Example 9.19

A simulation of the web-based trading site of a stock broker includes the time between arrivals of orders to buy and sell. Investors tend to react to what other investors are doing, so these buy and sell orders arrive in bursts. Therefore, rather than treat the time between arrivals as independent random variables, a time-series model should be developed.

We distinguish *multivariate input models* of a fixed, finite number of random variables (such as the two random variables *lead time* and *annual demand* in Example 9.18) from *time-series input models* of a (conceptually infinite) sequence of related random variables (such as the successive times between orders in Example 9.19). We will describe input models appropriate for these examples after reviewing two measures of dependence, the covariance and the correlation.

9.7.1 Covariance and Correlation

Let X_1 and X_2 be two random variables, and let $\mu_i = E(X_i)$ and $\sigma_i^2 = V(X_i)$ be the mean and variance of X_i, respectively. The *covariance* and *correlation* are measures of the linear dependence between X_1 and X_2. In other words, the covariance and correlation indicate how well the relationship between X_1 and X_2 is described by the model

$$(X_1 - \mu_1) = \beta(X_2 - \mu_2) + \epsilon$$

where ϵ is a random variable with mean 0 that is independent of X_2. If, in fact, $(X_1 - \mu_1) = \beta(X_2 - \mu_2)$, then this model is perfect. On the other hand, if X_1 and X_2 are statistically independent, then $\beta = 0$ and the model is of no value. In general, a positive value of β indicates that X_1 and X_2 tend to be above or below their means together; a negative value of β indicates that they tend to be on opposite sides of their means.

The covariance between X_1 and X_2 is defined to be

$$\text{cov}(X_1, X_2) = E[(X_1 - \mu_1)(X_2 - \mu_2)] = E(X_1 X_2) - \mu_1 \mu_2 \tag{9.22}$$

The value $\text{cov}(X_1, X_2) = 0$ implies $\beta = 0$ in our model of dependence, and $\text{cov}(X_1, X_2) < 0$ (>0) implies $\beta < 0$ (>0).

The covariance can take any value between $-\infty$ and ∞. The correlation standardizes the covariance to be between -1 and 1:

$$\rho = \text{corr}(X_1, X_2) = \frac{\text{cov}(X_1, X_2)}{\sigma_1 \sigma_2} \tag{9.23}$$

Again, the value $\text{corr}(X_1, X_2) = 0$ implies $\beta = 0$ in our model, and $\text{corr}(X_1, X_2) < 0$ (>0) implies $\beta < 0$ (>0). The closer ρ is to -1 or 1, the stronger the linear relationship is between X_1 and X_2.

Now suppose that we have a sequence of random variables $X_1, X_2, X_3, \ldots$ that are identically distributed (implying that they all have the same mean and variance), but could be dependent. We refer to such a sequence as a *time series* and to $\text{cov}(X_t, X_{t+h})$ and $\text{corr}(X_t, X_{t+h})$ as the *lag-h autocovariance* and *lag-h autocorrelation*, respectively. If the value of the autocovariance depends only on h and not on t, then we say that the time series is *covariance stationary*; this concept is discussed further in Chapter 11. For a covariance-stationary time series, we use the shorthand notation

$$\rho_h = \text{corr}(X_t, X_{t+h})$$

for the *lag-h* autocorrelation. Notice that autocorrelation measures the dependence between random variables that are separated by $h - 1$ others in the time series.

9.7.2 Multivariate Input Models

If X_1 and X_2 each are normally distributed, then dependence between them can be modeled by the bivariate normal distribution with parameters μ_1, μ_2, σ_1^2, σ_2^2, and $\rho = \text{corr}(X_1, X_2)$. Estimation of μ_1, μ_2, σ_1^2, and σ_2^2 was described in Section 9.3.2. To estimate ρ, suppose that we have n independent and identically distributed pairs (X_{11}, X_{21}), (X_{12}, X_{22}), ..., (X_{1n}, X_{2n}). Then the sample covariance is

$$\begin{aligned}\widehat{\text{cov}}(X_1, X_2) &= \frac{1}{n-1}\sum_{j=1}^{n}\left(X_{1j} - \bar{X}_1\right)\left(X_{2j} - \bar{X}_2\right) \\ &= \frac{1}{n-1}\left(\sum_{j=1}^{n} X_{1j}X_{2j} - n\bar{X}_1\bar{X}_2\right)\end{aligned} \tag{9.24}$$

where $\bar{X}_1$ and $\bar{X}_2$ are the sample means. The correlation is estimated by

$$\hat{\rho} = \frac{\widehat{\text{cov}}(X_1, X_2)}{\hat{\sigma}_1\hat{\sigma}_2} \tag{9.25}$$

where $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are the sample variances.

Example 9.20: Example 9.18 Continued

Let X_1 represent the average lead time to deliver (in months), and X_2 the annual demand, for industrial robots. The following data are available on demand and lead time for the last ten years:

lead time	demand
6.5	103
4.3	83
6.9	116
6.0	97
6.9	112
6.9	104
5.8	106
7.3	109
4.5	92
6.3	96

Standard calculations give $\bar{X}_1 = 6.14$, $\hat{\sigma}_1 = 1.02$, $\bar{X}_2 = 101.80$, and $\hat{\sigma}_2 = 9.93$ as estimates of μ_1, σ_1, μ_2, and σ_2, respectively. To estimate the correlation, we need

$$\sum_{j=1}^{10} X_{1j}X_{2j} = 6328.5$$

Therefore, $\widehat{\text{cov}} = [6328.5 - (10)(6.14)(101.80)]/(10-1) = 8.66$, and

$$\hat{\rho} = \frac{8.66}{(1.02)(9.93)} = 0.86$$

Clearly, lead time and demand are strongly dependent. Before we accept this model, however, lead time and demand should be checked individually to see whether they are represented well by normal distributions.

In particular, demand is a discrete-valued quantity, so the continuous normal distribution is certainly at best an approximation.

The following simple algorithm can be used to generate bivariate normal random variables:

Step 1. Generate Z_1 and Z_2, independent standard normal random variables (see Section 8.3.1).

Step 2. Set $X_1 = \mu_1 + \sigma_1 Z_1$

Step 3. Set $X_2 = \mu_2 + \sigma_2\left(\rho Z_1 + \sqrt{1-\rho^2}\, Z_2\right)$

Obviously, the bivariate normal distribution will not be appropriate for all multivariate-input modeling problems. It can be generalized to the k-variate normal distribution to model the dependence among more than two random variables, but, in many instances, a normal distribution is not appropriate in any form. We provide one method for handling nonnormal distributions in Section 9.7.4. Good references for other models are Johnson [1987] and Nelson and Yamnitsky [1998].

9.7.3 Time-Series Input Models

If $X_1, X_2, X_3, \ldots$ is a sequence of identically distributed, but dependent and covariance-stationary random variables, then there are a number of time series models that can be used to represent the process. We will describe two models that have the characteristic that the autocorrelations take the form

$$\rho_h = \text{corr}(X_t, X_{t+h}) = \rho^h$$

for $h = 1, 2, \ldots$ Notice that the *lag-h* autocorrelation decreases geometrically as the lag increases, so that observations far apart in time are nearly independent. For one model to be shown shortly, each X_t is normally distributed; for the other model, each X_t is exponentially distributed. More general time-series input models are described in Section 9.7.4 and in Nelson and Yamnitsky [1998].

AR(1) MODEL. Consider the time-series model

$$X_t = \mu + \phi(X_{t-1} - \mu) + \varepsilon_t \tag{9.26}$$

for $t = 2, 3, \ldots$, where $\varepsilon_2, \varepsilon_3, \ldots$ are independent and identically (normally) distributed with mean 0 and variance σ_ε^2, and $-1 < \phi < 1$. If the initial value X_1 is chosen appropriately (see shortly), then $X_1, X_2, \ldots$ are all normally distributed with mean μ, variance $\sigma_\varepsilon^2/(1-\phi^2)$, and

$$\rho_h = \phi^h$$

for $h = 1, 2, \ldots$. This time-series model is called the autoregressive order-1 model, or AR(1) for short.

Estimation of the parameter ϕ can be obtained from the fact that

$$\phi = \rho^1 = \text{corr}(X_t, X_{t+1})$$

the lag-1 autocorrelation. Therefore, to estimate ϕ, we first estimate the lag-1 autocovariance by

$$\begin{aligned}\widehat{\text{cov}}(X_t, X_{t+1}) &= \frac{1}{n-1}\sum_{t=1}^{n-1}\left(X_t - \bar{X}\right)\left(X_{t+1} - \bar{X}\right) \\ &\doteq \frac{1}{n-1}\left(\sum_{t=1}^{n-1} X_t X_{t+1} - (n-1)\bar{X}_2\right)\end{aligned} \tag{9.27}$$

and the variance $\sigma^2 = \text{var}(X)$ by the usual estimator $\hat{\sigma}^2$. Then

$$\hat{\phi} = \frac{\widehat{\text{cov}}(X_t, X_{t+1})}{\hat{\sigma}^2}$$

Finally, estimate μ and σ_ε^2 by $\hat{\mu} = \bar{X}$ and

$$\hat{\sigma}_\varepsilon^2 = \hat{\sigma}^2(1 - \hat{\phi}^2)$$

respectively.

The following algorithm generates a stationary AR(1) time series, given values of the parameters ϕ, μ, and σ_ε^2:

Step 1. Generate X_1 from the normal distribution with mean μ and variance $\sigma_\varepsilon^2/(1-\phi^2)$. Set $t = 2$.

Step 2. Generate ε_t from the normal distribution with mean 0 and variance σ_ε^2.

Step 3. Set $X_t = \mu + \phi(X_{t-1} - \mu) + \varepsilon_t$.

Step 4. Set $t = t + 1$ and go to Step 2.

EAR(1) MODEL. Consider the time-series model

$$X_t = \begin{cases} \phi X_{t-1}, & \text{with probability } \phi \\ \phi X_{t-1} + \varepsilon_t, & \text{with probability } 1-\phi \end{cases} \tag{9.28}$$

for $t = 2, 3, \ldots$, where $\varepsilon_2, \varepsilon_3, \ldots$ are independent and identically (exponentially) distributed with mean $1/\lambda$ and $0 \le \phi < 1$. If the initial value X_1 is chosen appropriately (see shortly), then $X_1, X_2, \ldots$ are all exponentially distributed with mean $1/\lambda$ and

$$\rho_h = \phi^h$$

for $h = 1, 2, \ldots$. This time-series model is called the exponential autoregressive order-1 model, or EAR(1) for short. Only autocorrelations greater than 0 can be represented by this model. Estimation of the parameters proceeds as for the AR(1) by setting $\hat{\phi} = \hat{\rho}$, the estimated lag-1 autocorrelation, and setting $\hat{\lambda} = 1/\bar{X}$.

The following algorithm generates a stationary EAR(1) time series, given values of the parameters ϕ and λ:

Step 1. Generate X_1 from the exponential distribution with mean $1/\lambda$. Set $t = 2$.

Step 2. Generate U from the uniform distribution on [0, 1]. If $U \le \phi$, then set

$$X_t = \phi X_{t-1}$$

Otherwise, generate ε_t from the exponential distribution with mean $1/\lambda$ and set

$$X_t = \phi X_{t-1} + \varepsilon_t$$

Step 3. Set $t = t + 1$ and go to Step 2.

Example 9.21: Example 9.19 Continued

The stock broker would typically have a large sample of data, but, for the sake of illustration, suppose that the following twenty time gaps between customer buy and sell orders had been recorded (in seconds): 1.95, 1.75, 1.58, 1.42, 1.28, 1.15, 1.04, 0.93, 0.84, 0.75, 0.68, 0.61, 11.98, 10.79, 9.71, 14.02, 12.62, 11.36, 10.22, 9.20. Standard calculations give $\bar{X} = 5.2$ and $\hat{\sigma}^2 = 26.7$. To estimate the lag-1 autocorrelation, we need

$$\sum_{j=1}^{19} X_t X_{t+1} = 924.1$$

Thus, $\widehat{\text{cov}} = [924.1 - (20-1)(5.2)^2]/(20-1) = 21.6$, and

$$\hat{\rho} = \frac{21.6}{26.7} = 0.8$$

Therefore, we could model the interarrival times as an EAR(1) process with $\hat{\lambda} = 1/5.2 = 0.192$ and $\hat{\phi} = 0.8$, provided that an exponential distribution is a good model for the individual gaps.

9.7.4 The Normal-to-Anything Transformation

The bivariate normal distribution and the AR(1) and EAR(1) time-series models are useful input models that are easy to fit and simulate. However, the marginal distribution is either normal or exponential, which is certainly not the best choice for many applications. Fortunately, we can start with a bivariate normal or AR(1) model and *transform* it to have any marginal distributions we want (including exponential).

Suppose we want to simulate a random variable X with cdf $F(x)$. Let Z be a standard normal random variable (mean 0 and variance 1), and let $\Phi(z)$ be its cdf. Then it can be shown that

$$R = \Phi(Z)$$

is a $U(0, 1)$ random variable. As we learned in Chapter 8, if we have a $U(0, 1)$ random variable, we can get X by using the inverse cdf transformation

$$X = F^{-1}[R] = F^{-1}[\Phi(Z)]$$

We refer this as the *normal to anything transformation*, or NORTA for short.

Of course, if all we want is X, then there is no reason to go to this trouble; we can just generate R directly, using the methods in Chapter 8. But suppose we want a bivariate random vector (X_1, X_2) such that X_1 and X_2 are correlated but their distributions are not normal. Then we can start with a bivariate normal random vector (Z_1, Z_2) and apply the NORTA transformation to obtain

$$X_1 = F_1^{-1}[\Phi(Z_1)] \text{ and } X_2 = F_2^{-1}[\Phi(Z_2)]$$

There is not even a requirement that F_1 and F_2 be from the same distribution family; for instance, F_1 could be an exponential distribution and F_2 a beta distribution.

The same idea applies for time series. If Z_t is generated by an AR(1) with $N(0, 1)$ marginals, then

$$X_t = F^{-1}[\Phi(Z_t)]$$

will be a time-series model with marginal distribution $F(x)$. To insure that Z_t is $N(0, 1)$, we set $\mu = 0$ and $\sigma_\varepsilon^2 = 1 - \phi^2$ in the AR(1) model.

Although the NORTA method is very general, there are two technical issues that must be addressed to implement it:

1. The NORTA approach requires being able to evaluate that standard normal cdf, $\Phi(z)$, and the inverse cdf of the distributions of interest, $F^{-1}(u)$. There is no closed-form expression for $\Phi(z)$ and no closed-form expression for $F^{-1}(u)$ for many distributions. Therefore, numerical approximations are required. Fortunately, these functions are built into many symbolic calculation and spreadsheet programs, and we give one example next. In addition, Bratley, Fox, and Schrage [1987] contains algorithms for many distributions.
2. The correlation between the standard normal random variables (Z_1, Z_2) is distorted when it passes through the NORTA transformation. To be more specific, if (Z_1, Z_2) have correlation ρ, then in

```
NORTARho := proc(rhoX, n)
local Z1, Z2, Ztemp, X1, X2, R1, R2, rho, rhoT, lower, upper;
randomize(123456);
Z1 := [random[normald[0,1]](n)]:
ZTemp := [random[normald[0,1]](n)]:
Z2 := [0]:
# set up bisection search
rho := rhoX:
if (rhoX < 0) then
     lower := -1:
     upper := 0:
else
     lower := 0:
     upper := 1:
fi:
Z2 := rho*Z1 + sqrt(1-rho^2)*ZTemp:
R1 := statevalf[cdf,normald[0,1]](Z1):
R2 := statevalf[cdf,normald[0,1]](Z2):
X1 := statevalf[icdf,exponential[1,0]](R1):
X2 := statevalf[icdf,beta[1,2]](R2):
rhoT := describe[linearcorrelation](X1, X2);
# do bisection search until 5% relative error
while abs(rhoT - rhoX)/abs(rhoX) > 0.05 do
     if (rhoT > rhoX) then
          upper := rho:
     else
          lower := rho:
     fi:
     rho := evalf((lower + upper)/2):
     Z2 := rho*Z1 + sqrt(1-rho^2)*ZTemp:
     R1 := statevalf[cdf,normald[0,1]](Z1):
     R2 := statevalf[cdf,normald[0,1]](Z2):
     X1 := statevalf[icdf,exponential[1,0]](R1):
     X2 := statevalf[icdf,beta[1,2]](R2):
     rhoT := describe[linearcorrelation](X1, X2);
end do;
RETURN(rho);
end;
```

Figure 9.6 Maple procedure to estimate the bivariate normal correlation required for the NORTA method.

general $X_1 = F_1^{-1}[\Phi(Z_1)]$ and $X_2 = F_2^{-1}[\Phi(Z_2)]$ will have a correlation $\rho_X \neq \rho$. The difference is often small, but not always.

The second issue is more critical, because in input-modeling problems we want to specify the bivariate or lag-1 correlation. Thus, we need to find the bivariate normal correlation ρ that gives us the input correlation ρ_X that we want (recall that we specify the time series model via the lag-1 correlation, $\rho_X = \text{corr}(X_t, X_{t+1})$). There has been much research on this problem, including Cario and Nelson [1996, 1998] and Biller and Nelson [2003]. Fortunately, it has been shown that ρ_X is a nondecreasing function of ρ, and ρ and ρ_X will always have the same sign. Thus, we can do a relatively simple search based on the following algorithm:

Step 1. Set $\rho = \rho_X$ to start.

Step 2. Generate a large number of bivariate normal pairs (Z_1, Z_2) with correlation ρ, and transform them into (X_1, X_2)'s, using the NORTA transformation.

Step 3. Compute the sample correlation between (X_1, X_2), using Equation (9.24), and call it $\hat{\rho}_I$. If $\hat{\rho}_I > \rho_X$, then reduce ρ and go to Step 2; if $\hat{\rho}_I < \rho_X$, then increase ρ and go to Step 2. If $\hat{\rho}_I \approx \rho_X$ then stop.

Example 9.22

Suppose we needed X_1 to have an exponential distribution with mean 1, X_2 to have a beta distribution with $\beta_1 = 1$, $\beta_2 = 1/2$, and the two of them to have correlation $\rho_X = 0.45$. Figure 9.6 shows a procedure in Maple that will estimate the required value of ρ. In the procedure, n is the number of sample pairs used to estimate the correlation. Running this procedure with n set to 1000 gives $\rho = 0.52$.

9.8 SUMMARY

Input-data collection and analysis require major time and resource commitments in a discrete-event simulation project. However, regardless of the validity or sophistication of the simulation model, unreliable inputs can lead to outputs whose subsequent interpretation could result in faulty recommendations.

This chapter discussed four steps in the development of models of input data: collecting the raw data, identifying the underlying statistical distribution, estimating the parameters, and testing for goodness of fit.

Some suggestions were given for facilitating the data-collection step. However, experience, such as that obtained by completing any of Exercises 1 through 5, will increase awareness of the difficulty of problems that can arise in data collection and of the need for planning.

Once the data have been collected, a statistical model should be hypothesized. Constructing a histogram is very useful at this point if sufficient data are available. A distribution based on the underlying process and on the shape of the histogram can usually be selected for further investigation.

The investigation proceeds with the estimation of parameters for the hypothesized distribution. Suggested estimators were given for distributions used often in simulation. In a number of instances, these are functions of the sample mean and sample variance.

The last step in the process is the testing of the distributional hypothesis. The $q - q$ plot is a useful graphical method for assessing fit. The Kolmogorov–Smirnov, chi-square, and Anderson–Darling goodness-of-fit tests can be applied to many distributional assumptions. When a distributional assumption is rejected, another distribution is tried. When all else fails, the empirical distribution could be used in the model.

Unfortunately, in some situations, a simulation study must be undertaken when there is not time or resources to collect data on which to base input models. When this happens, the analyst must use any available

information—such as manufacturer specifications and expert opinion—to construct the input models. When input models are derived without the benefit of data, it is particularly important to examine the sensitivity of the results to the models chosen.

Many, but not all, input processes can be represented as sequences of independent and identically distributed random variables. When inputs should exhibit dependence, then multivariate-input models are required. The bivariate normal distribution (and more generally the multivariate normal distribution) is often used to represent a finite number of dependent random variables. Time-series models are useful for representing a (conceptually infinite) sequence of dependent inputs. The NORTA transformation facilitates developing multivariate-input models with marginal distributions that are not normal.

REFERENCES

BILLER, B., AND B. L. NELSON [2003], "Modeling and Generating Multivariate Time Series with Arbitrary Marginals Using an Autoregressive Technique," *ACM Transactions on Modeling and Computer Simulation*, Vol. 13, pp. 211–237.

BRATLEY, P., B. L. FOX, AND L. E. SCHRAGE [1987], *A Guide to Simulation*, 2d ed., Springer-Verlag, New York.

CARIO, M. C., AND B. L. NELSON [1996], "Autoregressive to Anything: Time-Series Input Processes for Simulation," *Operations Research Letters*, Vol. 19, pp. 51–58.

CARIO, M. C., AND B. L. NELSON [1998], "Numerical Methods for Fitting and Simulating Autoregressive-to-Anything Processes," *INFORMS Journal on Computing*, Vol. 10, pp. 72–81.

CHOI, S. C., AND R. WETTE [1969], "Maximum Likelihood Estimation of the Parameters of the Gamma Distribution and Their Bias," *Technometrics*, Vol. 11, No. 4, pp. 683–890.

CHAMBERS, J. M., CLEVELAND, W. S., AND TUKEY, P. A. [1983], *Graphical Methods for Data Analysis*, CRC Press, Boca Raton, FL.

CONNOVER, W. J. [1998], *Practical Nonparametric Statistics*, 3d ed., Wiley, New York.

DURBIN, J. [1975], "Kolmogorov–Smirnov Tests When Parameters Are Estimated with Applications to Tests of Exponentiality and Tests on Spacings," *Biometrika*, Vol. 65, pp. 5–22.

FISHMAN, G. S. [1973], *Concepts and Methods in Discrete Event Digital Simulation*, Wiley, New York.

GUMBEL, E. J. [1943], "On the Reliability of the Classical Chi-squared Test," *Annals of Mathematical Statistics*, Vol. 14, pp. 253ff.

HINES, W. W., D. C. MONTGOMERY, D. M. GOLDSMAN, AND C. M. BORROR [2002], *Probability and Statistics in Engineering and Management Science*, 4th ed., Wiley, New York.

JOHNSON, M. A., S. LEE, AND J. R. WILSON [1994], "NPPMLE and NPPSIM: Software for Estimating and Simulating Nonhomogeneous Poisson Processes Having Cyclic Behavior," *Operations Research Letters*, Vol. 15, pp. 273–282.

JOHNSON, M. E. [1987], *Multivariate Statistical Simulation*, Wiley, New York.

LAW, A. M., AND W. D. KELTON [2000], *Simulation Modeling & Analysis*, 3d ed., McGraw–Hill, New York.

LILLIEFORS, H. W. [1967], "On the Kolmogorov–Smirnov Test for Normality with Mean and Variance Unknown," *Journal of the American Statistical Association*, Vol. 62, pp. 339–402.

LILLIEFORS, H. W. [1969], "On the Kolmogorov–Smirnov Test for the Exponential Distribution with Mean Unknown," *Journal of the American Statistical Association*, Vol. 64, pp. 387–389.

MANN, H. B., AND A. WALD [1942], "On the Choice of the Number of Intervals in the Application of the Chi-squared Test," *Annals of Mathematical Statistics*, Vol. 18, p. 50ff.

NELSON, B. L., AND M. YAMNITSKY [1998], "Input Modeling Tools for Complex Problems," in *Proceedings of the 1998 Winter Simulation Conference*, eds. D. Medeiros, E. Watson, J. Carson, and M. Manivannan, pp. 105–112, The Institute for Electrical and Electronics Engineers, Piscataway, NJ.

PEGDEN, C. D., R. E. SHANNON, AND R. P. SADOWSKI [1995], *Introduction to Simulation Using SIMAN*, 2d ed. McGraw–Hill, New York.

STUART, A., J. K. ORD, AND E. ARNOLD [1998], *Kendall's Advanced Theory of Statistics*, 6th ed., Vol. 2, Oxford University Press, Oxford, NY.

EXERCISES

1. In a college library, collect the following information at the books return counter:

   arrival of students for returning books
   service time taken by the counter clerk

   Consolidate the data collected and verify whether it follows any standard distribution. (Prior permission from concerned authorities may be required.)

2. Go to a bank having single window operation. Collect information on arrival of customers, service time, etc. The type of transaction may vary from customer to customer. From service times observed, classify according to the type of transaction and fit arrival and service parameters separately for each type of transaction. (Prior permission from concerned authorities may be required.)

3. Go to a major traffic intersection, and record the interarrival-time distributions from each direction. Some arrivals want to go straight, some turn left, some turn right. The interarrival-time distribution varies during the day and by day of the week. Every now and then an accident occurs.

4. Go to a grocery store, and construct the interarrival and service distributions at the checkout counters. These distributions might vary by time of day and by day of week. Record, also, the number of service channels available at all times. (Make sure that the management gives permission to perform this study.)

5. Go to a laundromat, and "relive" the authors' data-collection experience discussed in Example 9.1. (Make sure that the management gives permission to perform this study.)

