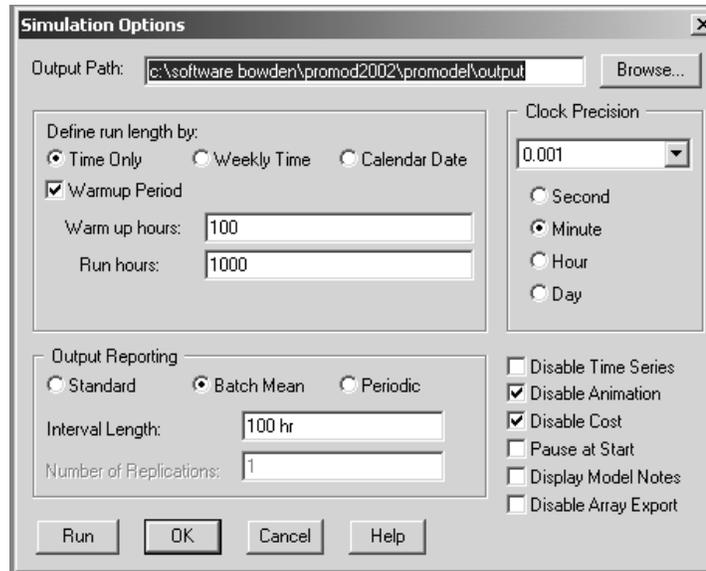
Variable Name	Total Changes	Average Minutes Per Change	Minimum Value	Maximum Value	Current Value	Average Value	
WIP	10202	0.588	13	31	27	24.369	(Rep 1)
WIP	10198	0.588	9	32	29	18.928	(Rep 2)
WIP	10207	0.587	14	33	22	24.765	(Rep 3)
WIP	10214	0.587	10	26	22	18.551	(Rep 4)
WIP	10208	0.587	8	27	25	16.929	(Rep 5)
WIP	10216	0.587	13	27	17	19.470	(Rep 6)
WIP	10205	0.587	8	30	24	19.072	(Rep 7)
WIP	10215	0.587	8	25	14	15.636	(Rep 8)
WIP	10209	0.587	9	26	20	17.412	(Rep 9)
WIP	10209	0.587	11	27	24	18.504	(Rep 10)
WIP	10208.3	0.587	10.3	28.4	22.4	19.364	(Average)
WIP	5.735	0.0	2.311	2.836	4.501	2.973	(Std. Dev.)
WIP	10205	0.587	8.959	26.756	19.790	17.640	(90% C.I. Low)
WIP	10211.6	0.587	11.64	30.044	25.009	21.087	(90% C.I. High)

If we decide to make one single long run and divide it into batch intervals to estimate the expected WIP inventory, we would select the Batch Mean option in the Output Reporting section of the Simulation Options menu (Figure L9.17). A guideline in Chapter 9 for making an initial assignment to the length of time for each batch interval was to set the batch interval length to the simulation run time you would use for replications, in this case 100 hours. Based on the guideline, the Simulation Options menu would be configured as shown in Figure L9.17 and would produce the output in Figure L9.18. To get the output report to display the results of all 10 batch intervals, you specify “<All>” Periods in the General Report Type settings window (Figure L9.19). We are approximately 90 percent confident that the true but unknown mean WIP inventory is between 18.88 and 23.19 jobs. If we desire a smaller confidence interval, we can increase the sample size by extending the run length of the simulation in increments of the batch interval length. For example, we would specify run hours of 1500 to collect 15 observations of average WIP inventory. Try this and see if the confidence interval becomes smaller. Note that increasing the batch interval length is also helpful.

Lab 9 Simulation Output Analysis

**FIGURE L9.17**

Warm-up hours, run hours, and batch interval length in the Simulation Options menu for running batch intervals. Note that the time units are specified on the batch interval length.



**FIGURE L9.18**

Ten batch intervals of the Green Machine Manufacturing Company simulation using a 100-hour warm-up and a 100-hour batch interval length.

```

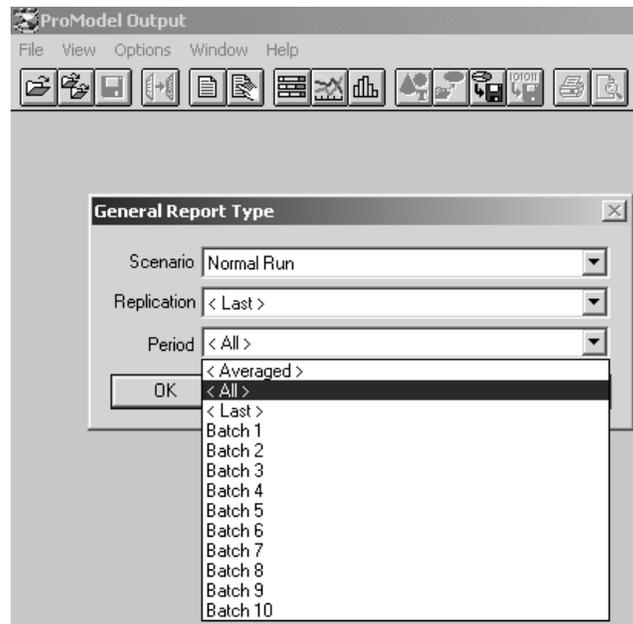
-----
General Report
Output from C:\Bowden Files\Word\McGraw 2nd Edition\GreenMachine.MOD
Date: Jul/16/2002   Time: 01:24:24 AM
-----
Scenario           : Normal Run
Replication        : 1 of 1
Period            : All
Warmup Time       : 100 hr
Simulation Time   : 1100 hr
-----
VARIABLES