6. Draw the pdf of normal distribution with $\mu = 6$, $\sigma = 3$.

7. On one figure, draw the pdfs of the Erlang distribution where $\theta = 1/2$ and $k = 1, 2, 4$, and 8

8. On one figure, draw the pdfs of the Erlang distribution where $\theta = 2$ and $k = 1, 2, 4$, and 8.

9. Draw the pdf of Poisson distribution with $\alpha = 3$, 5, and 6.

10. Draw the exponential pdf with $\lambda = 0.5$. In the same sheet, draw the exponential pdf with $\lambda = 1.5$.

11. Draw the exponential pdf with $\lambda = 1$. In the same sheet, draw the exponential pdf with $\lambda = 3$.

12. The following data are generated randomly from a gamma distribution:

1.691	1.437	8.221	5.976
1.116	4.435	2.345	1.782
3.810	4.589	5.313	10.90
2.649	2.432	1.581	2.432
1.843	2.466	2.833	2.361

   Compute the maximum-likelihood estimators $\hat{\beta}$ and $\hat{\theta}$.

13. The following data are generated randomly from a Weibull distribution where $\nu = 0$:

7.936	5.224	3.937	6.513
4.599	7.563	7.172	5.132
5.259	2.759	4.278	2.696
6.212	2.407	1.857	5.002
4.612	2.003	6.908	3.326

Compute the maximum-likelihood estimators $\hat{\alpha}$ and $\hat{\beta}$. (This exercise requires a programmable calculator, a computer, or a lot of patience.)

14. Time between failures (in months) of a particular bearing is assumed to follow normal distribution. The data collected over 50 failures are

11.394	10.728	6.680	8.050	8.382
8.740	8.287	7.979	5.857	13.521
12.000	9.496	9.248	6.529	12.137
11.383	8.135	11.752	10.040	8.615
8.686	6.416	9.987	11.282	4.732
9.344	7.019	6.735	12.176	4.247
10.099	6.254	5.557	9.376	5.780
7.129	7.835	9.648	4.381	5.801
8.334	9.454	8.486	7.256	10.963
10.544	10.433	10.425	10.078	7.709

Using Kolmogorov–Smirnov test, check whether the distribution follows normal.

15. Show that the Kolmogorov–Smirnov test statistic for Example 9.16 is $D = 0.1054$.

16. Records pertaining to the monthly number of job-related injuries at an underground coalmine were being studied by a federal agency. The values for the past 100 months were as follows:

Injuries per Month	*Frequency of Occurrence*
0	35
1	40
2	13
3	6
4	4
5	1
6	1

(a) Apply the chi-square test to these data to test the hypothesis that the underlying distribution is Poisson. Use the level of significance $\alpha = 0.05$.

(b) Apply the chi-square test to these data to test the hypothesis that the distribution is Poisson with mean 1.0. Again let $\alpha = 0.05$.

(c) What are the differences between parts (a) and (b), and when might each case arise?

17. The interarrival time of tools for repair to a service station is assumed to follow exponential with $\lambda = 1$. The data collected from 50 such arrivals are

1.299	0.234	1.182	0.943	0.038
0.010	2.494	1.104	0.330	0.324
0.059	1.375	1.660	1.748	0.706
2.198	0.537	0.904	1.910	0.387
3.508	2.784	0.237	1.137	0.990

1.002	1.594	0.404	1.467	0.905
1.000	0.143	0.697	0.442	0.395
0.861	1.952	0.016	0.167	2.245
0.812	1.035	0.688	0.565	0.155
0.465	0.451	0.507	0.224	1.441

Based on appropriate test, check whether the assumption is valid.

18. The time spent by customers (in minutes) based on a study conducted in the college canteen is

13.125	12.972	18.985	12.041	14.658
14.151	17.541	17.251	13.400	15.559
16.365	18.946	11.154	11.159	14.883
13.650	15.336	16.990	18.265	18.719
13.763	18.518	16.493	15.869	13.291
16.643	16.712	12.759	14.926	14.412
21.285	13.299	16.589	13.887	15.853
12.995	19.540	17.761	16.290	14.624
14.300	8.497	19.149	14.035	17.076
18.778	11.186	16.263	14.438	15.741

Using appropriate methods, determine how the time is distributed.

19. The time required for the transmission of a message (in minutes) is sampled electronically at a communications center. The last 50 values in the sample are as follows:

7.936	4.612	2.407	4.278	5.132
4.599	5.224	2.003	1.857	2.696
5.259	7.563	3.937	6.908	5.002
6.212	2.759	7.172	6.513	3.326
8.761	4.502	6.188	2.566	5.515
3.785	3.742	4.682	4.346	5.359
3.535	5.061	4.629	5.298	6.492
3.502	4.266	3.129	1.298	3.454
5.289	6.805	3.827	3.912	2.969
4.646	5.963	3.829	4.404	4.924

How are the transmission times distributed? Develop and test an appropriate model.

20. The time spent (in minutes) by a customer in a bus stop awaiting to board a bus is

1.07	10.69	11.81	12.81	13.75
7.19	16.25	12.32	6.72	13.92
6.62	6.10	20.21	9.58	14.13
11.27	3.00	12.53	8.01	14.46
7.28	14.12	7.59	9.33	11.16

10.38	11.13	3.56	4.57	17.85
11.97	16.96	5.04	13.77	6.60
14.34	11.70	11.95	9.24	9.65
13.88	8.93	12.72	9.00	0.89
13.39	10.37	20.53	9.92	3.49

Using appropriate methods, determine how the time is distributed.

21. Daily demands for transmission overhaul kits for the D-3 dragline were maintained by Earth Moving Tractor Company, with the following results:

0	2	0	0	0
1	0	1	1	1
0	1	0	0	0
2	0	1	0	1
0	1	0	0	2
1	0	1	0	0
0	0	0	0	0
1	0	1	0	1
0	0	3	0	1
1	0	0	0	0

How are the daily demands distributed? Develop and test an appropriate model.

22. A simulation is to be conducted of a job shop that performs two operations: milling and planing, in that order. It would be possible to collect data about processing times for each operation, then generate random occurrences from each distribution. However, the shop manager says that the times might be related; large milling jobs take lots of planing. Data are collected for the next 25 orders, with the following results in minutes:

Order	*Milling Time (Minutes)*	*Planing Time (Minutes)*	*Order*	*Milling Time (Minutes)*	*Planing Time (Minutes)*
1	12.3	10.6	14	24.6	16.6
2	20.4	13.9	15	28.5	21.2
3	18.9	14.1	16	11.3	9.9
4	16.5	10.1	17	13.3	10.7
5	8.3	8.4	18	21.0	14.0
6	6.5	8.1	19	19.5	13.0
7	25.2	16.9	20	15.0	11.5
8	17.7	13.7	21	12.6	9.9
9	10.6	10.2	22	14.3	13.2
10	13.7	12.1	23	17.0	12.5
11	26.2	16.0	24	21.2	14.2
12	30.4	18.9	25	28.4	19.1
13	9.9	7.7			

(a) Plot milling time on the horizontal axis and planing time on the vertical axis. Do these data seem dependent?

(b) Compute the sample correlation between milling time and planing time.
(c) Fit a bivariate normal distribution to these data.

23. Write a computer program to compute the maximum-likelihood estimators ($\hat{\alpha}$, $\hat{\beta}$) of the Weibull distribution. Inputs to the program should include the sample size, n; the observations, $x_1, x_2, \ldots, x_n$; a stopping criterion, ϵ (stop when $|f(\hat{\beta}_j)| \leq \epsilon$); and a print option, OPT (usually set = 0). Output would be the estimates $\hat{\alpha}$ and $\hat{\beta}$. If OPT = 1, additional output would be printed, as in Table 9.4, showing convergence. Make the program as "user friendly" as possible.

24. Examine a computer-software library or simulation-support environment to which you have access. Obtain documentation on data-analysis software that would be useful in solving exercises 7 through 24. Use the software as an aid in solving selected problems.

25. The duration of calls in minutes over a telephone line is

2.058 6.407 0.565 0.641 5.989 0.435 0.278 3.447 11.461 1.658 2.913 2.689 4.747 2.587

Develop an input model for the call duration data.

26. The following data represent the time to perform transactions in a bank, measured in minutes: 0.740, 1.28, 1.46, 2.36, 0.354, 0.750, 0.912, 4.44, 0.114, 3.08, 3.24, 1.10, 1.59, 1.47, 1.17, 1.27, 9.12, 11.5, 2.42, 1.77. Develop an input model for these data.

27. Two types of jobs (A and B) are released to the input buffer of a job shop as orders arrive, and the arrival of orders is uncertain. The following data are available from the last week of production:

Day	*Number of Jobs*	*Number of A's*
1	83	53
2	93	62
3	112	66
4	65	41
5	78	55

Develop an input model for the number of new arrivals of each type each day.

28. The following data are available on the processing time at a machine (in minutes): 0.64, 0.59, 1.1, 3.3, 0.54, 0.04, 0.45, 0.25, 4.4, 2.7, 2.4, 1.1, 3.6, 0.61, 0.20, 1.0, 0.27, 1.7, 0.04, 0.34. Develop an input model for the processing time.

29. In the process of the development of an inventory simulation model, demand for a component is

1	2	3	4	3	5	4	3
4	4	6	6	5	4	6	4
5	7	5	5	7	1	5	2
3	4	3	4	2	8	7	2
3	8	4	4	5	3	1	6

Using appropriate model, identify how the demand is distributed.

30. Using the web, research some of the input-modeling software packages mentioned in this chapter. What are their features? What distributions do they include?

10

Verification and Validation of Simulation Models

One of the most important and difficult tasks facing a model developer is the verification and validation of the simulation model. The engineers and analysts who use the model outputs to aid in making design recommendations and the managers who make decisions based on these recommendations—justifiably look upon a model with some degree of skepticism about its validity. It is the job of the model developer to work closely with the end users throughout the period of development and validation to reduce this skepticism and to increase the model's credibility.

The goal of the validation process is twofold: (1) to produce a model that represents true system behavior closely enough for the model to be used as a substitute for the actual system for the purpose of experimenting with the system, analyzing system behavior, and predicting system performance; and (2) to increase to an acceptable level the credibility of the model, so that the model will be used by managers and other decision makers.

Validation should not be seen as an isolated set of procedures that follows model development, but rather as an integral part of model development. Conceptually, however, the verification and validation process consists of the following components:

1. Verification is concerned with building the model correctly. It proceeds by the comparison of the conceptual model to the computer representation that implements that conception. It asks the questions: Is the model implemented correctly in the simulation software? Are the input parameters and logical structure of the model represented correctly?
2. Validation is concerned with building the correct model. It attempts to confirm that a model is an accurate representation of the real system. Validation is usually achieved through the calibration of the model, an iterative process of comparing the model to actual system behavior and using the

discrepancies between the two, and the insights gained, to improve the model. This process is repeated until model accuracy is judged to be acceptable.

This chapter describes methods that have been recommended and used in the verification and validation process. Most of the methods are informal subjective comparisons; a few are formal statistical procedures. The use of the latter procedures involves issues related to output analysis, the subject of Chapters 11 and 12. Output analysis refers to analysis of the data produced by a simulation and to drawing inferences from these data about the behavior of the real system. To summarize their relationship, validation is the process by which model users gain confidence that output analysis is making valid inferences about the real system under study.

Many articles and chapters in texts have been written on verification and validation. For discussion of the main issues, the reader is referred to Balci [1994, 1998, 2003], Carson [1986, 2002], Gass [1983], Kleijnen [1995], Law and Kelton [2000], Naylor and Finger [1967], Oren [1981], Sargent [2003], Shannon [1975], and van Horn [1969, 1971]. For statistical techniques relevant to various aspects of validation, the reader can obtain the foregoing references plus those by Balci and Sargent [1982a,b; 1984a], Kleijnen [1987], and Schruben [1980]. For case studies in which validation is emphasized, the reader is referred to Carson *et al.* [1981a,b], Gafarian and Walsh [1970], Kleijnen [1993], and Shechter and Lucas [1980]. Bibliographies on validation have been published by Balci and Sargent [1984b] and by Youngblood [1993].

10.1 MODEL BUILDING, VERIFICATION, AND VALIDATION

The first step in model building consists of observing the real system and the interactions among their various components and of collecting data on their behavior. But observation alone seldom yields sufficient understanding of system behavior. Persons familiar with the system, or any subsystem, should be questioned to take advantage of their special knowledge. Operators, technicians, repair and maintenance personnel, engineers, supervisors, and managers understand certain aspects of the system that might be unfamiliar to others. As model development proceeds, new questions may arise, and the model developers will return to this step of learning true system structure and behavior.

The second step in model building is the construction of a conceptual model—a collection of assumptions about the components and the structure of the system, plus hypotheses about the values of model input parameters. As is illustrated by Figure 10.1, conceptual validation is the comparison of the real system to the conceptual model.

The third step is the implementation of an operational model, usually by using simulation software and incorporating the assumptions of the conceptual model into the worldview and concepts of the simulation software. In actuality, model building is not a linear process with three steps. Instead, the model builder will return to each of these steps many times while building, verifying, and validating the model. Figure 10.1 depicts the ongoing model building process, in which the need for verification and validation causes continual comparison of the real system to the conceptual model and to the operational model and induces repeated modification of the model to improve its accuracy.

10.2 VERIFICATION OF SIMULATION MODELS

The purpose of model verification is to assure that the conceptual model is reflected accurately in the operational model. The conceptual model quite often involves some degree of abstraction about system operations or some amount of simplification of actual operations. Verification asks the following question: Is the conceptual model (assumptions about system components and system structure, parameter values, abstractions, and simplifications) accurately represented by the operational model?

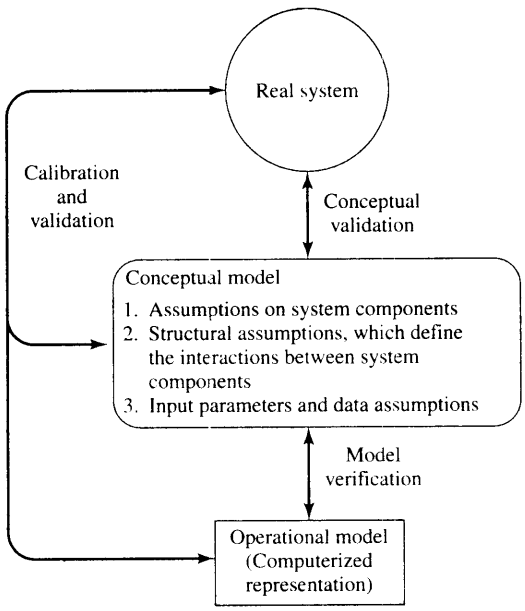


Figure 10.1 Model building, verification, and validation.

Many common-sense suggestions can be given for use in the verification process:

1. Have the operational model checked by someone other than its developer, preferably an expert in the simulation software being used.
2. Make a flow diagram that includes each logically possible action a system can take when an event occurs, and follow the model logic for each action for each event type. (An example of a logic flow diagram is given in Figures 2.2 and 2.3 for the model of a single-server queue.)
3. Closely examine the model output for reasonableness under a variety of settings of the input parameters. Have the implemented model display a wide variety of output statistics, and examine all of them closely.
4. Have the operational model print the input parameters at the end of the simulation, to be sure that these parameter values have not been changed inadvertently.
5. Make the operational model as self-documenting as possible. Give a precise definition of every variable used and a general description of the purpose of each submodel, procedure (or major section of code), component, or other model subdivision.
6. If the operational model is animated, verify that what is seen in the animation imitates the actual system. Examples of errors that can be observed through animation are automated guided vehicles (AGVs) that pass through one another on a unidirectional path or at an intersection and entities that disappear (unintentionally) during a simulation.
7. The Interactive Run Controller (IRC) or debugger is an essential component of successful simulation model building. Even the best of simulation analysts makes mistakes or commits logical errors when building a model. The IRC assists in finding and correcting those errors in the following ways:
   (a) The simulation can be monitored as it progresses. This can be accomplished by advancing the simulation until a desired time has elapsed, then displaying model information at that time. Another possibility is to advance the simulation until a particular condition is in effect, and then display information.

(b) Attention can be focused on a particular entity line, of code, or procedure. For instance, every time that an entity enters a specified procedure, the simulation will pause so that information can be gathered. As another example, every time that a specified entity becomes active, the simulation will pause.

(c) Values of selected model components can be observed. When the simulation has paused, the current value or status of variables, attributes, queues, resources, counters, and so on can be observed.

(d) The simulation can be temporarily suspended, or paused, not only to view information, but also to reassign values or redirect entities.

8. Graphical interfaces are recommended for accomplishing verification and validation [Borts-cheller and Saulnier, 1992]. The graphical representation of the model is essentially a form of self-documentation. It simplifies the task of understanding the model.

These suggestions are basically the same ones any software engineer would follow.

Among these common-sense suggestions, one that is very easily implemented, but quite often overlooked, especially by students who are learning simulation, is a close and thorough examination of model output for reasonableness (suggestion 3). For example, consider a model of a complex network of queues consisting of many service centers in series and parallel configurations. Suppose that the modeler is interested mainly in the response time, defined as the time required for a customer to pass through a designated part of the network. During the verification (and calibration) phase of model development, it is recommended that the program collect and print out many statistics in addition to response times, such as utilizations of servers and time-average number of customers in various subsystems. Examination of the utilization of a server, for example, might reveal that it is unreasonably low (or high), a possible error that could be caused by wrong specification of mean service time, or by a mistake in model logic that sends too few (or too many) customers to this particular server, or by any number of other possible parameter misspecifications or errors in logic.

In a simulation language that automatically collects many standard statistics (average queue lengths, average waiting times, etc.), it takes little or no extra programming effort to display almost all statistics of interest. The effort required can be considerably greater in a general-purpose language such as Java, C, or C++, which do not have statistics-gathering capabilities to aid the programmer.

Two sets of statistics that can give a quick indication of model reasonableness are *current contents* and *total count*. These statistics apply to any system having items of some kind flowing through it, whether these items be called customers, transactions, inventory, or vehicles. "Current contents" refers to the number of items in each component of the system at a given time. "Total count" refers to the total number of items that have entered each component of the system by a given time. In some simulation software, these statistics are kept automatically and can be displayed at any point in simulation time. In other simulation software, simple counters might have to be added to the operational model and displayed at appropriate times. If the current contents in some portion of the system are high, this condition indicates that a large number of entities are delayed. If the output is displayed for successively longer simulation run times and the current contents tend to grow in a more or less linear fashion, it is highly likely that a queue is unstable and that the server(s) will fall further behind as time continues. This indicates possibly that the number of servers is too small or that a service time is misspecified. (Unstable queues were discussed in Chapter 6.) On the other hand, if the total count for some subsystem is zero, this indicates that no items entered that subsystem—again, a highly suspect occurrence. Another possibility is that the current count and total count are equal to one. This could indicate that an entity has captured a resource, but never freed that resource. Careful evaluation of these statistics for various run lengths can aid in the detection of mistakes in model logic and data misspecifications. Checking for output reasonableness will usually fail to detect the more subtle errors, but it is one of the quickest ways to discover gross errors. To aid in error detection, it is best for the model developer to forecast a reasonable range for the value of selected output statistics before making a run of the model. Such a forecast reduces the possibility of rationalizing a discrepancy and failing to investigate the cause of unusual output.

For certain models, it is possible to consider more than whether a particular statistic is reasonable. It is possible to compute certain long-run measures of performance. For example, as seen in Chapter 6, the analyst can compute the long-run server utilization for a large number of queueing systems without any special assumptions regarding interarrival or service-time distributions. Typically, the only information needed is the network configuration, plus arrival and service rates. Any measure of performance that can be computed analytically and then compared to its simulated counterpart provides another valuable tool for verification. Presumably, the objective of the simulation is to estimate some measure of performance, such as mean response time, that cannot be computed analytically; but, as illustrated by the formulas in Chapter 6 for a number of special queues (*M/M/*1, *M/G/*1, etc.), all the measures of performance in a queueing system are interrelated. Thus, if a simulation model is predicting one measure (such as utilization) correctly, then confidence in the model's predictive ability for other related measures (such as response time) is increased (even though the exact relation between the two measures is, of course, unknown in general and varies from model to model). Conversely, if a model incorrectly predicts utilization, its prediction of other quantities, such as mean response time, is highly suspect.

Another important way to aid the verification process is the oft-neglected documentation phase. If a model builder writes brief comments in the operational model, plus definitions of all variables and parameters, plus descriptions of each major section of the operational model, it becomes much simpler for someone else, or the model builder at a later date, to verify the model logic. Documentation is also important as a means of clarifying the logic of a model and verifying its completeness.

A more sophisticated technique is the use of a trace. In general, a trace is a detailed computer printout which gives the value of every variable (in a specified set of variables) in a computer program, every time that one of these variables changes in value. A trace designed specifically for use in a simulation program would give the value of selected variables each time the simulation clock was incremented (i.e., each time an event occurred). Thus, a simulation trace is nothing more than a detailed printout of the state of the simulation model as it changes over time.

Example 10.1

When verifying the operational model (in a general purpose language such as FORTRAN, Pascal, C or C++, or most simulation languages) of the single-server queue model of Example 2.1, an analyst made a run over 16 units of time and observed that the time-average length of the waiting line was $\hat{L}_Q = 0.4375$ customer, which is certainly reasonable for a short run of only 16 time units. Nevertheless, the analyst decided that a more detailed verification would be of value.

The trace in Figure 10.2 gives the hypothetical printout from simulation time CLOCK = 0 to CLOCK = 16 for the simple single-server queue of Example 2.1. This example illustrates how an error can be found with a trace, when no error was apparent from the examination of the summary output statistics (such as $\hat{L}_Q$). Note that, at simulation time CLOCK = 3, the number of customers in the system is NCUST = 1, but the server is idle (STATUS = 0). The source of this error could be incorrect logic, or simply not setting the attribute STATUS to the value 1 (when coding in a general purpose language or most simulation languages).

In any case, the error must be found and corrected. Note that the less sophisticated practice of examining the summary measures, or output, did not detect the error. By using equation (6.1), the reader can verify that $\hat{L}_Q$ was computed correctly from the data ($\hat{L}_Q$ is the time-average value of NCUST minus STATUS):

$$\hat{L}_Q = \frac{(0-0)3+(1-0)2+(0-0)6+(1-0)1+(2-1)4}{3+2+6+1+4}$$

$$= \frac{7}{16} = 0.4375$$

as previously mentioned. Thus, the output measure, $\hat{L}_Q$, had a reasonable value and was computed correctly from the data, but its value was indeed wrong because the attribute STATUS was not assuming correct

Definition of Variables:

CLOCK = Simulation clock
EVTYP = Event type (start, arrival, departure, or stop)
NCUST = Number of customers in system at time 'CLOCK'
STATUS = Status of server (1–busy, 0–idle)

State of System Just After the Named Event Occurs:

CLOCK = 0	EVTYP = 'Start'	NCUST = 0	STATUS = 0
CLOCK = 3	EVTYP = 'Arrival'	NCUST = 1	STATUS = 0
CLOCK = 5	EVTYP = 'Depart'	NCUST = 0	STATUS = 0
CLOCK = 11	EVTYP = 'Arrival'	NCUST = 1	STATUS = 0
CLOCK = 12	EVTYP = 'Arrival'	NCUST = 2	STATUS = 1
CLOCK = 16	EVTYP = 'Depart'	NCUST = 1	STATUS = 1
.			
.			
.			

Figure 10.2 Simulation Trace of Example 2.1.

values. As is seen from Figure 10.2, a trace yields information on the actual history of the model that is more detailed and informative than the summary measures alone.

Most simulation software has a built-in capability to conduct a trace without the programmer having to do any extensive programming. In addition, a 'print' or 'write' statement can be used to implement a tracing capability in a general-purpose language.

As can be easily imagined, a trace over a large span of simulation time can quickly produce an extremely large amount of computer printout, which would be extremely cumbersome to check in detail for correctness. The purpose of the trace is to verify the correctness of the computer program by making detailed paper-and-pencil calculations. To make this practical, a simulation with a trace is usually restricted to a very short period of time. It is desirable, of course, to ensure that each type of event (such as ARRIVAL) occurs at least once, so that its consequences and effect on the model can be checked for accuracy. If an event is especially rare in occurrence, it may be necessary to use artificial data to force it to occur during a simulation of short duration. This is legitimate, as the purpose is to verify that the effect on the system of the rare event is as intended.

Some software allows a selective trace. For example, a trace could be set for specific locations in the model or could be triggered to begin at a specified simulation time. Whenever an entity goes through the designated locations, the simulation software writes a time-stamped message to a trace file. Some simulation software allows tracing a selected entity; any time the designated entity becomes active, the trace is activated and time-stamped messages are written. This trace is very useful in following one entity through the entire model. Another example of a selective trace is to set it for the occurrence of a particular condition. For example, whenever the queue before a certain resource reaches five or more, turn on the trace. This allows running the simulation until something unusual occurs, then examining the behavior from that point forward in time. Different simulation software packages support tracing to various extents. In practice, it is often implemented by the model developer by adding printed messages at appropriate points into a model.

Of the three classes of techniques—the common-sense techniques, thorough documentation, and traces—it is recommended that the first two always be carried out. Close examination of model output for reasonableness is especially valuable and informative. A generalized trace may provide voluminous data, far more than can be used or examined carefully. A selective trace can provide useful information on key model components and keep the amount of data to a manageable level.

10.3 CALIBRATION AND VALIDATION OF MODELS

Verification and validation, although conceptually distinct, usually are conducted simultaneously by the modeler. Validation is the overall process of comparing the model and its behavior to the real system and its behavior. Calibration is the iterative process of comparing the model to the real system, making adjustments (or even major changes) to the model, comparing the revised model to reality, making additional adjustments, comparing again, and so on. Figure 10.3 shows the relationship of model calibration to the overall validation process. The comparison of the model to reality is carried out by a variety of tests—some subjective, others objective. Subjective tests usually involve people, who are knowledgeable about one or more aspects of the system, making judgments about the model and its output. Objective tests always require data on the system's behavior, plus the corresponding data produced by the model. Then one or more statistical tests are performed to compare some aspect of the system data set with the same aspect of the model data set. This iterative process of comparing model with system and then revising both the conceptual and operational models to accommodate any perceived model deficiencies is continued until the model is judged to be sufficiently accurate.

A possible criticism of the calibration phase, were it to stop at this point, is that the model has been validated only for the one data set used—that is, the model has been "fitted" to one data set. One way to alleviate this criticism is to collect a new set of system data (or to reserve a portion of the original system data) to be used at this final stage of validation. That is, after the model has been calibrated by using the original system data set, a "final" validation is conducted, using the second system data set. If unacceptable discrepancies between the model and the real system are discovered in the "final" validation effort, the modeler must return to the calibration phase and modify the model until it becomes acceptable.

Validation is not an either/or proposition—no model is ever totally representative of the system under study. In addition, each revision of the model, as pictured in Figure 10.3, involves some cost, time, and effort. The modeler must weigh the possible, but not guaranteed, increase in model accuracy versus the cost of increased validation effort. Usually, the modeler (and model users) have some maximum discrepancy between model predictions and system behavior that would be acceptable. If this level of accuracy cannot be obtained within the budget constraints, either expectations of model accuracy must be lowered, or the model must be abandoned.

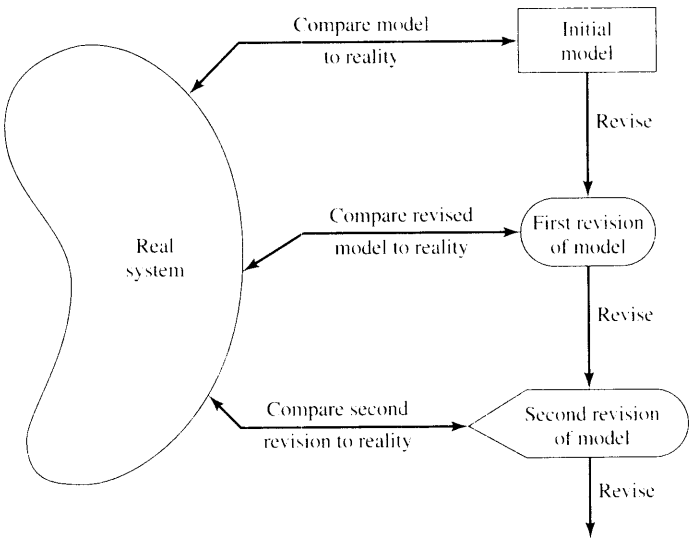


Figure 10.3 Iterative process of calibrating a model.

As an aid in the validation process, Naylor and Finger [1967] formulated a three-step approach that has been widely followed:

1. Build a model that has high face validity.
2. Validate model assumptions.
3. Compare the model input–output transformations to corresponding input–output transformations for the real system.

The next five subsections investigate these three steps in detail.

10.3.1 Face Validity

The first goal of the simulation modeler is to construct a model that appears reasonable on its face to model users and others who are knowledgeable about the real system being simulated. The potential users of a model should be involved in model construction from its conceptualization to its implementation, to ensure that a high degree of realism is built into the model through reasonable assumptions regarding system structure and through reliable data. Potential users and knowledgeable persons can also evaluate model output for reasonableness and can aid in identifying model deficiencies. Thus, the users can be involved in the calibration process as the model is improved iteratively by the insights gained from identification of the initial model deficiencies. Another advantage of user involvement is the increase in the model's perceived validity, or credibility, without which a manager would not be willing to trust simulation results as a basis for decision making.

Sensitivity analysis can also be used to check a model's face validity. The model user is asked whether the model behaves in the expected way when one or more input variables is changed. For example, in most queueing systems, if the arrival rate of customers (or demands for service) were to increase, it would be expected that utilizations of servers, lengths of lines, and delays would tend to increase (although by how much might well be unknown). From experience and from observations on the real system (or similar related systems), the model user and model builder would probably have some notion at least of the direction of change in model output when an input variable is increased or decreased. For most large-scale simulation models, there are many input variables and thus many possible sensitivity tests. The model builder must attempt to choose the most critical input variables for testing if it is too expensive or time consuming to vary all input variables. If real system data are available for at least two settings of the input parameters, objective scientific sensitivity tests can be conducted via appropriate statistical techniques.

10.3.2 Validation of Model Assumptions

Model assumptions fall into two general classes: structural assumptions and data assumptions. Structural assumptions involve questions of how the system operates and usually involve simplifications and abstractions of reality. For example, consider the customer queueing and service facility in a bank. Customers can form one line, or there can be an individual line for each teller. If there are many lines, customers could be served strictly on a first-come–first-served basis, or some customers could change lines if one line is moving faster. The number of tellers could be fixed or variable. These structural assumptions should be verified by actual observation during appropriate time periods and by discussions with managers and tellers regarding bank policies and actual implementation of these policies.

Data assumptions should be based on the collection of reliable data and correct statistical analysis of the data. (Example 9.1 discussed similar issues for a model of a laundromat.) For example, in the bank study previously mentioned, data were collected on

1. interarrival times of customers during several 2-hour periods of peak loading ("rush-hour" traffic);
2. interarrival times during a slack period;

3. service times for commercial accounts;
4. service times for personal accounts.

The reliability of the data was verified by consultation with bank managers, who identified typical rush hours and typical slack times. When combining two or more data sets collected at different times, data reliability can be further enhanced by objective statistical tests for homogeneity of data. (Do two data sets $\{X_i\}$ and $\{Y_i\}$ on service times for personal accounts, collected at two different times, come from the same parent population? If so, the two sets can be combined.) Additional tests might be required, to test for correlation in the data. As soon as the analyst is assured of dealing with a random sample (i.e., correlation is not present), the statistical analysis can begin.

The procedures for analyzing input data from a random sample were discussed in detail in Chapter 9. Whether done manually or by special-purpose software, the analysis consists of three steps:

1. Identify an appropriate probability distribution.
2. Estimate the parameters of the hypothesized distribution.
3. Validate the assumed statistical model by a goodness-of-fit test, such as the chi-square or Kolmogorov–Smirnov test, and by graphical methods.

The use of goodness-of-fit tests is an important part of the validation of data assumptions.

10.3.3 Validating Input–Output Transformations

The ultimate test of a model, and in fact the only objective test of the model as a whole, is the model's ability to predict the future behavior of the real system when the model input data match the real inputs and when a policy implemented in the model is implemented at some point in the system. Furthermore, if the level of some input variables (e.g., the arrival rate of customers to a service facility) were to increase or decrease, the model should accurately predict what would happen in the real system under similar circumstances. In other words, the structure of the model should be accurate enough for the model to make good predictions, not just for one input data set, but for the range of input data sets that are of interest.

In this phase of the validation process, the model is viewed as an input–output transformation—that is, the model accepts values of the input parameters and transforms these inputs into output measures of performance. It is this correspondence that is being validated.

Instead of validating the model input–output transformations by predicting the future, the modeler could use historical data that have been reserved for validation purposes only—that is, if one data set has been used to develop and calibrate the model, it is recommended that a separate data set be used as the final validation test. Thus, accurate "prediction of the past" can replace prediction of the future for the purpose of validating the model.

A model is usually developed with primary interest in a specific set of system responses to be measured under some range of input conditions. For example, in a queueing system, the responses may be server utilization and customer delay, and the range of input conditions (or input variables) may include two or three servers at some station and a choice of scheduling rules. In a production system, the response may be throughput (i.e., production per hour), and the input conditions may be a choice of several machines that run at different speeds, with each machine having its own breakdown and maintenance characteristics.

In any case, the modeler should use the main responses of interest as the primary criteria for validating a model. If the model is used later for a purpose different from its original purpose, the model should be revalidated in terms of the new responses of interest and under the possibly new input conditions.

A necessary condition for the validation of input–output transformations is that some version of the system under study exist, so that system data under at least one set of input conditions can be collected to compare to model predictions. If the system is in the planning stages and no system operating data can be

collected, complete input–output validation is not possible. Other types of validation should be conducted, to the extent possible. In some cases, subsystems of the planned system may exist, and a partial input–output validation can be conducted.

Presumably, the model will be used to compare alternative system designs or to investigate system behavior under a range of new input conditions. Assume for now that some version of the system is operating and that the model of the existing system has been validated. What, then, can be said about the validity of the model when different inputs are used?—that is, if model inputs are being changed to represent a new system design, or a new way to operate the system, or even hypothesized future conditions, what can be said about the validity of the model with respect to this new but nonexistent proposed system or to the system under new input conditions?

First, the responses of the two models under similar input conditions will be used as the criteria for comparison of the existing system to the proposed system. Validation increases the modeler's confidence that the model of the existing system is accurate. Second, in many cases, the proposed system is a modification of the existing system, and the modeler hopes that confidence in the model of the existing system can be transferred to the model of the new system. This transfer of confidence usually can be justified if the new model is a relatively minor modification of the old model in terms of changes to the operational model (it may be a major change for the actual system). Changes in the operational model ranging from relatively minor to relatively major include the following:

1. minor changes of single numerical parameters, such as the speed of a machine, the arrival rate of customers (with no change in distributional form of interarrival times), the number of servers in a parallel service center, or the mean time to failure or mean time to repair of a machine;
2. minor changes of the form of a statistical distribution, such as the distribution of a service time or a time to failure of a machine;
3. major changes in the logical structure of a subsystem, such as a change in queue discipline for a waiting-line model or a change in the scheduling rule for a job-shop model;
4. major changes involving a different design for the new system, such as a computerized inventory control system replacing an older noncomputerized system, or an automated storage-and-retrieval system replacing a warehouse system in which workers pick items manually using fork trucks.

If the change to the operational model is minor, such as in items 1 or 2, these changes can be carefully verified and output from the new model accepted with considerable confidence. If a sufficiently similar subsystem exists elsewhere, it might be possible to validate the submodel that represents the subsystem and then to integrate this submodel with other validated submodels to build a complete model. In this way, partial validation of the substantial model changes in items 3 and 4 might be possible. Unfortunately, there is no way to validate the input–output transformations of a model of a nonexisting system completely. In any case, within time and budget constraints, the modeler should use as many validation techniques as possible, including input–output validation of subsystem models if operating data can be collected on such subsystems.

Example 10.2 will illustrate some of the techniques that are possible for input–output validation and will discuss the concepts of an input variable, uncontrollable variable, decision variable, output or response variable, and input–output transformation in more detail.

Example 10.2: The Fifth National Bank of Jaspar

The Fifth National Bank of Jaspar, as shown in Figure 10.4, is planning to expand its drive-in service at the corner of Main Street. Currently, there is one drive-in window serviced by one teller. Only one or two transactions are allowed at the drive-in window, so it was assumed that each service time was a random sample from some underlying population. Service times $\{S_i, i = 1, 2, \ldots, 90\}$ and interarrival times $\{A_i, i = 1, 2, \ldots, 90\}$ were collected for the 90 customers who arrived between 11:00 A.M. and 1:00 P.M. on a Friday. This time slot

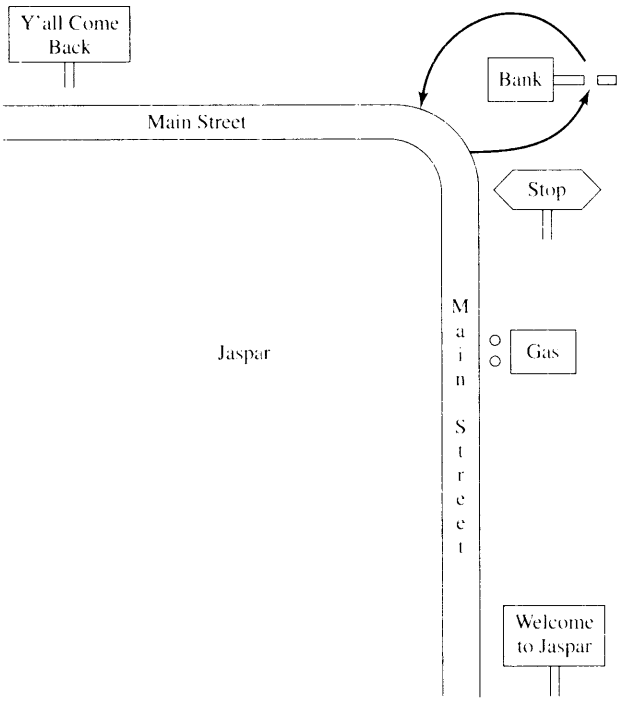


Figure 10.4 Drive-in window at the Fifth National Bank.

was selected for data collection after consultation with management and the teller because it was felt to be representative of a typical rush hour.

Data analysis (as outlined in Chapter 9) led to the conclusion that arrivals could be modeled as a Poisson process at a rate of 45 customers per hour and that service times were approximately normally distributed, with mean 1.1 minutes and standard deviation 0.2 minute. Thus, the model has two input variables:

1. interarrival times, exponentially distributed (i.e., a Poisson arrival process) at rate $\lambda = 45$ per hour;
2. service times, assumed to be $N(1.1, (0.2)^2)$.

Each input variable has a level: the rate ($\lambda = 45$ per hour) for the interarrival times, and the mean 1.1 minutes and standard deviation 0.2 minute for the service times. The interarrival times are examples of uncontrollable variables (i.e., uncontrollable by management in the real system). The service times are also treated as uncontrollable variables, although the level of the service times might be partially controllable. If the mean service time could be decreased to 0.9 minute by installing a computer terminal, the level of the service-time variable becomes a decision variable or controllable parameter. Setting all decision variables at some level constitutes a policy. For example, the current bank policy is one teller ($D_1 = 1$), mean service time $D_2 = 1.1$ minutes, and one line for waiting cars ($D_3 = 1$). (D_1, D_2, ... are used to denote decision variables.) Decision variables are under management's control; the uncontrollable variables, such as arrival rate and actual arrival times, are not under management's control. The arrival rate might change from time to time, but such change is treated as being due to external factors not under management control.

A model of current bank operations was developed and verified in close consultation with bank management and employees. Model assumptions were validated, as discussed in Section 10.3.2. The resulting

model is now viewed as a "black box" that takes all input-variable specifications and transforms them into a set of output or response variables. The output variables consist of all statistics of interest generated by the simulation about the model's behavior. For example, management is interested in the teller's utilization at the drive-in window (percent of time the teller is busy at the window), average delay in minutes of a customer from arrival to beginning of service, and the maximum length of the line during the rush hour. These input and output variables are shown in Figure 10.5 and are listed in Table 10.1, together with some additional output variables. The uncontrollable input variables are denoted by X, the decision variables by D, and the

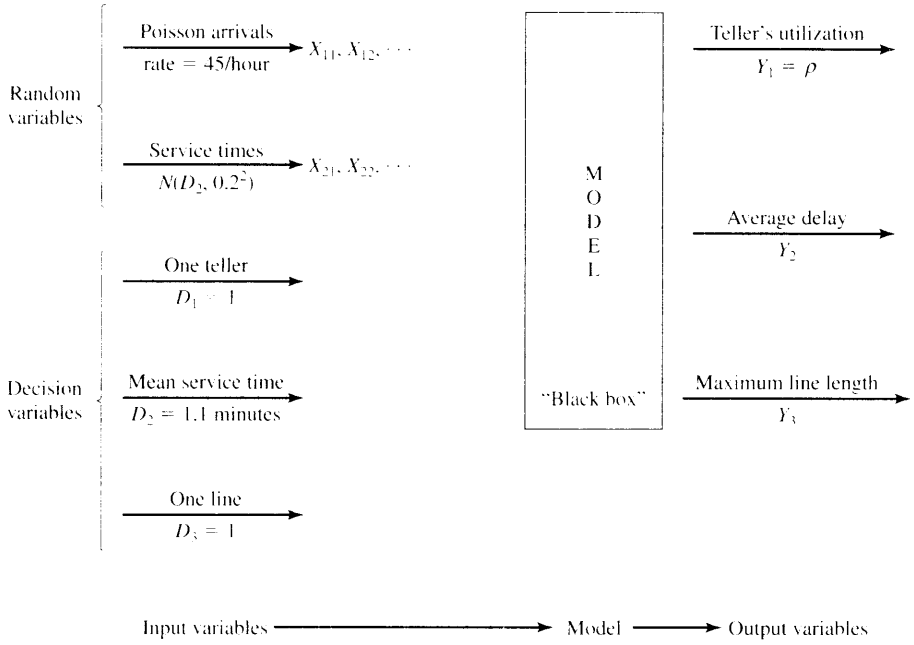


Figure 10.5 Model input–output transformation.

Table 10.1 Input and Output Variables for Model of Current Bank Operations

Input Variables	*Model Output Variables, Y*
D = decision variables	Variables of primary interest
X = other variables	to management (Y_1, Y_2, Y_3)
	Y_1 = teller's utilization
Poisson arrivals at rate = 45/hour	Y_2 = average delay
$X_{11}, X_{12}, \ldots$	Y_3 = maximum line length
Service times, $N(D_2, 0.2^2)$	Other output variables of
$X_{21}, X_{22}, \ldots$	secondary interest
	Y_4 = observed arrival rate
$D_1 = 1$ (one teller)	Y_5 = average service time
$D_2 = 1.1$ minutes (mean service time)	Y_6 = sample standard deviation of service
$D_3 = 1$ (one line)	times
	Y_7 = average length of waiting line

output variables by Y. From the "black box" point of view, the model takes the inputs X and D and produces the outputs Y, namely

$$(X, D) \longrightarrow Y$$

or

$$f(X, D) = Y$$

Here f denotes the transformation that is due to the structure of the model. For the Fifth National Bank study, the exponentially distributed interarrival time generated in the model (by the methods of Chapter 8) between customer $n - 1$ and customer n is denoted by X_{1n}. (Do not confuse X_{1n} with A_n; the latter was an observation made on the real system.) The normally distributed service time generated in the model for customer n is denoted by X_{2n}. The set of decision variables, or policy, is $D = (D_1, D_2, D_3) = (1, 1, 1)$ for current operations. The output, or response, variables are denoted by $Y = (Y_1, Y_2, \ldots, Y_7)$ and are defined in Table 10.1.

For validation of the input–output transformations of the bank model to be possible, real system data must be available, comparable to at least some of the model output Y of Table 10.1. The system responses should have been collected during the same time period (from 11:00 A.M. to 1:00 P.M. on the same Friday) in which the input data $\{A_i, S_i\}$ were collected. This is important because, if system response data were collected on a slower day (say, an arrival rate of 40 per hour), the system responses such as teller utilization (Z_1), average delay (Z_2), and maximum line length (Z_3) would be expected to be lower than the same variables during a time slot when the arrival rate was 45 per hour, as observed. Suppose that the delay of successive customers was measured on the same Friday between 11:00 A.M. and 1:00 P.M. and that the average delay was found to be $Z_2 = 4.3$ minutes. For the purpose of validation, we will consider this to be the true mean value $\mu_0 = 4.3$.

When the model is run with generated random variates X_{1n} and X_{2n}, it is expected that observed values of average delay, Y_2, should be close to $Z_2 = 4.3$ minutes. The generated input values (X_{1n} and X_{2n}) cannot be expected to replicate the actual input values (A_n and S_n) of the real system exactly, but they are expected to replicate the statistical pattern of the actual inputs. Hence, simulation-generated values of Y_2 are expected to be consistent with the observed system variable, $Z_2 = 4.3$ minutes. Now consider how the modeler might test this consistency.

The modeler makes a small number of statistically independent replications of the model. Statistical independence is guaranteed by using nonoverlapping sets of random numbers produced by the random-number generator or by choosing seeds for each replication independently (from a random number table). The results of six independent replications, each of 2 hours duration, are given in Table 10.2.

Table 10.2 Results of Six Replications of the First Bank Model

Replication	Y_4 *(Arrivals/Hour)*	Y_5 *(Minutes)*	$Y_2 =$ *Average Delay (Minutes)*
1	51	1.07	2.79
2	40	1.12	1.12
3	45.5	1.06	2.24
4	50.5	1.10	3.45
5	53	1.09	3.13
6	49	1.07	2.38
Sample mean			2.51
Standard deviation			0.82

Observed arrival rate Y_4 and sample average service time Y_5 for each replication of the model are also noted, to be compared with the specified values of 45/hour and 1.1 minutes, respectively. The validation test consists of comparing the system response, namely average delay $Z_2 = 4.3$ minutes, to the model responses, Y_2. Formally, a statistical test of the null hypothesis

$$H_0 : E(Y_2) = 4.3 \text{ minutes}$$
$$\text{versus} \tag{10.1}$$
$$H_1 : E(Y_2) \neq 4.3 \text{ minutes}$$

is conducted. If H_0 is not rejected, then, on the basis of this test, there is no reason to consider the model invalid. If H_0 is rejected, the current version of the model is rejected, and the modeler is forced to seek ways to improve the model, as illustrated by Figure 10.3. As formulated here, the appropriate statistical test is the t test, which is conducted in the following manner:

Choose a level of significance, α, and a sample size, n. For the bank model, choose

$$\alpha = 0.05, \quad n = 6$$

Compute the sample mean, $\bar{Y}_2$, and the sample standard deviation, S, over the n replications, by using Equations (9.1) and (9.2):

$$\bar{Y}_2 = \frac{1}{n}\sum_{i=1}^{n} Y_{2i} = 2.51 \text{ minutes}$$

and

$$S = \left[\frac{\sum_{i=1}^{n}(Y_{2i} - \bar{Y}_2)^2}{n-1}\right]^{1/2} = 0.82 \text{ minute}$$

where Y_{2i}, $i = 1, \ldots, 6$, are as shown in Table 10.2.

Get the critical value of t from Table A.5. For a two-sided test, such as that in equation (10.1), use $t_{\alpha/2,\, n-1}$; for a one-sided test, use $t_{\alpha,n-1}$ or $-t_{\alpha,n-1}$, as appropriate ($n - 1$ being the degrees of freedom). From Table A.5, $t_{0.025,5} = 2.571$ for a two-sided test.

Compute the test statistic

$$t_0 = \frac{\bar{Y}_2 - \mu_0}{S/\sqrt{n}} \tag{10.2}$$

where μ_0 is the specified value in the null hypothesis, H_0. Here $\mu_0 = 4.3$ minutes, so that

$$t_0 = \frac{2.51 - 4.3}{0.82/\sqrt{6}} = -5.34$$

For the two-sided test, if $|t_0| > t_{\alpha/2,n-1}$, reject H_0. Otherwise, do not reject H_0. [For the one-sided test with $H_1 : E(Y_2) > \mu_0$, reject H_0 if $t > t_{\alpha,n-1}$; with $H_1 : E(Y_2) < \mu_0$, reject H_0 if $t < -t_{\alpha,n-1}$.]

Since $|t| = 5.34 > t_{0.025,5} = 2.571$, reject H_0, and conclude that the model is inadequate in its prediction of average customer delay.

Recall that, in the testing of hypotheses, rejection of the null hypothesis H_0 is a strong conclusion, because

$$P(H_0 \text{ rejected} \mid H_0 \text{ is true}) = \alpha \tag{10.3}$$

and the level of significance α is chosen small, say $\alpha = 0.05$, as was done here. Equation (10.3) says that the probability of making the error of rejecting H_0 when H_0 is in fact true is low ($\alpha = 0.05$)—that is, the probability is small of declaring the model invalid when it is valid (with respect to the variable being tested). The assumptions justifying a t test are that the observations (Y_{2i}) are normally and independently distributed. Are these assumptions met in the present case?

1. The ith observation Y_{2i} is the average delay of all drive-in customers who began service during the ith simulation run of 2 hours; thus, by a Central Limit Theorem effect, it is reasonable to assume that each observation Y_{2i} is approximately normally distributed, provided that the number of customers it is based on is not too small.
2. The observations Y_{2i}, $i = 1, \ldots, 6$, are statistically independent by design—that is, by choice of the random-number seeds independently for each replication or by use of nonoverlapping streams.
3. The t statistic computed by Equation (10.2) is a robust statistic—that is, it is distributed approximately as the t distribution with $n - 1$ degrees of freedom, even when $Y_{21}, Y_{22}, \ldots$ are not exactly normally distributed, and thus the critical values in Table A.5 can reliably be used.

Now that the model of the Fifth National Bank of Jaspar has been found lacking, what should the modeler do? Upon further investigation, the modeler realized that the model contained two unstated assumptions:

1. When a car arrived to find the window immediately available, the teller began service immediately.
2. There is no delay between one service ending and the next beginning, when a car is waiting.