```

Variable Name	Total Changes	Average Minutes Per Change	Minimum Value	Maximum Value	Current Value	Average Value	
WIP	10202	0.588	13	31	27	24.369	(Batch 1)
WIP	10215	0.587	16	29	26	23.082	(Batch 2)
WIP	10216	0.587	9	30	22	17.595	(Batch 3)
WIP	10218	0.587	13	30	16	22.553	(Batch 4)
WIP	10202	0.588	13	33	28	23.881	(Batch 5)
WIP	10215	0.587	17	37	25	27.335	(Batch 6)
WIP	10221	0.587	8	29	18	20.138	(Batch 7)
WIP	10208	0.587	10	26	22	17.786	(Batch 8)
WIP	10214	0.587	9	29	20	16.824	(Batch 9)
WIP	10222	0.586	9	23	12	16.810	(Batch 10)
WIP	10213.3	0.587	11.7	29.7	21.6	21.037	(Average)
WIP	7.103	0.0	3.164	3.743	5.168	3.714	(Std. Dev.)
WIP	10209.2	0.587	9.865	27.530	18.604	18.884	(90% C.I. Low)
WIP	10217.4	0.587	13.534	31.869	24.595	23.190	(90% C.I. High)

**FIGURE L9.19**

*ProModel General Report Options set to display the results from all batch intervals.*



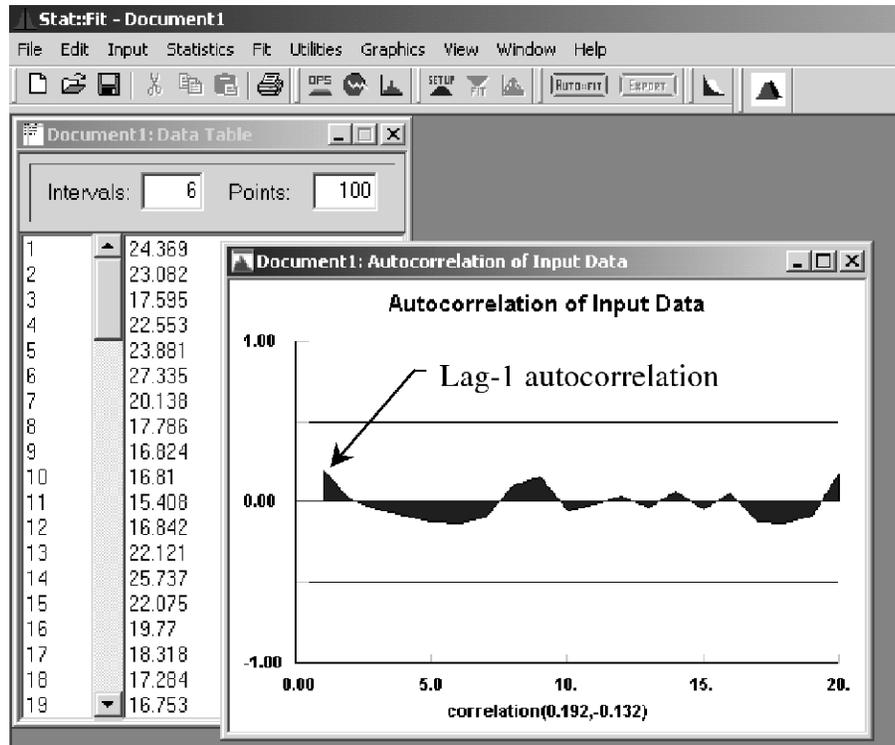
### L9.3.3 Required Batch Interval Length

The worrisome part of the batch interval method is whether the batch interval is long enough to acquire “approximately” independent observations. The word “approximately” is emphasized because we know that the observations within a simulation run are often autocorrelated (not independent). In fact, they are often positively correlated, which would bias the sample standard deviation of the batch means to the low side. And this would bias our estimate of the confidence interval’s half-width to the low side. We do not know if this is the case with our WIP inventory observations. However, if it was the case and we were not aware of it, we would think that our estimate of the mean WIP inventory was better than it really is. Therefore, we should do ourselves a favor and use Stat::Fit to check the lag-1 autocorrelation of the observations before we interpret the results. To get a decent estimate of the lag-1 autocorrelation, however, we need more than 10 observations. Section 9.6.2 of Chapter 9 stated that at least 100 observations are necessary to get a reliable estimate of the lag-1 autocorrelation. So let’s increase the simulation run length to 10,000 hours and produce an output report with 100 observations of the time-average WIP inventory.