Assumption 2 was found to be approximately correct, because a service time was considered to begin when the teller actually began service but was not considered to have ended until the car had exited the drive-in window and the next car, if any, had begun service, or the teller saw that the line was empty. On the other hand, assumption 1 was found to be incorrect because the teller had other duties—mainly, serving walk-in customers if no cars were present—and tellers always finished with a previous customer before beginning service on a car. It was found that walk-in customers were always present during rush hour; that the transactions were mostly commercial in nature, taking a considerably longer time than the time required to service drive-up customers; and that, when an arriving car found no other cars at the window, it had to wait until the teller finished with the present walk-in customer. To correct this model inadequacy, the structure of the model was changed to include the additional demand on the teller's time, and data were collected on service times of walk-in customers. Analysis of these data found that they were approximately exponentially distributed with a mean of 3 minutes.

The revised model was run, yielding the results in Table 10.3. A test of the null hypothesis $H_0 : E(Y_2) = 4.3$ minutes [as in equation (10.1)] was again conducted, according to the procedure previously outlined.

Choose $\alpha = 0.05$ and $n = 6$ (sample size).

Compute $\bar{Y}_2 = 4.78$ minutes, $S = 1.66$ minutes.

Look up, in Table A.5, the critical value $t_{0.25\,5} = 2.571$.

Compute the test statistic $t_0 = (\bar{Y}_2 - \mu_0)/S\sqrt{n} = 0.710$.

Since $|t_0| < t_{0.025,5} = 2.571$, do not reject H_0, and thus tentatively accept the model as valid.

Failure to reject H_0 must be considered as a weak conclusion unless the power of the test has been estimated and found to be high (close to 1)—that is, it can be concluded only that the data at hand ($Y_{21}, \ldots, Y_{26}$) were not sufficient to reject the hypothesis $H_0 : \mu_0 = 4.3$ minutes. In other words, this test detects no inconsistency between the sample data ($Y_{21}, \ldots, Y_{26}$) and the specified mean μ_0.

The power of a test is the probability of detecting a departure from $H_0 : \mu = \mu_0$ when in fact such a departure exists. In the validation context, the power of the test is the probability of detecting an invalid model.

Table 10.3 Results of Six Replications of the Revised Bank Model

Replication	Y_4 *(Arrivals/Hour)*	Y_5 *(Minutes)*	Y_3 = *Average Delay (Minutes)*
1	51	1.07	5.37
2	40	1.11	1.98
3	45.5	1.06	5.29
4	50.5	1.09	3.82
5	53	1.08	6.74
6	49	1.08	5.49
Sample mean			4.78
Standard deviation			1.66

The power may also be expressed as 1 minus the probability of a Type II, or β, error, where $\beta = P$(Type II error) = P(failing to reject $H_0 | H_1$ is true) is the probability of accepting the model as valid when it is not valid.

To consider failure to reject H_0 as a strong conclusion, the modeler would want β to be small. Now, β depends on the sample size n and on the true difference between $E(Y_3)$ and $\mu_0 = 4.3$ minutes—that is, on

$$\delta = \frac{|E(Y_3) - \mu_0|}{\sigma}$$

where σ, the population standard deviation of an individual Y_{3j}, is estimated by S. Tables A.10 and A.11 are typical operating-characteristic (OC) curves, which are graphs of the probability of a Type II error $\beta(\delta)$ versus δ for given sample size n. Table A.10 is for a two-sided t test; Table A.11 is for a one-sided t test. Suppose that the modeler would like to reject H_0 (model validity) with probability at least 0.90 if the true mean delay of the model, $E(Y_3)$, differed from the average delay in the system, $\mu_0 = 4.3$ minutes, by 1 minute. Then δ is estimated by

$$\hat{\delta} = \frac{|E(Y_3) - \mu_0|}{S} = \frac{1}{1.66} = 0.60$$

For the two-sided test with $\alpha = 0.05$, use of Table A.10 results in

$$\beta(\hat{\delta}) = \beta(0.6) = 0.75 \text{ for } n = 6$$

To guarantee that $\beta(\hat{\delta}) \leq 0.10$, as was desired by the modeler, Table A.10 reveals that a sample size of approximately $n = 30$ independent replications would be required—that is, for a sample size $n = 6$ and assuming that the population standard deviation is 1.66, the probability of accepting H_0 (model validity), when in fact the model is invalid ($|E(Y_3) - \mu_0| = 1$ minute), is $\beta = 0.75$, which is quite high. If a 1-minute difference is critical, and if the modeler wants to control the risk of declaring the model valid when model predictions are as much as 1 minute off, a sample size of $n = 30$ replications is required to achieve a power of 0.9. If this sample size is too high, either a higher β risk (lower power) or a larger difference δ must be considered.

In general, it is always best to control the Type II error, or β error, by specifying a critical difference δ and choosing a sample size by making use of an appropriate OC curve. (Computation of power and use of OC curves for a wide range of tests is discussed in Hines, Montgomery, Goldsman, and Borror [2002].) In summary, in the context of model validation, the Type I error is the rejection of a valid model and is easily

Table 10.4 Types of Error in Model Validation

Statistical Terminology	*Modeling Terminology*	*Associated Risk*
Type I: rejecting H_0 when H_0 is true	Rejecting a valid model	α
Type II: failure to reject H_0 when H_1 is true	Failure to reject an invalid model	β

controlled by specifying a small level of significance α (say $\alpha = 0.1$, 0.05, or 0.01). The Type II error is the acceptance of a model as valid when it is invalid. For a fixed sample size n, increasing α will decrease β, the probability of a Type II error. Once α is set, and the critical difference to be detected is selected, the only way to decrease β is to increase the sample size. A Type II error is the more serious of the two types of errors; thus, it is important to design the simulation experiments to control the risk of accepting an invalid model. The two types of error are summarized in Table 10.4, which compares statistical terminology to modeling terminology.

Note that validation is not to be viewed as an either/or proposition, but rather should be viewed in the context of calibrating a model, as conceptually exhibited in Figure 10.3. If the current version of the bank model produces estimates of average delay (Y_2) that are not close enough to real system behavior ($\mu_0 = 4.3$ minutes), the source of the discrepancy is sought, and the model is revised in light of this new knowledge. This iterative scheme is repeated until model accuracy is judged adequate.

Philosophically, the hypothesis-testing approach tries to evaluate whether the simulation and the real system are *the same* with respect to some output performance measure or measures. A different, but closely related, approach is to attempt to evaluate whether the simulation and the real-system performance measures are *close enough* by using confidence intervals.

We continue to assume that there is a known output performance measure for the existing system, denoted by μ_0, and an unknown performance measure of the simulation, μ, that we hope is close. The hypothesis-testing formulation tested whether $\mu = \mu_0$; the confidence-interval formulation tries to bound the difference $|\mu - \mu_0|$ to see whether it is $\leq \varepsilon$, a difference that is small enough to allow valid decisions to be based on the simulation. The value of ε is set by the analyst.

Specifically, if Y is the simulation output, and $\mu = E(Y)$, then we execute the simulation and form a confidence interval for μ, such as $\bar{Y} \pm t_{\alpha/2,n-1} S/\sqrt{n}$. The determination of whether to accept the model as valid or to refine the model depends on the best-case and worst-case error implied by the confidence interval.

1. Suppose the confidence interval does not contain μ_0. (See Figure 10.6(a).)
   - **(a)** If the best-case error is $> \varepsilon$, then the difference in performance is large enough, even in the best case, to indicate that we need to refine the simulation model.
   - **(b)** If the worst-case error is $\leq \varepsilon$, then we can accept the simulation model as close enough to be considered valid.
   - **(c)** If the best-case error is $\leq \varepsilon$, but the worst-case error is $> \varepsilon$, then additional simulation replications are necessary to shrink the confidence interval until a conclusion can be reached.
2. Suppose the confidence interval does contain μ_0. (See Figure 10.6(b).)
   - **(a)** If either the best-case or worst-case error is $> \varepsilon$, then additional simulation replications are necessary to shrink the confidence interval until a conclusion can be reached.
   - **(b)** If the worst-case error is $\leq \varepsilon$, then we can accept the simulation model as close enough to be considered valid.

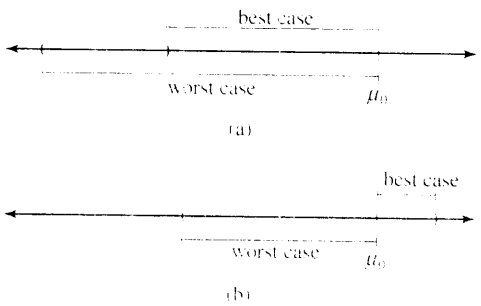


Figure 10.6 Validation of the input–output transformation (a) when the true value falls outside, (b) when the true value falls inside, the confidence interval.

In Example 10.2, $\mu_0 = 4.3$ minutes, and "close enough" was $\varepsilon = 1$ minute of expected customer delay. A 95% confidence interval, based on the 6 replications in Table 10.2, is

$$\bar{Y} \pm t_{n-1,0.975} S/\sqrt{n}$$

$$2.51 \pm 2.571(0.82/\sqrt{6})$$

yielding the interval [1.65, 3.37]. As in Figure 10.6(a), $\mu_0 = 4.3$ falls outside the confidence interval. Since in the best case $|3.37 - 4.3| = 0.93 < 1$, but in the worst case $|1.65 - 4.3| = 2.65 > 1$, additional replications are needed to reach a decision.

10.3.4 Input–Output Validation: Using Historical Input Data

When using artificially generated data as input data, as was done to test the validity of the bank models in Section 10.3.3, the modeler expects the model to produce event patterns that are compatible with, but not identical to, the event patterns that occurred in the real system during the period of data collection. Thus, in the bank model, artificial input data $\{X_{1n}, X_{2n}, n = 1, 2, \ldots\}$ for interarrival and service times were generated, and replicates of the output data Y_2 were compared to what was observed in the real system by means of the hypothesis test stated in equation (10.1). An alternative to generating input data is to use the actual historical record, $\{A_n, S_n, n = 1, 2, \ldots\}$, to drive the simulation model and then to compare model output with system data.

To implement this technique for the bank model, the data $A_1, A_2, \ldots$ and $S_1, S_2, \ldots$ would have to be entered into the model into arrays, or stored in a file to be read as the need arose. Just after customer n arrived at time $t_n = \sum_{i=1}^{n} A_i$, customer $n + 1$ would be scheduled on the future event list to arrive at future time $t_n + A_{n+1}$ (without any random numbers being generated). If customer n were to begin service at time t'_n, a service completion would be scheduled to occur at time $t'_n + S_n$. This event scheduling without random-number generation could be implemented quite easily in a general-purpose programming language or most simulation languages by using arrays to store the data or reading the data from a file.

When using this technique, the modeler hopes that the simulation will duplicate as closely as possible the important events that occurred in the real system. In the model of the Fifth National Bank of Jaspar, the arrival times and service durations will exactly duplicate what happened in the real system on that Friday between 11:00 A.M. and 1:00 P.M. If the model is sufficiently accurate, then the delays of customers, lengths of lines, utilizations of servers, and departure times of customers predicted by the model will be close to what actually happened in the real system. It is, of course, the model-builder's and model-user's judgment that determines the level of accuracy required.

To conduct a validation test using historical input data, it is important that all the input data (A_n, S_n, ...) and all the system response data, such as average delay (Z_2), be collected during the same time period. Otherwise, the comparison of model responses to system responses, such as the comparison of average delay in the model (Y_2) to that in the system (Z_2), could be misleading. The responses (Y_2 and Z_2) depend both on the inputs (A_n and S_n) and on the structure of the system (or model). Implementation of this technique could be difficult for a large system, because of the need for simultaneous data collection of all input variables and those response variables of primary interest. In some systems, electronic counters and devices are used to ease the data-collection task by automatically recording certain types of data. The following example was based on two simulation models reported in Carson *et al.* [1981a, b], in which simultaneous data collection and the subsequent validation were both completed successfully.

Example 10.3: The Candy Factory

The production line at the Sweet Lil' Things Candy Factory in Decatur consists of three machines that make, package, and box their famous candy. One machine (the candy maker) makes and wraps individual pieces of candy and sends them by conveyor to the packer. The second machine (the packer) packs the individual pieces into a box. A third machine (the box maker) forms the boxes and supplies them by conveyor to the packer. The system is illustrated in Figure 10.7.

Each machine is subject to random breakdowns due to jams and other causes. These breakdowns cause the conveyor to begin to empty or fill. The conveyors between the two makers and the packer are used as a temporary storage buffer for in-process inventory. In addition to the randomly occurring breakdowns, if the candy conveyor empties, a packer runtime is interrupted and the packer remains idle until more candy is produced. If the box conveyor empties because of a long random breakdown of the box machine, an operator manually places racks of boxes onto the packing machine. If a conveyor fills, the corresponding maker becomes idle. The purpose of the model is to investigate the frequency of those operator interventions that require manual loading of racks of boxes as a function of various combinations of individual machines and lengths of conveyor. Different machines have different production speeds and breakdown characteristics, and longer conveyors can hold more in-process inventory. The goal is to hold operator interventions to an acceptable level while maximizing production. Machine stoppages (whether due to a full or an empty conveyor) cause damage to the product, so this is also a factor in production.

A simulation model of the Candy Factory was developed, and a validation effort using historical inputs was conducted. Engineers in the Candy Factory set aside a 4-hour time slot from 7:00 A.M. to 11:00 A.M. to

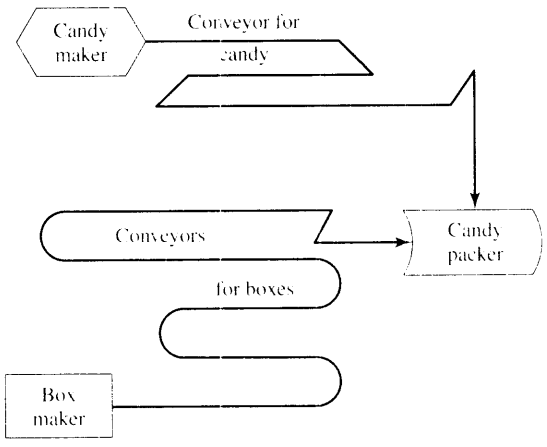


Figure 10.7 Production line at the candy factory.

collect data on an existing production line. For each machine—say, machine i—time to failure and downtime duration

$$T_{i1}, D_{i1}, T_{i2}, D_{i2}, \ldots$$

were collected. For machine $i(i = 1, 2, 3)$, T_{ij} is the jth runtime (or time to failure), and D_{ij} is the successive downtime. A runtime, T_{ij}, can be interrupted by a full or empty conveyor (as appropriate), but resumes when conditions are right. Initial system conditions at 7:00 A.M. were recorded so that they could be duplicated in the model as initial conditions at time 0. Additionally, system responses of primary interest—the production level (Z_1), and the number (Z_2) and time of occurrence (Z_3) of operator interventions—were recorded for comparison with model predictions.

The system input data, T_{ij} and D_{ij}, were fed into the model and used as runtimes and random downtimes. The structure of the model determined the occurrence of shutdowns due to a full or empty conveyor and the occurrence of operator interventions. Model response variables (Y_i, $i = 1, 2, 3$) were collected for comparison to the corresponding system response variables (Z_i, $i = 1, 2, 3$).

The closeness of model predictions to system performance aided the engineering staff considerably in convincing management of the validity of the model. These results are shown in Table 10.5. A simple display such as Table 10.5 can be quite effective in convincing skeptical engineers and managers of a model's validity—perhaps more effectively than the most sophisticated statistical methods!

With only one set of historical input and output data, only one set of simulated output data can be obtained, and thus no simple statistical tests are possible that are based on summary measures; but, if K historical input data sets are collected, and K observations $Z_{i1}, Z_{i2}, \ldots, Z_{iK}$ of some system response variable, Z_i, are collected, such that the output measure Z_{ij} corresponds to the jth input set, an objective statistical test becomes possible. For example, Z_{ij} could be the average delay of all customers who were served during the time the jth input data set was collected. With the K input data sets in hand, the modeler now runs the model K times, once for each input set, and observes the simulated results $W_{i1}, W_{i2}, \ldots, W_{iK}$ corresponding to Z_{ij}, $j = 1, \ldots, K$. Continuing the same example, W_{ij} would be the average delay predicted by the model for the jth input set. The data available for comparison appears as in Table 10.6.

If the K input data sets are fairly homogeneous, it is reasonable to assume that the K observed differences $d_j = Z_{ij} - W_{ij}, j = 1, \ldots, K$, are identically distributed. Furthermore, if the collection of the K sets of input data was separated in time—say, on different days—it is reasonable to assume that the K differences $d_1, \ldots, d_K$ are statistically independent and, hence, that the differences $d_1, \ldots, d_K$ constitute a random sample. In many cases, each Z_i and W_i is a sample average over customers, and so (by the Central Limit Theorem) the differences $d_j = Z_{ij} - W_{ij}$ are approximately normally distributed with some mean μ_d and variance σ_d^2. The appropriate statistical test is then a t test of the null hypothesis of no mean difference:

$$H_0 : \mu_d = 0$$

versus the alternative of significant difference:

$$H_1 : \mu_d \neq 0$$

Table 10.5 Validation of the Candy-Factory Model

Response, i	*System, Z_i*	*Model, Y_i*
1. Production level	897,208	883,150
2. Number of operator interventions	3	3
3. Time of occurrence	7:22, 8:41, 10:10	7:24, 8:42, 10:14

Table 10.6 Comparison of System and Model Output Measures for Identical Historical Inputs

Input Data Set	*System Output,* Z_{ij}	*Model Output,* W_{ij}	*Observed Difference,* d_j	*Squared Deviation from Mean,* $(d_j - \bar{d})^2$
1	Z_{i1}	W_{i1}	$d_1 = Z_{i1} - W_{i1}$	$(d_1 - \bar{d})^2$
2	Z_{i2}	W_{i2}	$d_2 = Z_{i2} - W_{i2}$	$(d_2 - \bar{d})^2$
3	Z_{i3}	W_{i3}	$d_3 = Z_{i3} - W_{i3}$	$(d_3 - \bar{d})^2$
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
K	Z_{iK}	W_{iK}	$d_K = Z_{iK} - W_{iK}$	$(d_K - \bar{d})^2$
			$\bar{d} = \frac{1}{K}\sum_{j=1}^{K} d_j$	$S_d^2 = \frac{1}{K-1}\sum_{j=1}^{K}(d_j - \bar{d})^2$

The proper test is a paired t test (Z_{i1} is paired with W_{i1}, each having been produced by the first input data set, and so on). First, compute the sample mean difference, $\bar{d}$ and the sample variance, S_d^2, by the formulas given in Table 10.6. Then, compute the t statistic as

$$t_0 = \frac{\bar{d} - \mu_d}{S_d/\sqrt{K}} \tag{10.4}$$

(with $\mu_d = 0$), and get the critical value $t_{\alpha/2,\,K-1}$ from Table A.5, where α is the prespecified significance level and $K - 1$ is the number of degrees of freedom. If $|t_0| > t_{\alpha/2,K-1}$, reject the hypothesis H_0 of no mean difference, and conclude that the model is inadequate. If $|t_0| \leq t_{\alpha/2,K-1}$, do not reject H_0, and hence conclude that this test provides no evidence of model inadequacy.

Example 10.4: The Candy Factory, Continued

Engineers at the Sweet Lil' Things Candy Factory decided to expand the initial validation effort reported in Example 10.3. Electronic devices were installed that could automatically monitor one of the production lines, and the validation effort of Example 10.3 was repeated with $K = 5$ sets of input data. The system and the model were compared on the basis of production level. The results are shown in Table 10.7.

Table 10.7 Validation of the Candy-Factory Model (Continued)

Input Data Set, j	*System Production,* Z_{1j}	*Model Production,* W_{1j}	*Observed Difference,* d_j	*Squared Deviation from Mean,* $(d_j - \bar{d})^2$
1	897,208	883,150	14,058	7.594×10^7
2	629,126	630,550	−1,424	4.580×10^7
3	735,229	741,420	−6,191	1.330×10^7
4	797,263	788,230	9,033	1.362×10^7
5	825,430	814,190	11,240	3.4772×10^7
			$\bar{d} = 5,343.2$	$S_d^2 = 7.580 \times 10^7$

A paired t test was conducted to test $H_0 : \mu_d 0$, or equivalently, $H_0 : E(Z_1) = E(W_1)$, where Z_1 is the system production level and W_1 is the production level predicted by the simulated model. Let the level of significance be $\alpha = 0.05$. Using the results in Table 10.7, the test statistic, as given by equation (10.4), is

$$t_0 = \frac{\bar{d}}{S_d/\sqrt{K}} = \frac{5343.2}{8705.85/\sqrt{5}} = 1.37$$

From Table A.5, the critical value is $t_{\alpha/2,K-1} = t_{0.025,4} = 2.78$. Since $|t_0| = 1.37 < t_{0.025,4} = 2.78$, the null hypothesis cannot be rejected on the basis of this test—that is, no inconsistency is detected between system response and model predictions in terms of mean production level. If H_0 had been rejected, the modeler would have searched for the cause of the discrepancy and revised the model, in the spirit of Figure 10.3.

10.3.5 Input - Output Validation: Using a Turing Test

In addition to statistical tests, or when no statistical test is readily applicable, persons knowledgeable about system behavior can be used to compare model output to system output. For example, suppose that five reports of system performance over five different days are prepared, and simulation output data are used to produce five "fake" reports. The 10 reports should all be in exactly the same format and should contain information of the type that managers and engineers have previously seen on the system. The 10 reports are randomly shuffled and given to the engineer, who is asked to decide which reports are fake and which are real. If the engineer identifies a substantial number of the fake reports, the model builder questions the engineer and uses the information gained to improve the model. If the engineer cannot distinguish between fake and real reports with any consistency, the modeler will conclude that this test provides no evidence of model inadequacy. For further discussion and an application to a real simulation, the reader is referred to Schruben [1980]. This type of validation test is commonly called a Turing test. Its use as model development proceeds can be a valuable tool in detecting model inadequacies and, eventually, in increasing model credibility as the model is improved and refined.

10.4 SUMMARY

Validation of simulation models is of great importance. Decisions are made on the basis of simulation results; thus, the accuracy of these results should be subject to question and investigation.

Quite often, simulations appear realistic on the surface because simulation models, unlike analytic models, can incorporate any level of detail about the real system. To avoid being "fooled" by this apparent realism, it is best to compare system data to model data and to make the comparison by using a wide variety of techniques, including an objective statistical test, if at all possible.

As discussed by Van Horn [1969, 1971], some of the possible validation techniques, in order of increasing cost-to-value ratios, include

1. Develop models with high face validity by consulting persons knowledgeable about system behavior on both model structure, model input, and model output. Use any existing knowledge in the form of previous research and studies, observation, and experience.
2. Conduct simple statistical tests of input data for homogeneity, for randomness, and for goodness of fit to assumed distributional forms.
3. Conduct a Turing test. Have knowledgeable people (engineers, managers) compare model output to system output and attempt to detect the difference.

4. Compare model output to system output by means of statistical tests.
5. After model development, collect new system data and repeat techniques 2 to 4.
6. Build the new system (or redesign the old one) conforming to the simulation results, collect data on the new system, and use the data to validate the model (not recommended if this is the only technique used).
7. Do little or no validation. Implement simulation results without validating. (Not recommended.)

It is usually too difficult, too expensive, or too time consuming to use all possible validation techniques for every model that is developed. It is an important part of the model-builder's task to choose those validation techniques most appropriate, both to assure model accuracy and to promote model credibility.

REFERENCES

BALCI, O. [1994], "Validation, Verification and Testing Techniques throughout the Life Cycle of a Simulation Study," *Annals of Operations Research*, Vol. 53, pp. 121–174.

BALCI, O. [1998] "Verification, Validation, and Testing," in *Handbook of Simulation*, J. Banks, ed., John Wiley, New York.

BALCI, O. [2003], "Verification, Validation, and Certification of Modeling and Simulation Applications," in *Proceedings of the* 2003 *Winter Simulation Conference*, ed. by S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, pp. 150–158, Association for Computing Machinery, New York.

BALCI, O., AND R. G. SARGENT [1982a], "Some Examples of Simulation Model Validation Using Hypothesis Testing," in *Proceedings of the Winter Simulation Conference*, ed. by H. J. Highland, Y. W. Chao, and O. S. Madrigal, pp. 620–629, Association for Computing Machinery, New York.

BALCI, O., AND R. G. SARGENT [1982b], "Validation of Multivariate Response Models Using Hotelling's Two-Sample T^2 Test," *Simulation*, Vol. 39, No. 6 (Dec), pp. 185–192.

BALCI, O., AND R. G. SARGENT [1984a], "Validation of Simulation Models via Simultaneous Confidence Intervals," *American Journal of Mathematical Management Sciences*, Vol. 4, Nos. 3 & 4, pp. 375–406.

BALCI, O., AND R. G. SARGENT [1984b], "A Bibliography on the Credibility Assessment and Validation of Simulation and Mathematical Models," *Simuletter*, Vol. 15, No. 3, pp. 15–27.

BORTSCHELLER, B. J., AND E. T. SAULNIER [1992], "Model Reusability in a Graphical Simulation Package," in *Proceedings of the Winter Simulation Conference*, ed. by J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson, pp. 764–772, Association for Computing Machinery, New York.

CARSON, J. S. [2002], "Model Verification and Validation," in *Proceedings of the Winter Simulation Conference*, ed. by E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, pp. 52–58, Association for Computing Machinery, New York.

CARSON, J. S., N. WILSON, D. CARROLL, AND C. H. WYSOWSKI [1981a], "A Discrete Simulation Model of a Cigarette Fabrication Process," *Proceedings of the Twelfth Modeling and Simulation Conference*, University of Pittsburgh, PA.

CARSON, J. S., N. WILSON, D. CARROLL, AND C. H. WYSOWSKI [1981b], "Simulation of a Filter Rod Manufacturing Process," *Proceedings of the* 1981 *Winter Simulation Conference*, ed. by T. I. Oren, C. M. Delfosse, and C. M. Shub, pp. 535–541, Association for Computing Machinery, New York.

CARSON, J. S., [1986], "Convincing Users of Model's Validity is Challenging Aspect of Modeler's Job," *Industrial Engineering*, June, pp. 76–85.

GAFARIAN, A. V., AND J. E. WALSH [1970], "Methods for Statistical Validation of a Simulation Model for Freeway Traffic near an On-Ramp," *Transportation Research*, Vol. 4, p. 379–384.

GASS, S. I. [1983], "Decision-Aiding Models: Validation, Assessment, and Related Issues for Policy Analysis," *Operations Research*, Vol. 31, No. 4, pp. 601–663.

HINES, W. W., D. C. MONTGOMERY, D. M. GOLDSMAN, AND C. M. BORROR [2002], *Probability and Statistics in Engineering*, 4th ed., Wiley, New York.

KLEIJNEN, J. P. C. [1987], *Statistical Tools for Simulation Practitioners*, Marcel Dekker, New York.

KLEIJNEN, J. P. C. [1993], "Simulation and Optimization in Production Planning: A Case Study," *Decision Support Systems*, Vol. 9, pp. 269–280.

KLEIJNEN, J. P. C. [1995], "Theory and Methodology: Verification and Validation of Simulation Models," *European Journal of Operational Research*, Vol. 82, No. 1, pp. 145–162.

LAW, A. M., AND W. D. KELTON [2000], *Simulation Modeling and Analysis*, 3d ed., McGraw-Hill, New York.

NAYLOR, T. H., AND J. M. FINGER [1967], "Verification of Computer Simulation Models," *Management Science*, Vol. 2, pp. B92–B101.

OREN, T. [1981], "Concepts and Criteria to Assess Acceptability of Simulation Studies: A Frame of Reference," *Communications of the Association for Computing Machinery*, Vol. 24, No. 4, pp. 180–89.

SARGENT, R. G. [2003], "Verification and Validation of Simulation Models," in *Proceedings of the 2003 Winter Simulation Conference*, ed. by S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, pp. 37–48, Association for Computing Machinery, New York.

SCHECTER, M., AND R. C. LUCAS [1980], "Validating a Large Scale Simulation Model of Wilderness Recreation Travel," *Interfaces*, Vol. 10, pp. 11–18.

SCHRUBEN, L. W. [1980], "Establishing the Credibility of Simulations," *Simulation*, Vol. 34, pp. 101–105.

SHANNON, R. E. [1975], *Systems Simulation: The Art and Science*, Prentice-Hall, Englewood Cliffs, N.J.

VAN HORN, R. L. [1969], "Validation," in *The Design of Computer Simulation Experiments*, ed. by T. H. Naylor, Duke University Press, Durham, N.C.

VAN HORN, R. L. [1971], "Validation of Simulation Results," *Management Science*, Vol. 17, pp. 247–258.

YOUNGBLOOD, S. M. [1993] "Literature Review and Commentary on the Verification, Validation and Accreditation of Models," in *Proceedings of the Summer Computer Simulation Conference*, Laurel, MD.

EXERCISES

1. A simulation model of a job shop was developed to investigate different scheduling rules. To validate the model, the scheduling rule currently used was incorporated into the model and the resulting output was compared against observed system behavior. By searching the previous year's database records, it was estimated that the average number of jobs in the shop was 22.5 on a given day. Seven independent replications of the model were run, each of 30 days' duration, with the following results for average number of jobs in the shop:

   18.9 22.0 19.4 22.1 19.8 21.9 20.2

   (a) Develop and conduct a statistical test to evaluate whether model output is consistent with system behavior. Use the level of significance $\alpha = 0.05$.

   (b) What is the power of this test if a difference of two jobs is viewed as critical? What sample size is needed to guarantee a power of 0.8 or higher? (Use $\alpha = 0.05$.)

2. System data for the job shop of Exercise 1 revealed that the average time spent by a job in the shop was approximately 4 working days. The model made the following predictions, on seven independent replications, for average time spent in the shop:

   3.70 4.21 4.35 4.13 3.83 4.32 4.05

   (a) Is model output consistent with system behavior? Conduct a statistical test, using the level of significance $\alpha = 0.01$.

   (b) If it is important to detect a difference of 0.5 day, what sample size is needed to have a power of 0.90? Interpret your results in terms of model validity or invalidity. (Use $\alpha = 0.01$.)

3. For the job shop of Exercise 1, four sets of input data were collected over four different 10-day periods, together with the average number of jobs in the shop (Z_i) for each period. The input data were used to

drive the simulation model for four runs of 10 days each, and model predictions of average number of jobs in the shop (Y_i) were collected, with these results:

i	1	2	3	4
Z_i	21.7	19.2	22.8	19.4
Y_i	24.6	21.1	19.7	24.9

(a) Conduct a statistical test to check the consistency of system output and model output. Use the level of significance $\alpha = 0.05$.

(b) If a difference of two jobs is viewed as important to detect, what sample size is required to guarantee a probability of at least 0.80 of detecting this difference if it indeed exists? (Use $\alpha = 0.05$.)

4. Find several examples of actual simulations reported in the literature in which the authors discuss validation of their model. Is enough detail given to judge the adequacy of the validation effort? If so, compare the reported validation with the criteria set forth in this chapter. Did the authors use any validation technique not discussed in this chapter? [Several potential sources of articles on simulation applications include the journal *Interfaces* and *Simulation*, and the *Winter Simulation Conference Proceedings* at www.informs-cs.org.]

5. **(a)** Compare validation in simulation to the validation of theories in the physical sciences.

(b) Compare the issues involved and the techniques available for validation of models of physical systems versus models of social systems.

(c) Contrast the difficulties, and compare the techniques, in validating a model of a manually operated warehouse with fork trucks and other manually operated vehicles, versus a model of a facility with automated guided vehicles, conveyors, and an automated storage-and-retrieval system.

(d) Repeat (c) for a model of a production system involving considerable manual labor and human decision making, versus a model of the same production system after it has been automated.

11

Output Analysis for a Single Model

Output analysis is the examination of data generated by a simulation. Its purpose is either to predict the performance of a system or to compare the performance of two or more alternative system designs. This chapter deals with the analysis of a single system; Chapter 12 deals with the comparison of two or more systems. The need for statistical output analysis is based on the observation that the output data from a simulation exhibits random variability when random-number generators are used to produce the values of the input variables—that is, two different streams or sequences of random numbers will produce two sets of outputs, which (probably) will differ. If the performance of the system is measured by a parameter θ, the result of a set of simulation experiments will be an estimator $\hat{\theta}$ of θ. The precision of the estimator $\hat{\theta}$ can be measured by the standard error of $\hat{\theta}$ or by the width of a confidence interval for θ. The purpose of the statistical analysis is either to estimate this standard error or confidence interval or to figure out the number of observations required to achieve a standard error or confidence interval of a given size—or both.

Consider a typical output variable, Y, the total cost per week of an inventory system; Y should be treated as a random variable with an unknown distribution. A simulation run of length 1 week provides a single sample observation from the population of all possible observations on Y. By increasing the run length, the sample size can be increased to n observations, $Y_1, Y_2, \ldots, Y_n$, based on a run length of n weeks. However, these observations do not constitute a random sample, in the classical sense, because they are not statistically independent. In this case, the inventory on hand at the end of one week is the beginning inventory on hand for the next week, and thus the value of Y_i has some influence on the value of Y_{i+1}. Thus, the sequence of random variables $Y_1, Y_2, \ldots, Y_n$, could be autocorrelated (i.e., correlated with itself). This autocorrelation, which is a measure of a lack of statistical independence, means that classical methods of statistics, which assume independence, are not directly applicable to the analysis of these output data. The methods must be properly modified and the simulation experiments properly designed for valid inferences to be made.

In addition to the autocorrelation present in most simulation output data, the specification of the initial conditions of the system at time 0 can pose a problem for the simulation analyst and could influence the output data. For example, the inventory on hand and the number of backorders at time 0 would most likely influence the value of Y_1, the total cost for week 1. Because of the autocorrelation, these initial conditions would also influence the costs (Y_2, ..., Y_n) for subsequent weeks. The specified initial conditions, if not chosen well, can have an especially deleterious effect on attempting to estimate the steady-state (long-run) performance of a simulation model. For purposes of statistical analysis, the effect of the initial conditions is that the output observations might not be identically distributed and that the initial observations might not be representative of the steady-state behavior of the system.

Section 11.1 distinguishes between two types of simulation—transient versus steady state—and defines commonly used measures of system performance for each type of simulation. Section 11.2 illustrates by example the inherent variability in a stochastic (i.e., probabilistic) discrete-event simulation and thereby demonstrates the need for a statistical analysis of the output. Section 11.3 covers the statistical estimation of performance measures. Section 11.4 discusses the analysis of transient simulations, Section 11.5 the analysis of steady-state simulations.

11.1 TYPES OF SIMULATIONS WITH RESPECT TO OUTPUT ANALYSIS

In the analyzing of simulation output data, a distinction is made between terminating or transient simulations and steady-state simulations. A *terminating* simulation is one that runs for some duration of time T_E, where E is a specified event (or set of events) that stops the simulation. Such a simulated system "opens" at time 0 under well-specified *initial conditions* and "closes" at the *stopping time* T_E. The next four examples are terminating simulations.

Example 11.1

The Shady Grove Bank opens at 8:30 A.M. (time 0) with no customers present and 8 of the 11 tellers working (initial conditions) and closes at 4:30 P.M. (time T_E = 480 minutes). Here, the event E is merely the fact that the bank has been open for 480 minutes. The simulation analyst is interested in modeling the interaction between customers and tellers over the entire day, including the effect of starting up and of closing down at the end of the day.

Example 11.2

Consider the Shady Grove Bank of Example 11.1, but restricted to the period from 11:30 A.M. (time 0) to 1:30 P.M., when it is especially busy. The simulation run length is T_E = 120 minutes. The initial conditions at time 0 (11:30 A.M.) could be specified in essentially two ways: (1) the real system could be observed at 11:30 on a number of different days and a distribution of number of customers in system (at 11:30 A.M.) could be estimated, then these data could be used to load the simulation model with customers at time 0; or (2) the model could be simulated from 8:30 A.M. to 11:30 A.M. without collecting output statistics, and the ending conditions at 11:30 A.M. used as initial conditions for the 11:30 A.M. to 1:30 P.M. simulation.

Example 11.3

A communications system consists of several components plus several backup components. It is represented schematically in Figure 11.1. Consider the system over a period of time, T_E, until the system fails. The stopping event E is defined by E = {A fails, or D fails, or (B and C both fail)}. Initial conditions are that all components are new at time 0.

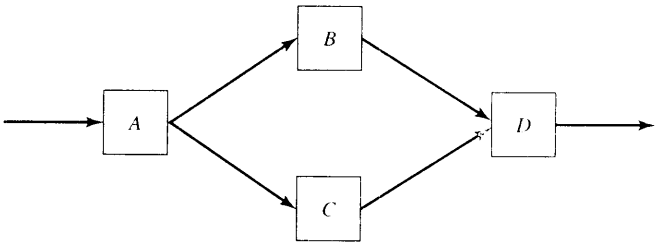


Figure 11.1 Example of a communications system.

Notice that, in the bank model of Example 11.1, the stopping time $T_E = 480$ minutes is known, but in Example 11.3, the stopping time T_E is generally unpredictable in advance; in fact, T_E is probably the output variable of interest, as it represents the total time until the system breaks down. One goal of the simulation might be to estimate $E(T_E)$, the mean time to system failure.

Example 11.4

A widget-manufacturing process runs continuously from Monday mornings until Saturday mornings. The first shift of each workweek is used to load inventory buffers and chemical tanks with the components and catalysts needed to make the final product (28 varieties of widget). These components and catalysts are made continually throughout the week, except for the last shift Friday night, which is used for cleanup and maintenance. Thus, most inventory buffers are near empty at the end of the week. During the first shift on Monday, a buffer stock is built up to cover the eventuality of breakdown in some part of the process. It is desired to simulate this system during the first shift (time 0 to time $T_E = 8$ hours) to study various scheduling policies for loading inventory buffers.

In the simulating of a terminating system, the initial conditions of the system at time 0 must be specified, and the stopping time T_E—or, alternatively, the stopping event E—must be well defined. Although it is certainly true that the Shady Grove Bank in Example 11.1 will open again the next day, the simulation analyst has chosen to consider it a terminating system because the object of interest is one day's operation, including start up and close down. On the other hand, if the simulation analyst were interested in some other aspect of the bank's operations, such as the flow of money or operation of automated teller machines, then the system might be considered as a nonterminating one. Similar comments apply to the communications system of Example 11.3. If the failed component were replaced and the system continued to operate, and, most important, if the simulation analyst were interested in studying its long-run behavior, it might be considered as a nonterminating system. In Example 11.3, however, interest is in its short-run behavior, from time 0 until the first system failure at time T_E. *Therefore, whether a simulation is considered to be terminating depends on both the objectives of the simulation study and the nature of the system.*

Example 11.4 is a terminating system, too. It is also an example of a *transient* (or nonstationary) simulation: the variables of interest are the in-process inventory levels, which are increasing from zero or near zero (at time 0) to full or near full (at time 8 hours).

A *nonterminating system* is a system that runs continuously, or at least over a very long period of time. Examples include assembly lines that shut down infrequently, continuous production systems of many different types, telephone systems and other communications systems such as the Internet, hospital emergency rooms, police dispatching and patrolling operations, fire departments, and continuously operating computer networks.

A simulation of a nonterminating system starts at simulation time 0 under initial conditions defined by the analyst and runs for some analyst-specified period of time T_E. (Significant problems arise concerning the specification of these initial and stopping conditions, problems that we discuss later.) Usually, the analyst wants to study steady-state, or long-run, properties of the system—that is, properties that are not influenced

by the initial conditions of the model at time 0. *A steady-state simulation is a simulation whose objective is to study long-run, or steady-state, behavior of a nonterminating system.* The next two examples are steady-state simulations.

Example 11.5

Consider the widget-manufacturing process of Example 11.4, beginning with the second shift when the complete production process is under way. It is desired to estimate long-run production levels and production efficiencies. For the relatively long period of 13 shifts, this may be considered as a steady-state simulation. To obtain sufficiently precise estimates of production efficiency and other response variables, the analyst could decide to simulate for any length of time, T_E (even longer than 13 shifts)—that is, T_E is not determined by the nature of the problem (as it was in terminating simulations); rather, it is set by the analyst as one parameter in the design of the simulation experiment.

Example 11.6

HAL Inc., a large computer-service bureau, has many customers worldwide. Thus, its large computer system with many servers, workstations, and peripherals runs continuously, 24 hours per day. To handle an increased work load, HAL is considering additional CPUs, memory, and storage devices in various configurations. Although the load on HAL's computers varies throughout the day, management wants the system to be able to accommodate sustained periods of peak load. Furthermore, the time frame in which HAL's business will change in any substantial way is unknown, so there is no fixed planning horizon. Thus, a steady-state simulation at peak-load conditions is appropriate. HAL systems staff develops a simulation model of the existing system with the current peak work load and then explores several possibilities for expanding capacity. HAL is interested in long-run average throughput and utilization of each computer. The stopping time, T_E, is determined not by the nature of the problem, but rather by the simulation analyst, either arbitrarily or with a certain statistical precision in mind.

11.2 STOCHASTIC NATURE OF OUTPUT DATA

Consider one run of a simulation model over a period of time $[0, T_E]$. Since the model is an input-output transformation, as illustrated by Figure 10.5, and since some of the model input variables are random variables, it follows that the model output variables are random variables. Three examples are now given to illustrate the nature of the output data from stochastic simulations and to give a preliminary discussion of several important properties of these data. Do not be concerned if some of these properties and the associated terminology are not entirely clear on a first reading. They will be explained carefully later in the chapter.

Example 11.7: Able and Baker, Revisited

Consider the Able–Baker technical-support call center problem (Example 2.2) which involved customers arriving according to the distribution of Table 2.11 and being served either by Able, whose service-time distribution is given in Table 2.12, or by Baker, whose service-time distribution is given in Table 2.13. The purpose of the simulation is to estimate Able's utilization, ρ, and the mean time spent in the system per customer, w, over the first 2 hours of the workday. Therefore, each run of the model is for a 2-hour period, with the system being empty and idle at time 0. Four statistically independent runs were made by using four distinct streams of random numbers to generate the interarrival and service times. Table 11.1 presents the results. The estimated utilization for run r is given by $\hat{\rho}_r$ and the estimated average system time by $\hat{w}_r$ (i.e., $\hat{w}_r$ is the sample average time in system for all customers served during run r). Notice that, in this sample, the observed utilization ranges from 0.708 to 0.875 and the observed average system time ranges from 3.74 minutes to 4.53 minutes. The stochastic nature of the output data $\{\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3, \hat{\rho}_4\}$ and $\{\hat{w}_1, \hat{w}_2, \hat{w}_3, \hat{w}_4\}$ is demonstrated by the results in Table 11.1.

Table 11.1 Results of Four Independent Runs of 2-Hour Duration of the Able–Baker Queueing Problem

Run, r	*Able's Utilization* $\hat{\rho}_r$	*Average System Time,* $\hat{w}_r$ *(Minutes)*
1	0.808	3.74
2	0.875	4.53
3	0.708	3.84
4	0.842	3.98

There are two general questions that we will address by a statistical analysis—say, of the observed utilizations $\hat{\rho}_r$, $r = 1,\ldots, 4$:

1. estimation of the true utilization $\rho = E(\hat{\rho}_r)$ by a single number, called a point estimate;
2. estimation of the error in our point estimate, in the form either of a standard error or of a confidence interval.

These questions are addressed in Section 11.4 for terminating simulations, such as Example 11.7. Classical methods of statistics may be used because $\hat{\rho}_1$, $\hat{\rho}_2$, $\hat{\rho}_3$, and $\hat{\rho}_4$ constitute a random sample—that is, they are independent and identically distributed. In addition, $\rho = E(\hat{\rho}_r)$ is the parameter being estimated, so each $\hat{\rho}_r$ is an unbiased estimate of the true mean utilization ρ. The analysis of Example 11.7 is considered in Example 11.10 of Section 11.4. A survey of statistical methods applicable to terminating simulations is given by Law [1980]. Additional guidance may be found in Alexopoulos and Seila [1998], Kleijnen [1987], Law and Kelton [2000], and Nelson [2001].

The next example illustrates the effects of correlation and initial conditions on the estimation of long-run mean measures of performance of a system.

Example 11.8

Consider a single-server queue with Poisson arrivals at an average rate of one every 10 minutes ($\lambda = 0.1$ per minute) and service times that are normally distributed, with mean 9.5 minutes and standard deviation 1.75 minutes.[1] This is an *M/G/*1 queue, which was described and analyzed in Section 6.4.1. By Equation (6.11), the true long-run server utilization is $\rho = \lambda E(S) = (0.1)(9.5) = 0.95$. We typically would not need to simulate such a system, because we can analyze it mathematically; but we simulate it here to illustrate difficulties that occur in trying to estimate the long-run mean queue length, L_Q, defined by Equation (6.4).

Suppose we run a single simulation for a total of 5000 minutes and observe the output process $\{L_Q(t), 0 \le t \le 5000\}$, where $L_Q(t)$ is the number of customers in the waiting line at time t minutes. To make this continuous-time process a little easier to analyze, we divide the time interval [0, 5000) into five equal subintervals of 1000 minutes and compute the average number of customers in queue for each interval individually. Specifically, the average number of customers in the queue from time $(j-1)1000$ to $j(1000)$ is

$$Y_j = \frac{1}{1000}\int_{(j-1)1000}^{j(1000)} L_Q(t)\,dt, \quad j = 1,\ldots,5 \tag{11.1}$$

[1] The range of a service time is restricted to ±5 standard deviations, to exclude the possibility of a negative service time; that range covers well over 99.999% of the normal distribution.

Thus, $Y_1 = \int_0^{1000} L_Q(t)\,dt / 1000$ is the time-weighted average number of customers in the queue from time 0 to time 1000, $Y_2 = \int_{1000}^{2000} L_Q(t)\,dt / 1000$ is the same average over [1000, 2000), and so on. Equation (11.1) is a special case of Equation (6.4). The observations $\{Y_1, Y_2, Y_3, Y_4, Y_5\}$ provide an example of "batching" of raw simulation data—in this case, $L_Q\{(t), 0 \le t \le 5000\}$—and the Y_j are called *batch means*. The use of batch means in analyzing output data is discussed in Section 11.5.5. For now, simply notice that batching transforms the continuous-time queue-length process, $\{L_Q(t), 0 \le t \le 5000\}$, into a discrete-time batch-means process $\{Y_i, i = 1, 2, 3, 4, 5\}$ where each Y_i is an estimator of L_Q.

The simulation results of three statistically independent replications are shown in Table 11.2. Each replication, or run, uses a distinct stream of random numbers. For replication 1, Y_{1j} is the batch mean for batch j (the jth interval), as defined by Equation (11.1); similarly, Y_{2j} and Y_{3j} are defined for batch j for replications 2 and 3, respectively. Table 11.2 also gives the sample mean over each replication, $\bar{Y}_r$., for replications $r = 1,2,3$.[2] That is,

$$\bar{Y}_r = \frac{1}{5}\sum_{j=1}^{5} Y_{rj}, \qquad r = 1, 2, 3 \tag{11.2}$$

It probably will not surprise you that, if we take batch averages first, then average the batch means, or just average everything together, we get the same thing. In other words, each $\bar{Y}_r$. is equivalent to the time average over the entire interval [0, 5000) for replication r, as given by Equation (6.4).

Table 11.2 illustrates the inherent variability in stochastic simulations both *within* a single replication and *across* different replications. Consider the variability within replication 3, in which the average queue length over the batching intervals varies from a low of $Y_{31} = 7.67$ customers during the first 1000 minutes to a high of $Y_{33} = 20.36$ customers during the third subinterval of 1000 minutes. Table 11.2 also shows the variability across replications. Compare Y_{15} to Y_{25} to Y_{35}, the average queue lengths over the intervals 4000 to 5000 minutes across all three replications.

Suppose, for the moment, that a simulation analyst makes only one replication of this model and gets the result $\bar{Y}_1 = 3.75$ customers as an estimate of mean queue length, L_Q. How precise is the estimate? This question is usually answered by attempting to estimate the standard error of $\bar{Y}_1$. or by forming a confidence interval. The simulation analyst might think that the five batch means $Y_{11}, Y_{12}, \ldots, Y_{15}$ could be regarded as a random sample; however, the terms in the sequence are not independent, and in fact they are autocorrelated,

Table 11.2 Batched Average Queue Length for Three Independent Replications

Batching Interval (Minutes)	*Batch, j*	*Replication*		
		1, Y_{1j}	*2,* Y_{2j}	*3,* Y_{3j}
[0, 1000)	1	3.61	2.91	7.67
[1000, 2000)	2	3.21	9.00	19.53
[2000, 3000)	3	2.18	16.15	20.36
[3000, 4000)	4	6.92	24.53	8.11
[4000, 5000)	5	2.82	25.19	12.62
[0, 5000)		$\bar{Y}_{1\cdot} = 3.75$	$\bar{Y}_{2\cdot} = 15.56$	$\bar{Y}_{3\cdot} = 13.66$

[2]The dot, as in the subscript $r\cdot$, indicates summation over the second subscript; the bar, as in $\bar{Y}_r$., indicates an average.

because all of the data are obtained from within one replication. If $Y_{11}, \ldots, Y_{15}$ were mistakenly assumed to be independent observations, and their autocorrelation were ignored, the usual classical methods of statistics might severely underestimate the standard error of $\bar{Y}_{1\cdot}$, possibly resulting in the simulation analyst's thinking that a high degree of precision had been achieved. On the other hand, the averages across the three replications, $\bar{Y}_{1\cdot}, \bar{Y}_{2\cdot}$, and $\bar{Y}_{3\cdot}$, can be regarded as independent observations, because they are derived from three different replications.

Intuitively, Y_{11} and Y_{12} are correlated because in replication 1 the queue length at the end of the time interval [0, 1000) is the queue length at the beginning of the interval [1000, 2000)—similarly for any two adjacent batches within a given replication. If the system is congested at the end of one interval, it will be congested for a while at the beginning of the next time interval. Similarly, periods of low congestion tend to follow each other. Within a replication, say for $Y_{r1}, Y_{r2}, \ldots, Y_{r5}$, high values of a batch mean tend to be followed by high values and low values by low. This tendency of adjacent observations to have like values is known as positive autocorrelation. The effect of ignoring autocorrelation when it is present is discussed in more detail in Section 11.5.2.

Now suppose that the purpose of the *M/G/1* queueing simulation of Example 11.8 is to estimate "steady-state" mean queue length, that is, mean queue length under "typical operating conditions over the long run." However, each of the three replications was begun in the empty and idle state (no customers in the queue and the server available). The empty and idle initial state means that, within a given replication, there will be a higher-than- "typical" probability that the system will be uncongested for times close to 0. The practical effect is that an estimator of L_Q—say, $\bar{Y}_{r\cdot}$ for replication r—will be biased low [i.e., $E(\bar{Y}_{r\cdot}) < L_Q$]. The extent of the bias decreases as the run length increases, but, for short-run-length simulations with atypical initial conditions, this initialization bias can produce misleading results. The problem of initialization bias is discussed further in Section 11.5.1.