It would take a while to enter the 100 observations into Stat::Fit one at a time. Therefore, save the ProModel output report with the results from the 100 batch intervals into a Microsoft Excel file format that can then be copied into the Stat::Fit data table. The procedure is the same as used in Section L9.2.4 to enter the observations under the Current Value column from the VARIABLES sheet tab of the Excel spreadsheet into Stat::Fit—except that you are now entering the

**FIGURE L9.20**

*Stat::Fit Autocorrelation plot of the observations collected over 100 batch intervals. Lag-1 autocorrelation is within the  $-0.20$  to  $+0.20$  range.*



observations under the Average Value column from the VARIABLES sheet tab of the Excel spreadsheet into Stat::Fit.

Figure L9.20 illustrates the Stat::Fit results that you will want to verify. The lag-1 autocorrelation value is the first value plotted in Stat::Fit's Autocorrelation of Input Data plot. Note that the plot begins at lag-1 and continues to lag-20. For this range of lag values, the highest autocorrelation is 0.192 and the lowest value is  $-0.132$  (see correlation(0.192,  $-0.132$ ) at the bottom of the plot). Therefore, we know that the lag-1 autocorrelation is within the  $-0.20$  to  $+0.20$  range recommended in Section 9.6.2 of Chapter 9, which is required before proceeding to the final step of "rebatching" the data into between 10 and 30 larger batches. We have enough data to form 10 batches with a length of 1000 hours (10 batches at 1000 hours each equals 10,000 hours of simulation time, which we just did). The results of "rebatching" the data in the Microsoft Excel spreadsheet are shown in Table L9.3. Note that the first batch mean of 21.037 in Table L9.3 is the average of the first 10 batch means that we originally collected. The second batch mean of 19.401 is the average of the next 10 batch means and so on. The confidence

**TABLE L9.3 Final Results of the 100 Batch Intervals  
Combined into 10 Batch Intervals**

<i>WIP Average Value</i>	
21.037	(Batch 1)
19.401	(Batch 2)
21.119	(Batch 3)
20.288	(Batch 4)
20.330	(Batch 5)
20.410	(Batch 6)
22.379	(Batch 7)
22.024	(Batch 8)
19.930	(Batch 9)
22.233	(Batch 10)
20.915	(Average)
1.023	(Std. Dev.)
20.322	(90% C.I. Low)
21.508	(90% C.I. High)

interval is very narrow due to the long simulation time (1000 hours) of each batch. The owner of GMMC, Mr. Robert Vaughn, should be very pleased with the precision of the time-average WIP inventory estimate.

The batch interval method requires a lot of work to get it right. Therefore, if the time to simulate through the warm-up period is relatively short, the replications method should be used. Reserve the batch interval method for simulations that require a very long time to reach steady state.

## L9.4 Exercises

1. An average of 100 customers per hour arrive to the Picayune Mutual Bank. It takes a teller an average of two minutes to serve a customer. Interarrival and service times are exponentially distributed. The bank currently has four tellers working. Bank manager Rich Gold wants to compare the following two systems with regard to the average time customers spend in the bank.

### System #1

A separate queue is provided for each teller. Assume that customers choose the shortest queue when entering the bank, and that customers cannot jockey between queues (jump to another queue).

### System #2

A single queue is provided for customers to wait for the first available teller.

- Assume that there is no move time within the queues. Run 15 replications of an eight-hour simulation to complete the following:
- For each system, record the 90 percent confidence interval using a 0.10 level of significance for the average time customers spend in the bank.
  - Estimate the number of replications needed to reduce the half-width of each confidence interval by 25 percent. Run the additional replications for each system and compute new confidence intervals using a 0.10 level of significance.
  - Based on the results from part *b*, would you recommend one system over the other? Explain.
- Rerun the simulation of the Green Machine Manufacturing Company of Section L9.3 to produce 100 batch means with a 50-hour interval length. Use a 100-hour warm-up. Check the lag-1 autocorrelation of the 100 time-average WIP observations to see if it falls between  $-0.20$  and  $+0.20$ . If it does not, increase the batch interval length by 50 percent and continue the process until an acceptable lag-1 autocorrelation is achieved. Once the lag-1 autocorrelation is acceptable, check to see if the 100 observations are normally distributed. After that, rebatch the 100 observations into 20 batch means and compute a 95 percent confidence interval. Are you comfortable with the resulting confidence interval? Explain.
  - Use the simulation model created for the DumpOnMe facility presented in Exercise 11 of Lab Section L7.12 to complete the following:
    - For a simulation run length of 60 minutes, record the 90 percent confidence interval for the scale's average utilization based on 30 replications.
    - For a 60-minute warm-up and a 60-minute simulation run length beyond the warm-up time, record the 90 percent confidence interval for the scale's average utilization based on 30 replications.
    - Based only on the two confidence intervals, do you think that a warm-up period is necessary for the DumpOnMe model? Explain.
  - See Lab Chapter 11 for exercises involving the use of SimRunner's implementation of the Welch moving average technique.