11.3 MEASURES OF PERFORMANCE AND THEIR ESTIMATION

Consider the estimation of a performance parameter, θ (or ϕ), of a simulated system. It is desired to have a point estimate and an interval estimate of θ (or ϕ). The length of the interval estimate is a measure of the error in the point estimate. The simulation output data are of the form $\{Y_1, Y_2, \ldots, Y_n\}$ for estimating θ; we refer to such output data as *discrete-time data*, because the index n is discrete valued. The simulation output data are of the form $\{Y(t), 0 \le t \le T_E\}$ for estimating ϕ; we refer to such output data as *continuous-time data*, because the index t is continuous valued. For example, Y_i might be the delay of customer i, or the total cost in week i; $Y(t)$ might be the queue length at time t, or the number of backlogged orders at time t. The parameter θ is an ordinary mean; ϕ will be referred to as a time-weighted mean. Whether we call the performance parameter θ or ϕ does not really matter; we use two different symbols here simply to provide a distinction between ordinary means and time-weighted means.

11.3.1 Point Estimation

The point estimator of θ based on the data $\{Y_1, \ldots, Y_n\}$ is defined by

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} Y_i \tag{11.3}$$

where $\hat{\theta}$ is a sample mean based on a sample of size n. Computer simulation languages may refer to this as a "discrete-time," "collect," "tally" or "observational" statistic.

The point estimator $\hat{\theta}$ is said to be unbiased for θ if its expected value is θ—that is, if

$$E(\hat{\theta}) = \theta \tag{11.4}$$

In general, however,

$$E(\hat{\theta}) \neq \theta \tag{11.5}$$

and $E(\hat{\theta}) - \theta$ is called the *bias* in the point estimator θ. It is desirable to have estimators that are unbiased, or, if this is not possible, have a small bias relative to the magnitude of θ. Examples of estimators of the form of Equation (11.3) include $\hat{w}$ and $\hat{w}_Q$ of Equations (6.5) and (6.7), in which case Y_i is the time spent in the (sub)system by customer i.

The point estimator of ϕ based on the data $\{Y(t), 0 \leq t \leq T_E\}$, where T_E is the simulation run length, is defined by

$$\hat{\phi} = \frac{1}{T_E}\int_0^{T_E} Y(t)\,dt \tag{11.6}$$

and is called a time average of $Y(t)$ over $[0, T_E]$. Simulation languages may refer to this as a "continuous-time," "discrete-change" or "time-persistent" statistic. In general,

$$E(\hat{\theta}) \neq \theta \tag{11.7}$$

and $\hat{\phi}$ is said to be biased for ϕ. Again, we would like to obtain unbiased or low-bias estimators. Examples of time averages include L and $\hat{L}_Q$ of Equations (6.3) and (6.4) and Y_j of Equation (11.1).

Generally, θ and ϕ are regarded as mean measures of performance of the system being simulated. Other measures usually can be put into this common framework. For example, consider estimation of the proportion of days on which sales are lost through an out-of-stock situation. In the simulation, let

$$Y_i = \begin{cases} 1, & \text{if out of stock on day } i \\ 0, & \text{otherwise} \end{cases}$$

With n equal to the total number of days, $\hat{\theta}$ defined by Equation (11.3) is a point estimator of θ, the proportion of out-of-stock days. For a second example, consider estimation of the proportion of time queue length is greater than k_0 customers (for example, $k_0 = 10$). If $L_Q(t)$ represents simulated queue length at time t, then (in the simulation) define

$$Y(t) = \begin{cases} 1, & \text{if } L_Q(t) > k_0 \\ 0, & \text{otherwise} \end{cases}$$

Then $\hat{\phi}$, as defined by Equation (11.6), is a point estimator of ϕ, the proportion of time that the queue length is greater than k_0 customers. Thus, estimation of proportions or probabilities is a special case of the estimation of means.

A performance measure that does not fit this common framework is a quantile or percentile. Quantiles describe the level of performance that can be delivered with a given probability, p. For instance, suppose that Y represents the delay in queue that a customer experiences in a service system, measured in minutes. Then the 0.85 quantile of Y is the value θ such that

$$\Pr\{Y \leq \theta\} = p \tag{11.8}$$

where $p = 0.85$ in this case. As a percentage, θ is the $100p$th or 85th percentile of customer delay. Therefore, 85% of all customers will experience a delay of θ minutes or less. Stated differently, a customer has only

a 0.15 probability of experiencing a delay of longer than θ minutes. A widely used performance measure is the median, which is the 0.5 quantile or 50th percentile.

The problem of estimating a quantile is the inverse of the problem of estimating a proportion or probability. Consider Equation (11.8). In estimating a proportion, θ is given and p is to be estimated; but, in estimating a quantile, p is given and θ is to be estimated.

The most intuitive method for estimating a quantile is to form a histogram of the observed values of Y, then find a value $\hat{\theta}$ such that $100p\%$ of the histogram is to the left of (smaller than) $\hat{\theta}$. For instance, if we observe $n = 250$ customer delays $\{Y_1, \ldots, Y_{250}\}$, then an estimate of the 85th percentile of delay is a value $\hat{\theta}$ such that $(0.85)(250) = 212.5 \approx 213$ of the observed values are less than or equal to θ. An obvious estimate is, therefore, to set $\hat{\theta}$ equal to the 213th smallest value in the sample (this requires sorting the data). When the output is a continuous-time process, such as the queue-length process $\{L_Q(t), 0 \le t \le T_E\}$, then a histogram gives the *fraction of time* that the process spent at each possible level (queue length in this example). However, the method for quantile estimation remains the same: Find a value $\hat{\theta}$ such that $100p\%$ of the histogram is to the left of $\hat{\theta}$.

11.3.2 Confidence-Interval Estimation

To understand confidence intervals fully, it is important to understand the difference between a *measure of error* and a *measure of risk*. One way to make the difference clear is to contrast a confidence interval with a *prediction interval* (which is another useful output-analysis tool).

Both confidence and prediction intervals are based on the premise that the data being produced by the simulation is represented well by a probability model. Suppose that model is the normal distribution with mean θ and variance σ^2, both unknown. To make the example concrete, let $\bar{Y}_i$. be the average cycle time for parts produced on the ith replication (representing a day of production) of the simulation. Therefore, θ is the mathematical expectation of $\bar{Y}_i$., and σ is represents the day-to-day variation of the average cycle time.

Suppose our goal is to estimate θ. If we are planning to be in business for a long time, producing parts day after day, then θ is a relevant parameter, because it is the long-run mean daily cycle time. Our average cycle time will vary from day to day, but over the long run the *average of the averages* will be close to θ.

The natural estimator for θ is the overall sample mean of R independent replications, $\bar{Y}.. = \sum_{i=1}^{R} \bar{Y}_i./R$. But $\bar{Y}..$ is not θ, it is an estimate, based on a sample, and it has error. A confidence interval is a measure of that error. Let

$$S^2 = \frac{1}{R-1}\sum_{i=1}^{R}(Y_i. - \bar{Y}..)^2$$

be the sample variance across the R replications. The usual confidence interval, which assumes the Y_i. are normally distributed, is

$$\bar{Y}.. \pm t_{\alpha/2,R-1}\frac{S}{\sqrt{R}}$$

where $t_{\alpha/2,R-1}$ is the quantile of the t distribution with $R - 1$ degrees of freedom that cuts off $\alpha/2$ of the area of each tail. (See Table A.5.) We cannot know for certain exactly how far $\bar{Y}..$ is from θ, but the confidence interval attempts to bound that error. Unfortunately, the confidence interval itself may be wrong. A confidence level, such as 95%, tells us how much we can trust the interval to actually bound the error between $\bar{Y}..$ and θ. The more replications we make, the less error there is in $\bar{Y}..$, and our confidence interval reflects that because $t_{\alpha/2,R-1}S/\sqrt{R}$ will tend to get smaller as R increases, converging to 0 as R goes to infinity.

Now suppose we need to make a promise about what the average cycle time will be on a particular day. A good guess is our estimator $\bar{Y}_{..}$, but it is unlikely to be exactly right. Even θ itself, which is the center of the distribution, is not likely to be the *actual average cycle time* on any particular day, because the daily average cycle time varies. A prediction interval, on the other hand, is designed to be wide enough to contain the actual average cycle time on any particular day with high probability. A prediction interval is a measure of risk; a confidence interval is a measure of error.

The normal-theory prediction interval is

$$\bar{Y}_{..} \pm t_{\alpha/2,R-1} S \sqrt{1+\frac{1}{R}}$$

The length of this interval will not go to 0 as R increases. In fact, in the limit it becomes

$$\theta \pm z_{\alpha/2}\sigma$$

to reflect the fact that, no matter how much we simulate, our daily average cycle time still varies.

In summary, a prediction interval is a measure of risk, and a confidence interval is a measure of error. We can simulate away error by making more and more replications, but we can never simulate away risk, which is an inherent part of the system. We can, however, do a better job of evaluating risk by making more replications.

Example 11.9

Suppose that the overall average of the average cycle time on 120 replications of a manufacturing simulation is 5.80 hours, with a sample standard deviation of 1.60 hours. Since $t_{0.025,119} = 1.98$, a 95% confidence interval for the long-run expected daily average cycle time is $5.80 \pm 1.98(1.60/\sqrt{120})$ or 5.80 ± 0.29 hours. Thus, our best guess of the long-run average of the daily average cycle times is 5.80 hours, but there could be as much as ± 0.29 hours error in this estimate.

On any particular day, we are 95% confident that the average cycle time for all parts produced on that day will be

$$5.80 \pm 1.98(1.60)\sqrt{1+\frac{1}{120}}$$

or 5.80 ± 3.18 hours. The ± 3.18 hours reflects the inherent variability in the daily average cycle times and the fact that we want to be 95% confident of covering the actual average cycle time on a particular day (rather than simply covering the long-run average).

11.4 OUTPUT ANALYSIS FOR TERMINATING SIMULATIONS

Consider a terminating simulation that runs over a simulated time interval $[0, T_E]$ and results in observations $Y_1, \ldots, Y_n$. The sample size, n, may be a fixed number, or it may be a random variable (say, the number of observations that occur during time T_E). A common goal in simulation is to estimate

$$\theta = E\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right)$$

When the output data are of the form $\{Y(t), 0 \le t \le T_E\}$, the goal is to estimate

$$\phi = E\left(\frac{1}{T_E}\int_0^{T_E} Y(t)\,dt\right)$$

The method used in each case is the method of *independent replications*. The simulation is repeated a total of R times, each run using a different random number stream and independently chosen initial conditions (which includes the case that all runs have identical initial conditions). We now address this problem.

11.4.1 Statistical Background

Perhaps the most confusing aspect of simulation output analysis is distinguishing *within-replication* data from *across-replication* data, and understanding the properties and uses of each. The issue can be further confused by the fact that simulation languages often provide only summary measures, like sample means, sample variances, and confidence intervals, rather than all of the raw data. Sometimes these summary measures are *all* the simulation language provides without a lot of extra work.

To illustrate the key ideas, think in terms of the simulation of a manufacturing system and two performance measures of that system, the cycle time for parts (time from release into the factory until completion) and the work in process (WIP, the total number of parts in the factory at any time). In computer applications, these two measures could correspond to the response time and the length of the task queue at the CPU; in a service application, they could be the time to fulfill a customer's request and the number of requests on the "to do" list; in a supply-chain application, they could be the order fill time and the inventory level. Similar measures appear in many systems.

Here is the usual set up for something like cycle time: Let Y_{ij} be the cycle time for the jth part produced in the ith replication. If each replication represents two shifts of production, then the number of parts produced in each replication might differ. Table 11.3 shows, symbolically, the results of R replications.

The across-replication data are formed by summarizing within-replication data: $\bar{Y}_{i\cdot}$ is the sample mean of the n_i cycle times from the ith replication, S_i^2 is the sample variance of the same data, and

$$H_i = t_{\alpha/2,n_i-1}\frac{S}{\sqrt{n_i}} \tag{11.9}$$

is a confidence-interval half-width based on this dataset.

From the across-replication data, we compute overall statistics, the average of the daily cycle time averages

$$\bar{Y}_{\cdot\cdot} = \frac{1}{R}\sum_{i=1}^{R}\bar{Y}_{i\cdot} \tag{11.10}$$

Table 11.3 Within- and Across-Replication Cycle-Time Data

Within-Rep Data				*Across-Rep Data*
Y_{11}	Y_{12}	$\cdots$	Y_{1n_1}	$\bar{Y}_{1\cdot}, S_1^2, H_1$
Y_{21}	Y_{22}	$\cdots$	Y_{2n_2}	$\bar{Y}_{2\cdot}, S_2^2, H_2$
$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
Y_{R1}	Y_{R2}	$\cdots$	Y_{Rn_R}	$\bar{Y}_{R\cdot}, S_R^2, H_R$
				$\bar{Y}_{\cdot\cdot}, S^2, H$

the sample variance of the daily cycle time averages

$$S^2 = \frac{1}{R-1}\sum_{i=1}^{R}(\bar{Y}_i. - \bar{Y}..)^2 \tag{11.11}$$

and finally, the confidence-interval half-width

$$H = t_{\alpha/2,R-1}\frac{S}{\sqrt{R}} \tag{11.12}$$

The quantity $S/\sqrt{R}$ is the standard error, which is sometimes interpreted as the average error in $\bar{Y}..$ as an estimator of θ. Notice that S^2 is *not* the average of the within-replication sample variances, S_i^2; rather, it is the sample variance of the within-replication averages $\bar{Y}_1., \bar{Y}_2., \ldots, \bar{Y}_R..$

Within a replication, work in process (WIP) is a continuous-time output, denoted $Y_i(t)$. The stopping time for the ith replication, T_{E_i}, could be a random variable, in general; in this example, it is the end of the second shift. Table 11.4 is an abstract representation of the data produced.

The within-replication sample mean and variance are defined appropriately for continuous-time data:

$$\bar{Y}_i. = \frac{1}{T_{E_i}}\int_0^{T_{E_i}} Y_i(t)\,dt \tag{11.13}$$

and

$$S_i^2 = \frac{1}{T_{E_i}}\int_0^{T_{E_i}} (Y_i(t) - \bar{Y}_i.)^2\,dt \tag{11.14}$$

A definition for H_i is more problematic, but, to be concrete, take it to be

$$H_i = z_{\alpha/2}\frac{S_i}{\sqrt{T_{E_i}}}. \tag{11.15}$$

Frankly, it is difficult to conceive of a situation in which H_i is relevant, a topic we discuss later. Although the definitions of the within-replication data change for continuous-time data, the across-replication statistics are unchanged, and this is a critical observation.

Table 11.4 Within- and Across-Replication WIP Data

Within-Rep Data	*Across-Rep Data*
$Y_1(t), 0 \le t \le T_{E_1}$	$\bar{Y}_1., S_1^2, H_1$
$Y_2(t), 0 \le t \le T_{E_2}$	$\bar{Y}_2., S_2^2, H_2$
$\vdots$	$\vdots$
$Y_R(t), 0 \le t \le T_{E_R}$	$\bar{Y}_R., S_R^2, H_R$
	$\bar{Y}.., S^2, H$

Here are the key points that must be understood:

- The overall sample average, $\bar{Y}_{..}$, and the individual replication sample averages, $\bar{Y}_{i.}$, are always unbiased estimators of the expected daily average cycle time or daily average WIP.
- Across-replication data are independent (since they are based on different random numbers), are identically distributed (since we are running the same model on each replication), and tend to be normally distributed if they are averages of within-replication data, as they are here. This implies that the confidence interval $\bar{Y}_{..} \pm H$ is often pretty good.
- Within-replication data, on the other hand, might have none of these properties. The individual cycle times may not be identically distributed (if the first few parts of the day find the system empty); they are almost certainly not independent (because one part follows another); and whether they are normally distributed is difficult to know in advance. For this reason, S_i^2 and H_i, which are computed under the assumption of independent and identically distributed (i.i.d.) data, tend not to be useful (although there are exceptions).
- There are situations in which $\bar{Y}_{..}$ and $\bar{Y}_{i.}$ are valid estimators of the expected cycle time for an *individual part* or the *expected WIP at any point in time*, rather than the daily average. (See Section 11.5 on steady-state simulations.) Even when this is the case, the confidence interval $\bar{Y}_{..} \pm H$ is valid, and $\bar{Y}_{i.} \pm H_i$ is not. The difficulty occurs because S_i^2 is a reasonable estimator of the variance of the cycle time, but S_i^2/n_i and S_i^2/T_{E_i} are not good estimators of the Var$[\bar{Y}_{i.}]$—more on this in Section 11.5.2.

Example 11.10: The Able–Baker Problem, Continued

Consider Example 11.7, the Able–Baker technical-support call center problem, with the data for $R = 4$ replications given in Table 11.1. The four utilization estimates, $\hat{\rho}_r$, are time averages of the form of Equation (11.13). The simulation produces output data of the form

$$Y_r(t) = \begin{cases} 1, & \text{if Able is busy at time } t \\ 0, & \text{otherwise} \end{cases}$$

and $\hat{\rho}_r = \bar{Y}_{r.}$ as computed by Equation (11.13) with $T_E = 2$ hours. Similarly, the four average system times, $\hat{w}_1, \ldots, \hat{w}_4$, are analogous to $\bar{Y}_{r.}$ of Table 11.3,: where Y_{ri} is the actual time spent in system by customer i on replication r.

First, suppose that the analyst desires a 95% confidence interval for Able's true utilization, ρ. Using Equation (11.10) compute an overall point estimator

$$\bar{Y}_{..} = \hat{\rho} = \frac{0.808 + 0.875 + 0.708 + 0.842}{4} = 0.808$$

Using Equation (11.11), compute its estimated variance:

$$S^2 = \frac{(0.808 - 0.808)^2 + \cdots + (0.842 - 0.808)^2}{4 - 1} = (0.072)^2$$

Thus, the standard error of $\hat{\rho} = 0.808$ is estimated by s.e. $(\hat{\rho}) = S/\sqrt{4} = 0.036$. Obtain $t_{0.025,3} = 3.18$ from Table A.5, and compute the 95% confidence interval half-width by (11.12) as

$$H = t_{0.025,3} \frac{S}{\sqrt{4}} = (3.18)(0.036) = 0.114$$

giving 0.808 ± 0.114 or, with 95% confidence,

$$0.694 \le \rho \le 0.922$$

In a similar fashion, compute a 95% confidence interval for mean time in system w:

$$\hat{w} = \frac{3.74 + 4.53 + 3.84 + 3.98}{4} = 4.02 \text{ minutes}$$

$$S^2 = \frac{(3.74 - 4.02)^2 + \cdots + (3.98 - 4.02)^2}{3 - 1} = (0.352)^2$$

so that

$$H = t_{0.025,3} \frac{S}{\sqrt{4}} = (3.18)(0.176) = 0.560$$

or

$$4.02 - 0.56 \le w \le 4.02 + 0.56$$

Thus, the 95% confidence interval for w is $3.46 \le w \le 4.58$.

11.4.2 Confidence Intervals with Specified Precision

By Expression (11.12), the half-length H of a $100(1 - \alpha)\%$ confidence interval for a mean θ, based on the t distribution, is given by

$$H = t_{\alpha/2,R-1} \frac{S}{\sqrt{R}}$$

where S^2 is the sample variance and R is the number of replications. Suppose that an error criterion ϵ is specified; in other words, it is desired to estimate θ by $\bar{Y}_{..}$ to within $\pm\epsilon$ with high probability—say, at least $1 - \alpha$. Thus, it is desired that a sufficiently large sample size, R, be taken to satisfy

$$P(|\bar{Y}_{..} - \theta| < \epsilon) \ge 1 - \alpha$$

When the sample size, R, is fixed, no guarantee can be given for the resulting error. But if the sample size can be increased, an error criterion can be specified.

Assume that an initial sample of size R_0 replications has been observed—that is, the simulation analyst initially makes R_0 independent replications. We must have $R_0 \ge 2$, with 10 or more being desirable. The R_0 replications will be used to obtain an initial estimate S_0^2 of the population variance σ^2. To meet the half-length criterion, a sample size R must be chosen such that $R \ge R_0$ and

$$H = t_{\alpha/2,R-1} \frac{S_0}{\sqrt{R}} \le \epsilon \tag{11.16}$$

Solving for R in Inequality (11.23) shows that R is the smallest integer satisfying $R \ge R_0$ and

$$R \ge \left(\frac{t_{\alpha/2,R-1} S_0}{\epsilon} \right)^2 \tag{11.17}$$

Since $t_{\alpha/2,R-1} \geq z_{\alpha/2}$, an initial estimate for R is given by

$$R \geq \left(\frac{z_{\alpha/2} S_0}{\epsilon} \right)^2 \tag{11.18}$$

where $z_{\alpha/2}$ is the $100(1 - \alpha/2)$ percentage point of the standard normal distribution from Table A.3. And since $t_{\alpha/2,R-1} \approx z_{\alpha/2}$ for large R (say, $R \geq 50$), the second inequality for R is adequate when R is large. After determining the final sample size, R, collect $R - R_0$ additional observations (i.e., make $R - R_0$ additional replications, or start over and make R total replications) and form the $100(1-\alpha)\%$ confidence interval for θ by

$$\bar{Y}_{..} - t_{\alpha/2,R-1} \frac{S}{\sqrt{R}} \leq \theta \leq \bar{Y}_{..} + t_{\alpha/2,R-1} \frac{S}{\sqrt{R}} \tag{11.19}$$

where $\bar{Y}_{..}$ and S^2 are computed on the basis of all R replications, $\bar{Y}_{..}$ by Equation (11.10), and S^2 by Equation (11.11). The half-length of the confidence interval given by Inequality (11.19) should be approximately, ϵ or smaller; however, with the additional $R - R_0$ observations, the variance estimator S^2 could differ somewhat from the initial estimate S_0^2, possibly causing the half-length to be greater than desired. If the confidence interval (11.19) is too large, the procedure may be repeated, using Inequality (11.17), to determine an even larger sample size.

Example 11.11

Suppose that it is desired to estimate Able's utilization in Example 11.7 to within ±0.04 with probability 0.95. An initial sample of size $R_0 = 4$ is taken, with the results given in Table 11.1. An initial estimate of the population variance is $S_0^2 = (0.072)^2 = 0.00518$. (See Example 11.10 for the relevant data.) The error criterion is $\epsilon = 0.04$, and the confidence coefficient is $1 - \alpha = 0.95$. From Inequality (11.18), the final sample size must be at least as large as

$$\frac{z_{0.025}^2 S_0^2}{\epsilon^2} = \frac{(1.96)^2 (0.00518)}{(0.04)^2} = 12.44$$

Next, Inequality (11.17) can be used to test possible candidates ($R = 13, 14, \ldots$) for final sample size, as follows:

R	13	14	15
$t_{0.025,R-1}$	2.18	2.16	2.14
$\frac{t_{0.025,R-1}^2 S_0^2}{\epsilon^2}$	15.39	15.10	14.83

Thus, $R = 15$ is the smallest integer satisfying Inequality (11.17), so $R - R_0 = 15 - 4 = 11$ additional replications are needed. After obtaining the additional outputs, we would again need to compute the half-width H to ensure that it is as small as is desired.

11.4.3 Quantiles

To present the interval estimator for quantiles, it is helpful to review the interval estimator for a mean in the special case when the mean represents a proportion or probability, p. In this book, we have chosen to treat a proportion or probability as just a special case of a mean. However, in many statistics texts, probabilities are treated separately.

When the number of independent replications $Y_1, \ldots, Y_R$ is large enough that $t_{\alpha/2,n-1} \doteq z_{\alpha/2}$, the confidence interval for a probability p is often written as

$$\hat{p} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{R-1}}$$

where $\hat{p}$ is the sample proportion (tedious algebra shows that this formula for the half-width is precisely equivalent to Equation (11.12) when used in estimating a proportion).

As mentioned in Section 11.3, the quantile-estimation problem is the inverse of the probability-estimation problem: Find θ such that $\Pr\{Y \le \theta\} = p$. Thus, to estimate the p quantile, we find that value $\hat{\theta}$ such that $100p\%$ of the data in a histogram of Y is to the left of $\hat{\theta}$ (or stated differently, the npth smallest value of $Y_1, \ldots, Y_R$).

Extending this idea, an approximate $(1 - \alpha)100\%$ confidence interval for θ can be obtained by finding two values: θ_l that cuts off $100p_l\%$ of the histogram and θ_u that cuts off $100p_u\%$ of the histogram, where

$$p_l = p - z_{\alpha/2}\sqrt{\frac{p(1-p)}{R-1}}$$
$$p_u = p + z_{\alpha/2}\sqrt{\frac{p(1-p)}{R-1}} \qquad (11.20)$$

(Recall that we know p.) In terms of sorted values, $\hat{\theta}_l$ is the Rp_l smallest value (rounded down) and $\hat{\theta}_u$ is the Rp_u smallest value (rounded up), of $Y_1, \ldots, Y_R$.

Example 11.12

Suppose that we want to estimate the 0.8 quantile of the time to failure (in hours) for the communications system in Example 11.3 and form a 95% confidence interval for it. A histogram of $R = 500$ independent replications is shown in Figure 11.2.

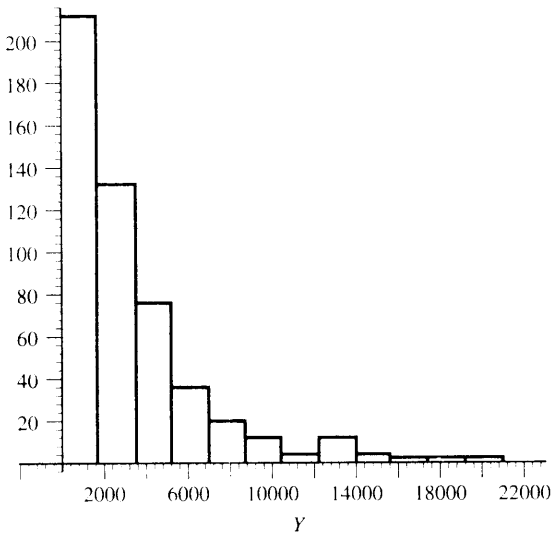


Figure 11.2 Failure data in hours for 500 replications of the communications system.

The point estimator is $\hat{\theta} = 4644$ hours, because 80% of the data in the histogram is to the left of 4644. Equivalently, it is the $500 \times 0.8 = 400$th smallest value of the sorted data.

To obtain the confidence interval we first compute

$$p_l = p - z_{\alpha/2}\sqrt{\frac{p(1-p)}{R-1}} = 0.8 - 1.96\sqrt{\frac{0.8(0.2)}{499}} = 0.765$$

$$p_u = p + z_{\alpha/2}\sqrt{\frac{p(1-p)}{R-1}} = 0.8 + 1.96\sqrt{\frac{0.8(0.2)}{499}} = 0.835$$

The lower bound of the confidence interval is $\hat{\theta}_l = 4173$ (the $500 \times p_l = 382$nd smallest value, rounding down); the upper bound of the confidence interval is $\hat{\theta}_u = 5119$ hours (the $500 \times p_u = 418$th smallest value, rounding up).

11.4.4 Estimating Probabilities and Quantiles from Summary Data

Knowing the equation for the confidence interval half-width is important if all the simulation software provides is $\bar{Y}_{..}$ and H and you need to work out the number of replications required to get a prespecified precision, or if you need to estimate a probability or quantile. You know the number of replications, so the sample standard deviation can be extracted from H by using the formula

$$S = \frac{H\sqrt{R}}{t_{\alpha/2,R-1}}$$

With this information, the method in Section 11.4.2 can be employed.

The more difficult problem is estimating a probability or quantile from summary data. When all we have available is the sample mean and confidence-interval halfwidth (which gives us the sample standard deviation), then one approach is to use a normal-theory approximation for the probabilities or quantiles we desire, specifically

$$\Pr\{\bar{Y}_i \le c\} \approx \Pr\left\{Z \le \frac{c - \bar{Y}_{..}}{S}\right\}$$

and

$$\hat{\theta} \approx \bar{Y}_{..} + z_p S$$

The following example illustrates how this is done.

Example 11.13

From 25 replications of the manufacturing simulation, a 90% confidence interval for the daily average WIP is 218 ± 32. What is the probability that the daily average WIP is less than 350? What is the 85th percentile of daily average WIP?

First, we extract the standard deviation:

$$S = \frac{H\sqrt{R}}{t_{0.05,24}} = \frac{32\sqrt{25}}{1.71} = 93$$

Then, we use the normal approximations and Table A.3 to get

$$\Pr\{\bar{Y}_{.} \le 350\} \approx \Pr\left\{Z \le \frac{350-218}{93}\right\} = \Pr\{Z \le 1.42\} = 0.92$$

and

$$\hat{\theta} \approx \bar{Y}_{.} + z_{0.85}S = 218 + 1.04(93) = 315 \text{ parts}$$

There are shortcomings to obtaining our probabilities and quantiles this way. The approximation depends heavily on whether the output variable of interest is normally distributed. If the output variable itself is not an average, then this approximation is suspect. Therefore, we expect the approximation to work well for statements about the average daily cycle time, for instance, but very poorly for the cycle time of an individual part.

11.5 OUTPUT ANALYSIS FOR STEADY-STATE SIMULATIONS

Consider a single run of a simulation model whose purpose is to estimate a *steady-state*, or *long-run*, characteristic of the system. Suppose that the single run produces observations $Y_1, Y_2, \ldots$, which, generally, are samples of an autocorrelated time series. The steady-state (or long-run) measure of performance, θ, is defined by

$$\theta = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} Y_i \tag{11.21}$$

with probability 1, where the value of θ is independent of the initial conditions. (The phrase "with probability 1" means that essentially all simulations of the model, using different random numbers, will produce series Y_i, $i = 1, 2, \ldots$ whose sample average converges to θ.) For example, if Y_i was the time customer i spent talking to an operator, then θ would be the long-run average time a customer spends talking to an operator; and, because θ is defined as a limit, it is independent of the call center's conditions at time 0. Similarly, the steady-state performance for a continuous-time output measure $\{Y(t), t \ge 0\}$, such as the number of customers in the call center's hold queue, is defined as

$$\phi = \lim_{T_E \to \infty} \frac{1}{T_E} \int_0^{T_E} Y(t)\, dt$$

with probability 1.

Of course, the simulation analyst could decide to stop the simulation after some number of observations—say, n—have been collected; or the simulation analyst could decide to simulate for some length of time T_E that determines n (although n may vary from run to run). The sample size n (or T_E) is a *design* choice; it is not inherently determined by the nature of the problem. The simulation analyst will choose simulation run length (n or T_E) with several considerations in mind:

1. Any bias in the point estimator that is due to artificial or arbitrary initial conditions. (The bias can be severe if run length is too short, but generally it decreases as run length increases.)
2. The desired precision of the point estimator, as measured by the standard error or confidence interval half-width.
3. Budget constraints on computer resources.

The next subsection discusses initialization bias and the following subsections outline two methods of estimating point-estimator variability. For clarity of presentation, we discuss only estimation of θ from a discrete-time output process. Thus, when discussing one replication (or run), the notation

$$Y_1, Y_2, Y_3, \ldots$$

will be used; if several replications have been made, the output data for replication r will be denoted by

$$Y_{r1}, Y_{r2}, Y_{r3}, \ldots \tag{11.22}$$

11.5.1 Initialization Bias in Steady-State Simulations

There are several methods of reducing the point-estimator bias caused by using artificial and unrealistic initial conditions in a steady-state simulation. The first method is to initialize the simulation in a state that is more representative of long-run conditions. This method is sometimes called intelligent initialization. Examples include

1. setting the inventory levels, number of backorders, and number of items on order and their arrival dates in an inventory simulation;
2. placing customers in queue and in service in a queueing simulation;
3. having some components failed or degraded in a reliability simulation.

There are at least two ways to specify the initial conditions intelligently. If the system exists, collect data on it and use these data to specify more nearly typical initial conditions. This method sometimes requires a large data-collection effort. In addition, if the system being modeled does not exist—for example, if it is a variant of an existing system—this method is impossible to implement. Nevertheless, it is recommended that simulation analysts use any available data on existing systems to help initialize the simulation, as this will usually be better than assuming the system to be "completely stocked," "empty and idle," or "brand new" at time 0.

A related idea is to obtain initial conditions from a second model of the system that has been simplified enough to make it mathematically solvable. The queueing models in Chapter 6 are very useful for this purpose. The simplified model can be solved to find long-run expected or most likely conditions—such as the expected number of customers in the queue—and these conditions can be used to initialize the simulation.

A second method to reduce the impact of initial conditions, possibly used in conjunction with the first, is to divide each simulation run into two phases: first, an initialization phase, from time 0 to time T_0, followed by a data-collection phase from time T_0 to the stopping time $T_0 + T_E$—that is, the simulation begins at time 0 under specified initial conditions I_0 and runs for a specified period of time T_0. Data collection on the response variables of interest does not begin until time T_0 and continues until time $T_0 + T_E$. The choice of T_0 is quite important, because the system state at time T_0, denoted by I, should be more nearly representative of steady-state behavior than are the original initial conditions at time 0, I_0. In addition, the length T_E of the data-collection phase should be long enough to guarantee sufficiently precise estimates of steady-state behavior. Notice that the system state, I, at time T_0 is a random variable and to say that the system has reached an approximate steady state is to say that the probability distribution of the system state at time T_0 is sufficiently close to the steady-state probability distribution as to make the bias in point estimates of response variables negligible. Figure 11.3 illustrates the two phases of a steady-state simulation. The effect of starting a simulation run of a queueing model in the empty and idle state, as well as several useful plots to aid the simulation analyst in choosing an appropriate value of T_0, are given in the following example.

Example 11.14

Consider the *M/G/1* queue discussed in Example 11.8. Suppose that a total of 10 independent replications were made ($R = 10$), each replication beginning in the empty and idle state. The total simulation run length on each replication was $T_0 + T_E = 15{,}000$ minutes. The response variable was queue length, $L_Q(t, r)$, at time t,

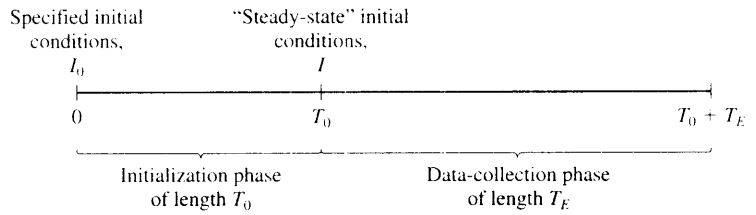


Figure 11.3 Initialization and data collection phases of a steady-state simulation run.

where the second argument, r, denotes the replication ($r = 1,\ldots, 10$). The raw output data were batched, as in Example 11.8, Equation (11.1), in batching intervals of 1000 minutes, to produce the following batch means:

$$Y_{rj} = \frac{1}{1000}\int_{(j-1)1000}^{j(1000)} L_Q(t, r)\,dt \tag{11.23}$$

for replication $r = 1,\ldots, 10$ and for batch $j = 1, 2,\ldots, 15$. The estimator in Equation (11.23) is simply the time-weighted-average queue length over the time interval $[(j-1)1000, j(1000))$, similar to that in Equation (6.4). The 15 batch means for the 10 replications are given in Table 11.5.

Normally we average all the batch means *within* each replication to obtain a replication average. However, our goal at this stage is to identify the trend in the data due to initialization bias and find out when it dissipates. To do this, we will average corresponding batch means *across* replications and plot them (this idea is usually attributed to Welch [1983]). Such averages are known as *ensemble averages*. Specifically, for each batch j, define the ensemble average across all R replications to be

$$\bar{Y}_{\cdot j} = \frac{1}{R}\sum_{r=1}^{R} Y_{rj} \tag{11.24}$$

($R = 10$ here). The ensemble averages $\bar{Y}_{\cdot j}$, $j = 1, \ldots, 15$ are displayed in the third column of Table 11.6. Notice that $\bar{Y}_{\cdot 1} = 4.03$ and $\bar{Y}_{\cdot 2} = 5.45$ are estimates of mean queue length over the time periods [0, 1000) and [1000, 2000), respectively, and they are less than all other ensemble averages $\bar{Y}_{\cdot j}$ ($j = 3,\ldots, 15$). The simulation analyst may suspect that this is due to the downward bias in these estimators, which in turn is due to the queue being empty and idle at time 0. This downward bias is further illustrated in the plots that follow.

Figure 11.4 is a plot of the ensemble averages, $\bar{Y}_{\cdot j}$, versus $1000j$, for $j = 1, 2,\ldots, 15$. The actual values, $\bar{Y}_{\cdot j}$, are the discrete set of points in circles, which have been connected by straight lines as a visual aid. Figure 11.4 illustrates the downward bias of the initial observations. As time becomes larger, the effect of the initial conditions on later observations lessens and the observations appear to vary around a common mean. When the simulation analyst feels that this point has been reached, then the data-collection phase begins.

Table 11.6 also gives the cumulative average sample mean after deleting zero, one, and two batch means from the beginning—that is, using the ensemble average batch means $\bar{Y}_{\cdot j}$, when deleting d observations out of a total of n observations, compute

$$\bar{Y}_{\cdot\cdot}(n, d) = \frac{1}{n-d}\sum_{i=d+1}^{n} \bar{Y}_{\cdot i} \tag{11.25}$$

The results in Table 11.6 for the *M/G/1* simulation are for $d = 0$, 1, and 2, and $n = d + 1,\ldots, 15$. These cumulative averages with deletion, namely $\bar{Y}_{\cdot\cdot}(n, d)$, are plotted for comparison purposes in Figure 11.5. We do not recommend using cumulative averages to determine the initialization phase, for reasons given next.

Table 11.5 Individual Batch Means (Y_{ij}) for $M/G/1$ Simulation with Empty and Idle Initial State

	Batch														
Replication	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
1	3.61	3.21	2.18	6.92	2.82	1.59	3.55	5.60	3.04	2.57	1.41	3.07	4.03	2.70	2.71
2	2.91	9.00	16.15	24.53	25.19	21.63	24.47	8.45	8.53	14.84	23.65	27.58	24.19	8.58	4.06
3	7.67	19.53	20.36	8.11	12.62	22.15	14.10	9.87	23.96	24.50	14.56	6.08	4.83	16.04	23.41
4	6.62	1.75	12.87	8.77	1.25	1.16	1.92	6.29	4.74	17.43	18.24	18.59	4.62	2.76	1.57
5	2.18	1.32	2.14	2.18	2.59	1.20	4.11	6.21	7.31	1.58	2.16	3.08	2.32	2.21	3.32
6	0.93	3.54	4.80	0.72	2.95	5.56	1.96	2.07	2.74	3.45	14.24	13.39	7.87	0.94	3.19
7	1.12	2.59	5.05	1.16	2.72	5.12	5.03	4.14	4.98	15.81	9.29	2.14	8.72	29.80	28.94
8	1.54	5.94	5.33	2.91	2.69	1.91	3.27	3.61	10.35	9.66	4.13	6.14	7.90	2.61	7.95
9	8.93	4.78	0.74	2.56	9.43	18.63	8.14	1.49	4.51	1.69	12.62	11.28	3.32	3.42	3.35
10	4.78	2.84	10.39	5.87	1.01	2.59	16.77	27.25	26.81	20.96	7.26	2.32	5.04	8.50	9.11

Table 11.6 Summary of Data for $M/G/1$ Simulation: Ensemble Batch Means and Cumulative Means, Averaged Over 10 Replications

Run Length T	*Batch* j	*Average Batch Mean,* $\bar{Y}_{\cdot j}$	*Cumulative Average (No Deletion),* $\bar{Y}_{\cdot\cdot}(j, 0)$	*Cumulative Average (Delete 1),* $\bar{Y}_{\cdot\cdot}(j, 1)$	*Cumulative Average (Delete 2),* $\bar{Y}_{\cdot\cdot}(j, 2)$
1,000	1	4.03	4.03	—	—
2,000	2	5.45	4.74	5.45	—
3,000	3	8.00	5.83	6.72	8.00
4,000	4	6.37	5.96	6.61	7.18
5,000	5	6.33	6.04	6.54	6.90
6,000	6	8.15	6.39	6.86	7.21
7,000	7	8.33	6.67	7.11	7.44
8,000	8	7.50	6.77	7.16	7.45
9,000	9	9.70	7.10	7.48	7.77
10,000	10	11.25	7.51	7.90	8.20
11,000	11	10.76	7.81	8.18	8.49
12,000	12	9.37	7.94	8.29	8.58
13,000	13	7.28	7.89	8.21	8.46
14,000	14	7.76	7.88	8.17	8.40
15,000	15	8.76	7.94	8.21	8.43

Figure 11.4 Ensemble averages $\bar{Y}_{\cdot j}$ for $M/G/1$ queue.

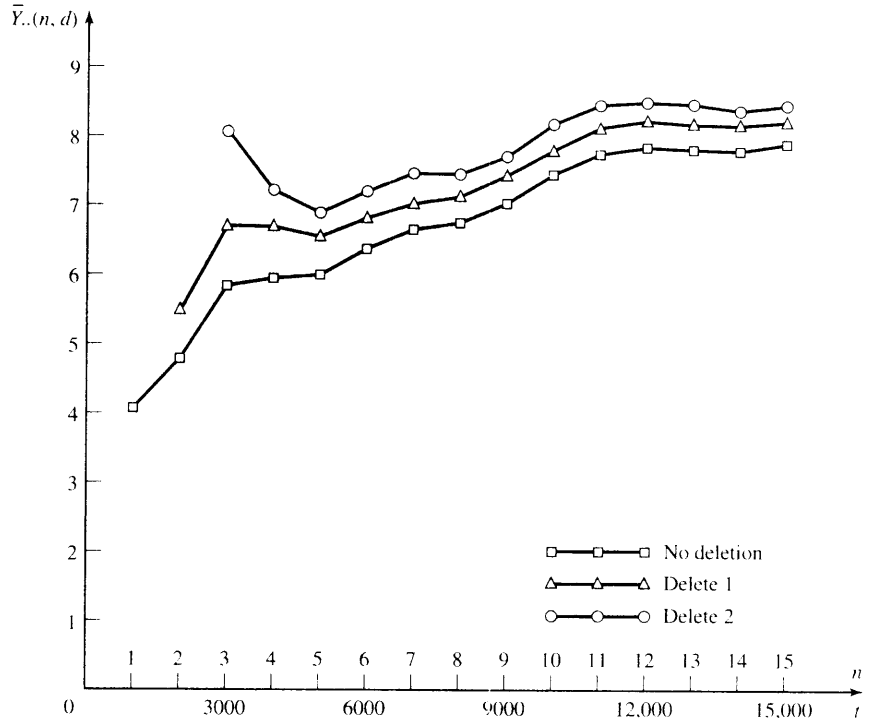


Figure 11.5 Cumulative average queue length $\bar{Y}_{..}(n, d)$ versus time $1000n$.

From Figures 11.4 and 11.5, it is apparent that downward bias is present, and this initialization bias in the point estimator can be reduced by deletion of one or more observations. For the 15 ensemble average batch means, it appears that the first two observations have considerably more bias than any of the remaining ones. The effect of deleting first one and then two batch means is also illustrated in Table 11.6 and Figure 11.5. As expected, the estimators increase in value as more data are deleted; that is, $\bar{Y}_{..}(15, 2) = 8.43$ and $\bar{Y}_{..}(15, 1) = 8.21$ are larger than $\bar{Y}_{..}(15, 0) = 7.94$. It also appears from Figure 11.5 that $\bar{Y}_{..}(n, d)$ is increasing for $n = 5, 6, \ldots, 11$ (and all $d = 0, 1, 2$), and thus there may still be some initialization bias. It seems, however, that deletion of the first two batches removes most of the bias.

Unfortunately, there is no widely accepted, objective, and proven technique to guide how much data to delete to reduce initialization bias to a negligible level. Plots can, at times, be misleading, but they are still recommended. Several points should be kept in mind:

1. Ensemble averages, such as Figure 11.4, will reveal a smoother and more precise trend as the number of replications, R, is increased. Since each ensemble average is the sample mean of i.i.d. observations, a confidence interval based on the t distribution can be placed around each point, as shown in Figure 11.6, and these intervals can be used to judge whether or not the plot is precise enough to judge that bias has diminished. *This is the preferred method to determine a deletion point.*
2. Ensemble averages can be smoothed further by plotting a moving average, rather than the original ensemble averages. In a moving average, each plotted point is actually the average of several adjacent ensemble averages. Specifically, the jth plot point would be

$$\tilde{Y}_{\cdot j} = \frac{1}{2m+1} \sum_{i=j-m}^{j+m} \bar{Y}_{\cdot i}$$

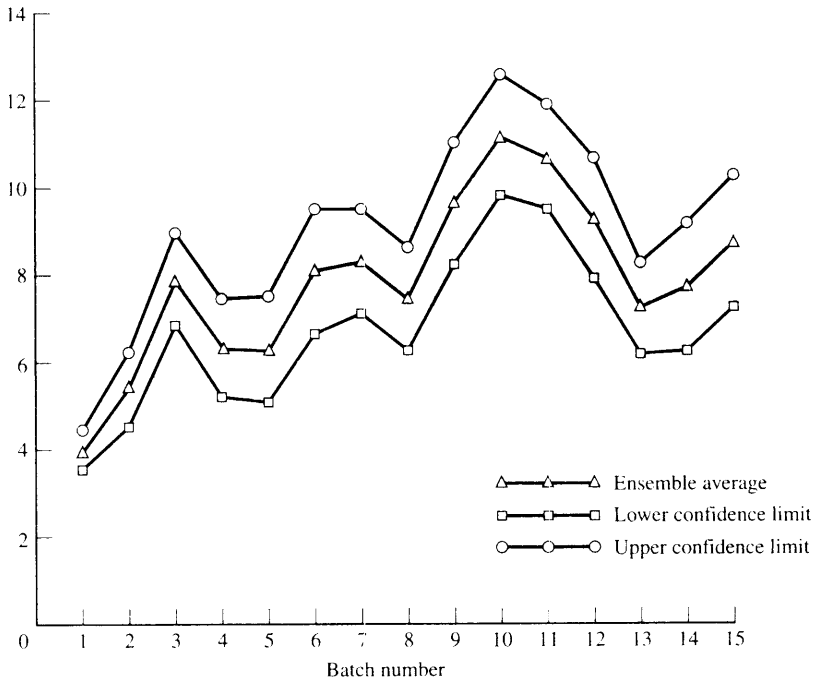


Figure 11.6 Ensemble averages $\bar{Y}_{\cdot j}$ for $M/G/1$ queue with 95% confidence intervals.

for some $m \geq 1$, rather than the original ensemble average $\bar{Y}_{\cdot j}$. The value of m is typically chosen by trial and error until a smooth plot is obtained. See Law and Kelton [2000] or Welch [1983] for further discussion of smoothing.

3. Cumulative averages, such as in Figure 11.5, become less variable as more data are averaged. Therefore, it is expected that the left side of the curve will always be less smooth than the right side. More importantly, cumulative averages tend to converge more slowly to long-run performance than do ensemble averages, because cumulative averages contain all observations, including the most biased ones from the beginning of the run. *For this reason, cumulative averages should be used only if it is not feasible to compute ensemble averages*, such as when only a single replication is possible.
4. Simulation data, especially from queueing models, usually exhibit positive autocorrelation. The more correlation present, the longer it takes for $\bar{Y}_{\cdot j}$ to approach steady state. The positive correlation between successive observations (i.e., batch means) $\bar{Y}_{\cdot 1}, \bar{Y}_{\cdot 2}, \ldots$ can be seen in Figure 11.4.
5. In most simulation studies, the analyst is interested in several different output performance measures at once, such as the number in queue, customer waiting time, and utilization of the servers. Unfortunately, different performance measures could approach steady state at different rates. Thus, it is important to examine each performance measure individually for initialization bias and use a deletion point that is adequate for all of them.

There has been no shortage of solutions to the initialization-bias problem. Unfortunately, for every "solution" that works well in some situations, there are other situations in which either it is not applicable or it performs poorly. Important ideas include testing for bias (e.g., Kelton and Law [1983], Schruben [1980], Goldsman, Schruben, and Swain [1994]); modeling the bias (e.g., Snell and Schruben [1985]); and randomly sampling the initial conditions on multiple replications (e.g., Kelton [1989]).

11.5.2 Error Estimation for Steady-State Simulation

If $\{Y_1, \ldots, Y_n\}$ are not statistically independent, then S^2/n, given by Equation (11.11), is a biased estimator of the true variance, $V(\hat{\theta})$. This is almost always the case when $\{Y_1, \ldots, Y_n\}$ is a sequence of output observations from within a single replication. In this situation, $Y_1, Y_2, \ldots$ is an autocorrelated sequence, sometimes called a time series. Example 11.8 (the *M/G/1* queue) provides an illustration of this situation.

Suppose that our point estimator for θ is the sample mean $\bar{Y} = \sum_{i=1}^{n} Y_i/n$. A general result from mathematical statistics is that the variance of $\bar{Y}$ is[3]

$$V(\bar{Y}) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \text{cov}(Y_i, Y_j) \tag{11.26}$$

where $\text{cov}(Y_i, Y_i) = V(Y_i)$. To construct a confidence interval for θ, an estimate of $V(\bar{Y})$ is required. But obtaining an estimate of (11.26) is pretty much hopeless, because each term $\text{cov}(Y_i, Y_j)$ could be different, in general. Fortunately, systems that have a steady state will, if simulated long enough to pass the transient phase (such as the production-line startup in Example 11.4), produce an output process that is approximately *covariance stationary*. Intuitively, stationarity implies that Y_{i+k} depends on Y_{i+1} in the same manner as Y_k depends on Y_1. In particular, the covariance between two random variables in the time series depends only on the number of observations between them, called the *lag*.

For a covariance-stationary time series, $\{Y_1, Y_2, \ldots\}$, define the lag-k autocovariance by

$$\gamma_k = \text{cov}(Y_1, Y_{1+k}) = \text{cov}(Y_i, Y_{i+k}) \tag{11.27}$$

which, by definition of covariance stationarity, is not a function of i. For $k = 0$, γ_0 becomes the population variance σ^2—that is,

$$\gamma_0 = \text{cov}(Y_i, Y_{i+0}) = V(Y_i) = \sigma^2 \tag{11.28}$$

The lag-k autocorrelation is the correlation between any two observations k apart. It is defined by

$$\rho_k = \frac{\gamma_k}{\sigma^2} \tag{11.29}$$

and has the property that

$$-1 \le \rho_k \le 1, \quad k = 1, 2, \ldots$$

If a time series is covariance stationary, then Equation (11.26) can be simplified substantially. Tedious algebra shows that

$$V(\bar{Y}) = \frac{\sigma^2}{n}\left[1 + 2\sum_{k=1}^{n-1}\left(1 - \frac{k}{n}\right)\rho_k\right] \tag{11.30}$$

where ρ_k is the lag-k autocorrelation given by Equation (11.29).

When $\rho_k > 0$ for all k (or most k), the time series is said to be positively autocorrelated. In this case, large observations tend to be followed by large observations, small observations by small ones. Such a series will tend to drift slowly above and then below its mean. Figure 11.7(a) is an example of a stationary time series exhibiting positive autocorrelation. The output data from most queueing simulations are positively autocorrelated.

On the other hand, if some of the $\rho_k < 0$, the series $Y_1, Y_2, \ldots$ will display the characteristics of negative autocorrelation. In this case, large observations tend to be followed by small observations, and vice versa. Figure 11.7(b) is an example of a stationary time series exhibiting negative autocorrelation. The output of certain inventory simulations might be negatively autocorrelated.

[3]This general result can be derived from the fact that, for two random variables Y_1 and Y_2, $V(Y_1 \pm Y_2) = V(Y_1) + V(Y_2) \pm 2\text{cov}(Y_1, Y_2)$.

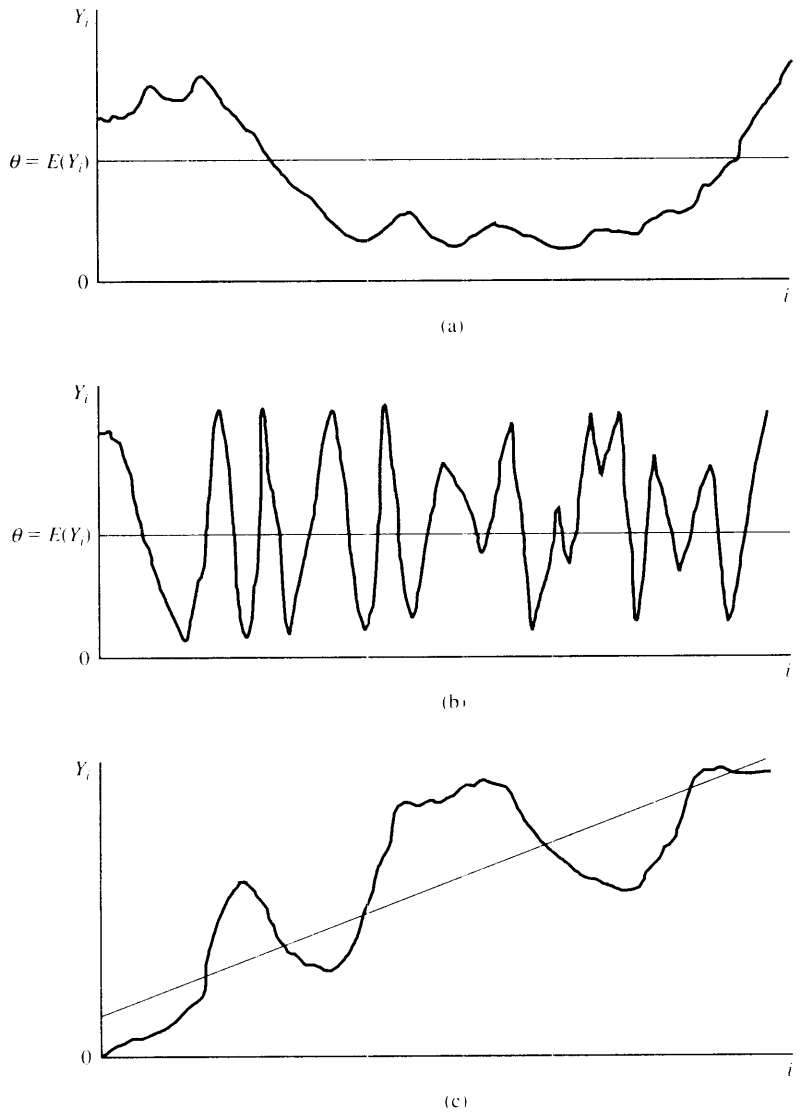


Figure 11.7 (a) Stationary time series Y_i exhibiting positive autocorrelation; (b) stationarytime series Y_i exhibiting negative autocorrelation; (c) nonstationary time series with an upward trend.

Figure 11.7(c) also shows an example of a time series with an upward trend. Such a time series is not stationary; the probability distribution of Y_i is changing with the index i.

Why does autocorrelation make it difficult to estimate $V(\bar{Y})$? Recall that the standard estimator for the variance of a sample mean is S^2/n. By using Equation (11.30), it can be shown [Law, 1977] that the expected value of the variance estimator S^2/n is

$$E\left(\frac{S^2}{n}\right) = BV(\bar{Y}) \tag{11.31}$$

where

$$B = \frac{n/c - 1}{n - 1} \tag{11.32}$$

and c is the quantity in brackets in Equation (11.30). The effect of the autocorrelation on the oxestimator S^2/n is derived by an examination of Equations (11.30) and (11.32). There are essentially three possibilities:

Case 1

If the Y_i are independent, then $\rho_k = 0$ for $k = 1, 2, 3, \ldots$. Therefore, $c = 1 + 2\sum_{k=1}^{n-1}(1 - k/n)\rho_k = 1$ and Equation (11.30) reduces to the familiar σ^2/n. Notice also that $B = 1$, so S^2/n is an unbiased estimator of $V(\bar{Y})$. The Y_i will always be independent when they are obtained from different replications; that independence is the primary reason that we prefer experiment designs calling for multiple replications.

Case 2

If the autocorrelations ρ_k are primarily positive, then $c = 1 + 2\sum_{k=1}^{n-1}(1 - k/n)\rho_k > 1$, so that $n/c < n$, and hence $B < 1$. Therefore, S^2/n is biased low as an estimator of $V(\bar{Y})$. If this correlation were ignored, the nominal $100(1 - \alpha)\%$ confidence interval given by Expression (11.12) would be too short, and its true confidence coefficient would be less than $1 - \alpha$. The practical effect would be that the simulation analyst would have unjustified confidence (in the apparent precision of the point estimator) due to the shortness of the confidence interval. If the correlations ρ_k are large, B could be quite small, implying a significant underestimation.

Case 3

If the autocorrelations ρ_k are substantially negative, then $0 \le c < 1$, and it follows that $B > 1$ and S^2/n is biased high for $V(\bar{Y})$. In other words, the true precision of the point estimator $\bar{Y}$ would be greater than what is indicated by its variance estimator S^2/n, because

$$V(\bar{Y}) < E\left(\frac{S^2}{n}\right)$$

As a result, the nominal $100(1 - \alpha)\%$ confidence interval of Expression (11.12) would have true confidence coefficient greater than $1 - \alpha$. This error is less serious than Case 2, because we are unlikely to make incorrect decisions if our estimate is actually more precise than we think it is.

A simple example demonstrates why we are especially concerned about positive correlation: Suppose you want to know how students on a university campus will vote in an upcoming election. To estimate their preferences, you plan to solicit 100 responses. The standard experiment is to randomly select 100 students to poll; call this experiment A. An alternative is to randomly select 20 students and ask each of them to state their preference 5 times in the same day; call this experiment B. Both experiments obtain 100 responses, but clearly an estimate based on experiment B will be less precise (will have larger variance) than an estimate based on experiment A. Experiment A obtains 100 independent responses, whereas experiment B obtains only 20 independent responses and 80 dependent ones. The five opinions from any one student are perfectly positively correlated (assuming a student names the same candidate all five times). Although this is an extreme example, it illustrates that estimates based on positively correlated data are more variable than estimates based on independent data. Therefore, a confidence interval or other measure of error should account correctly for dependent data, but S^2/n does not.

Two methods for eliminating or reducing the deleterious effects of autocorrelation upon estimation of a mean are given in the following sections. Unfortunately, some simulation languages either use or facilitate the use of S^2/n as an estimator of $V(\bar{Y})$, the variance of the sample mean, in all situations. If used uncritically in a simulation with positively autocorrelated output data, the downward bias in S^2/n and the resulting

shortness of a confidence interval for θ will convey the impression of much greater precision than actually exists. When such positive autocorrelation is present in the output data, the true variance of the point estimator, $\bar{Y}$, can be many times greater than is indicated by S^2/n.

11.5.3 Replication Method for Steady-State Simulations

If initialization bias in the point estimator has been reduced to a negligible level (through some combination of intelligent initialization and deletion), then the method of independent replications can be used to estimate point-estimator variability and to construct a confidence interval. The basic idea is simple: Make R replications, initializing and deleting from each one the same way.

If, however, significant bias remains in the point estimator and a large number of replications are used to reduce point-estimator variability, the resulting confidence interval can be misleading. This happens because *bias is not affected by the number of replications* (R); it is affected only by deleting more data (i.e., increasing T_0) or extending the length of each run (i.e., increasing T_E). Thus, increasing the number of replications (R) could produce shorter confidence intervals around the "wrong point." Therefore, it is important to do a thorough job of investigating the initial-condition bias.

If the simulation analyst decides to delete d observations of the total of n observations in a replication, then the point estimator of θ is $\bar{Y}_{\cdot\cdot}(n, d)$, defined by Equation (11.25)—that is, the point estimator is the average of the remaining data. The basic raw output data, $\{Y_{rj},\ r = 1, \ldots, R;\ j = 1, \ldots, n\}$, are exhibited in Table 11.7. Each Y_{rj} is derived in one of the following ways:

Case 1

Y_{rj} is an individual observation from within replication r; for example, Y_{rj} could be the delay of customer j in a queue, or the response time to job j in a job shop.

Case 2

Y_{rj} is a batch mean from within replication r of some number of discrete-time observations. (Batch means are discussed further in Section 11.5.5.)

Case 3

Y_{rj} is a batch mean of a continuous-time process over time interval j; for instance, as in Example 11.14, Equation (11.23) defines Y_{rj} as the time-average (batch mean) number in queue over the interval $[1000(j-1), 1000j)$.

In Case 1, the number d of deleted observations and the total number of observations n might vary from one replication to the next, in which case replace d by d_r and n by n_r. For simplicity, assume that d and n are constant over replications. In Cases 2 and 3, d and n will be constant.

Table 11.7 Raw Output Data from a Steady-State Simulation

	Observations						*Replication*
Replication	*1*	⋯	*d*	*d + 1*	⋯	*n*	*Averages*
1	$Y_{1,1}$	⋯	$Y_{1,d}$	$Y_{1,d+1}$	⋯	$Y_{1,n}$	$\bar{Y}_1(n, d)$
2	$Y_{2,1}$	⋯	$Y_{2,d}$	$Y_{2,d+1}$	⋯	$Y_{2,n}$	$\bar{Y}_2(n, d)$
⋮	⋮		⋮	⋮		⋮	⋮
R	$Y_{R,1}$	⋯	$Y_{R,d}$	$Y_{R,d+1}$	⋯	$Y_{R,n}$	$\bar{Y}_R(n, d)$
	$Y_{\cdot 1}$	⋯	$\bar{Y}_{\cdot d}$	$\bar{Y}_{\cdot,d+1}$	⋯	$\bar{Y}_{\cdot n}$	$\bar{Y}_{\cdot\cdot}(n, d)$

When using the replication method, each replication is regarded as a single sample for the purpose of estimating θ. For replication r, define

$$\bar{Y}_r(n, d) = \frac{1}{n-d}\sum_{j=d+1}^{n} Y_{rj} \tag{11.33}$$

as the sample mean of all (nondeleted) observations in replication r. Because all replications use different random-number streams and all are initialized at time 0 by the same set of initial conditions (I_0), the replication averages

$$\bar{Y}_1(n, d), \ldots, \bar{Y}_R(n, d)$$

are independent and identically distributed random variables—that is, they constitute a random sample from some underlying population having unknown mean

$$\theta_{n,d} = \mathrm{E}[\bar{Y}_r(n, d)] \tag{11.34}$$

The overall point estimator, given in Equation (11.25), is also given by

$$\bar{Y}_{\cdot\cdot}(n, d) = \frac{1}{R}\sum_{r=1}^{R} \bar{Y}_r(n, d) \tag{11.35}$$

as can be seen from Table 11.7 or from using Equation (11.24). Thus, it follows that

$$E[\bar{Y}_{\cdot\cdot}(n, d)] = \theta_{n,d}$$

also. If d and n are chosen sufficiently large, then $\theta_{n,d} \approx \theta$, and $\bar{Y}_{\cdot\cdot}(n, d)$ is an approximately unbiased estimator of θ. The bias in $\bar{Y}_{\cdot\cdot}(n, d)$ is $\theta_{n,d} - \theta$.

For convenience, when the value of n and d are understood, abbreviate $\bar{Y}_r(n,d)$ (the mean of the undeleted observations from the rth replication) and $\bar{Y}_{\cdot\cdot}(n, d)$ (the mean of $\bar{Y}_1(n, d), \ldots, \bar{Y}_R(n, d)$ by $\bar{Y}_r$. and $\bar{Y}_{\cdot\cdot}$, respectively. To estimate the standard error of $\bar{Y}_{\cdot\cdot}$, first compute the sample variance,

$$S^2 = \frac{1}{R-1}\sum_{r=1}^{R}(\bar{Y}_r - \bar{Y}_{\cdot\cdot})^2 = \frac{1}{R-1}\left(\sum_{r=1}^{R}\bar{Y}_r^2 - R\bar{Y}_{\cdot\cdot}^2\right) \tag{11.36}$$

The standard error of $\bar{Y}_{\cdot\cdot}$ is given by

$$\text{s.e.}(\bar{Y}_{\cdot\cdot}) = \frac{S}{\sqrt{R}} \tag{11.37}$$

A $100(1 - \alpha)\%$ confidence interval for θ, based on the t distribution, is given by

$$\bar{Y}_{\cdot\cdot} - t_{\alpha/2,R-1}\frac{S}{\sqrt{R}} \le \theta \le \bar{Y}_{\cdot\cdot} + t_{\alpha/2,R-1}\frac{S}{\sqrt{R}} \tag{11.38}$$

where $t_{\alpha/2,R-1}$ is the $100(1-\alpha/2)$ percentage point of a t distribution with $R-1$ degrees of freedom. This confidence interval is valid only if the bias of $\bar{Y}_{..}$ is approximately zero.

As a rough rule, the length of each replication, beyond the deletion point, should be at least ten times the amount of data deleted. In other words, $(n-d)$ should at least $10d$ (or more generally, T_E should be at least $10T_0$). Given this run length, the number of replications should be as many as time permits, up to about 25 replications. Kelton [1986] established that there is little value in dividing the available time into more than 25 replications, so, if time permits making more than 25 replications of length $T_0 + 10T_0$, then make 25 replications of longer than $T_0 + 10T_0$, instead. Again, these are rough rules that need not be followed slavishly.

Example 11.15

Consider again the $M/G/1$ queueing simulation of Examples 11.8 and 11.14. Suppose that the simulation analyst decides to make $R = 10$ replications, each of length $T_E = 15{,}000$ minutes, each starting at time 0 in the empty and idle state, and each initialized for $T_0 = 2000$ minutes before data collection begins. The raw output data consist of the batch means defined by Equation (11.23); recall that each batch mean is simply the average number of customers in queue for a 1000-minute interval. The first two batch means are deleted ($d = 2$). The purpose of the simulation is to estimate, by a 95% confidence interval, the long-run time-average queue length, denoted by L_Q.

The replication averages $\bar{Y}_r(15,2), r = 1,2,\ldots,10$, are shown in Table 11.8 in the rightmost column. The point estimator is computed by Equation (11.35) as

$$\bar{Y}_{..}(15,2) = 8.43$$

Its standard error is given by Equation (11.37) as

$$\text{s.e.}(\bar{Y}_{..}(15,2)) = 1.59$$

Table 11.8 Data Summary for M/G/1 Simulation by Replication

Replication, r	*Sample Mean for Replication r*		
	(No Deletion) $\bar{Y}_r(15,0)$	*(Delete 1)* $\bar{Y}_r(15,1)$	*(Delete 2)* $\bar{Y}_r(15,2)$
1	3.27	3.24	3.25
2	16.25	17.20	17.83
3	15.19	15.72	15.43
4	7.24	7.28	7.71
5	2.93	2.98	3.11
6	4.56	4.82	4.91
7	8.44	8.96	9.45
8	5.06	5.32	5.27
9	6.33	6.14	6.24
10	10.10	10.48	11.07
$\bar{Y}_{..} = (15,d)$	7.94	8.21	8.43
$\sum_{r=1}^{R} \bar{Y}_r^2$	826.20	894.68	938.34
S^2	21.75	24.52	25.30
S	4.66	4.95	5.03
$S/\sqrt{10} = \text{s.e.}(\bar{Y}_{..})$	1.47	1.57	1.59

and using $\alpha = 0.05$ and $t_{0.025,9} = 2.26$, the 95% confidence interval for long-run mean queue length is given by Inequality (11.38) as

$$8.43 - 2.26(1.59) \le L_Q \le 8.43 + 2.26(1.59)$$

or

$$4.84 \le L_Q \le 12.02$$

The simulation analyst may conclude with a high degree of confidence that the long-run mean queue length is between 4.84 and 12.02 customers. The confidence interval computed here as given by Inequality (11.38) should be used with caution, because a key assumption behind its validity is that enough data have been deleted to remove any significant bias due to initial conditions—that is, that d and n are sufficiently large that the bias $\theta_{n,d} - \theta$ is negligible.

Example 11.16

Suppose that, in Example 11.15, the simulation analyst had decided to delete one batch ($d = 1$) or no batches ($d = 0$). The quantities needed to compute 95% confidence intervals are shown in Table 11.8. The resulting 95% confidence intervals are computed by Inequality (11.38) as follows:

$$(d = 1) \quad 4.66 = 8.21 - 2.26(1.57) \le L_Q \le 8.21 + 2.26(1.57) = 11.76$$
$$(d = 0) \quad 4.62 = 7.94 - 2.26(1.47) \le L_Q \le 7.94 + 2.26(1.47) = 11.26$$

Notice that, for a fixed total sample size, n, two things happen as fewer data are deleted:

1. The confidence interval shifts downward, reflecting the greater downward bias in $\bar{Y}..(n, d)$ as d decreases.
2. The standard error of $\bar{Y}..(n, d)$, namely $S/\sqrt{R}$, decreases as d decreases.

In this example, $\bar{Y}..(n, d)$ is based on a run length of $T_E = 1000(n - d) = 15{,}000 - 1000d$ minutes. Thus, as d decreases, T_E increases, and, in effect, the sample mean $\bar{Y}..$ is based on a larger "sample size" (i.e., longer run length). In general, the larger the sample size, the smaller the standard error of the point estimator. This larger sample size can be due to a longer run length (T_E) per replication, or to more replications (R).

Therefore, there is a trade-off between reducing bias and increasing the variance of a point estimator, when the total sample size (R and $T_0 + T_E$) is fixed. The more deletion (i.e., the larger T_0 is and the smaller T_E is, keeping $T_0 + T_E$ fixed), the less bias but greater variance there is in the point estimator.

Recall that each batch in Examples 11.15 and 11.16 consists of 1000 minutes of simulated time. Therefore, discarding $d = 2$ batches really means discarding 2000 minutes of data, a substantial amount. It is not uncommon for very large deletions to be necessary to overcome the initial conditions.

11.5.4 Sample Size in Steady-State Simulations

Suppose it is desired to estimate a long-run performance measure, θ, within $\pm\epsilon$, with confidence $100(1 - \alpha)\%$. In a steady-state simulation, a specified precision may be achieved either by increasing the number of replications (R) or by increasing the run length (T_E). The first solution, controlling R, is carried out as given in Section 11.4.2 for terminating simulations.

Example 11.17

Consider the data in Table 11.8 for the *M/G/1* queueing simulation as an initial sample of size $R_0 = 10$. Assuming that $d = 2$ observations were deleted, the initial estimate of variance is $S_0^2 = 25.30$. Suppose that it is desired to estimate long-run mean queue length, L_Q, within $\epsilon = 2$ customers with 90% confidence. The final sample size needed must satisfy Inequality (11.17). Using $\alpha = 0.10$ in Inequality (11.18) yields an initial estimate:

$$R \geq \left(\frac{z_{0.05} S_0}{\epsilon} \right)^2 = \frac{1.645^2 (25.30)}{2^2} = 17.1$$

Thus, at least 18 replications will be needed. Proceeding as in Example 11.11, next try $R = 18$, $R = 19$, ... as follows:

R	18	19
$t_{0.05,R-1}$	1.74	1.73
$\left(\frac{t_{0.05,R-1} S_0}{\epsilon} \right)^2$	19.15	18.93

$R = 19 \geq (t_{0.05,18} S_0 / \epsilon)^2 = 18.93$ is the smallest integer R satisfying Inequality (11.17), so a total sample size of $R = 19$ replications is needed to estimate L_Q to within ±2 customers. Therefore, $R - R_0 = 19 - 10 = 9$ additional replications are needed to achieve the specified error.

An alternative to increasing R is to increase total run length $T_0 + T_E$ within each replication. If the calculations in Section 11.4.2, as illustrated in Example 11.17, indicate that $R - R_0$ additional replications are needed beyond the initial number, R_0, then an alternative is to increase run length $(T_0 + T_E)$ in the same proportion (R/R_0) to a new run length $(R/R_0)(T_0 + T_E)$. Thus, additional data will be deleted, from time 0 to time $(R/R_0)T_0$, and more data will be used to compute the point estimates, as illustrated by Figure 11.8. However, the total amount of simulation effort is the same as if we had simply increased the number of replications but maintained the same run length. The advantage of increasing total run length per replication and deleting a fixed proportion $[T_0/(T_0 + T_E)]$ of the total run length is that any residual bias in the point estimator should be further reduced by the additional deletion of data at the beginning of the run. A possible

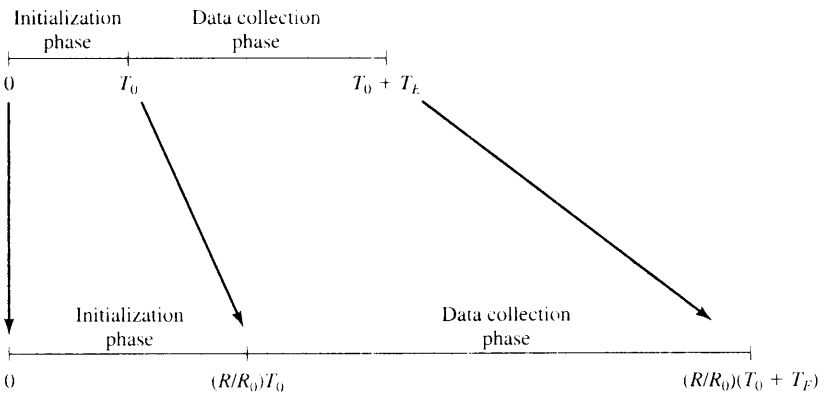


Figure 11.8 Increasing runlength to achieve specified accuracy.

disadvantage of the method is that, in order to continue the simulation of all R replications [from time $T_0 + T_E$ to time $(R/R_0)(T_0 + T_E)$], it is necessary to have saved the state of the model at time $T_0 + T_E$ and to be able to restart the model and run it for the additional required time. Otherwise, the simulations would have to be rerun from time 0, which could be time consuming for a complex model. Some simulation languages have the capability to save enough information that a replication can be continued from time T_E onward, rather than having to start over from time 0.

Example 11.18

In Example 11.17, suppose that run length was to be increased to achieve the desired error, ±2 customers. Since $R/R_0 = 19/10 = 1.9$, the run length should be almost doubled to $(R/R_0)(T_0 + T_E) = 1.9(15{,}000) = 28{,}500$ minutes. The data collected from time 0 to time $(R/R_0)T_0 = 1.9(2000) = 3800$ minutes would be deleted, and the data from time 3800 to time 28,500 used to compute new point estimates and confidence intervals.

11.5.5 Batch Means for Interval Estimation in Steady-State Simulations

One disadvantage of the replication method is that data must be deleted on each replication and, in one sense, deleted data are wasted data, or at least lost information. This suggests that there might be merit in using an experiment design that is based on a single, long replication. The disadvantage of a single-replication design arises when we try to compute the standard error of the sample mean. Since we only have data from within one replication, the data are dependent, and the usual estimator is biased.

The method of *batch means* attempts to solve this problem by dividing the output data from one replication (after appropriate deletion) into a few large batches and then treating the means of these batches as if they were independent. When the raw output data after deletion form a continuous-time process, $\{Y(t), T_0 \le t \le T_0 + T_E\}$, such as the length of a queue or the level of inventory, then we form k batches of size $m = T_E/k$ and compute the batch means as

$$\bar{Y}_j = \frac{1}{m}\int_{T_0+(j-1)m}^{T_0+jm} Y(t+T_0)\,dt$$

for $j = 1, 2, \ldots, k$. In other words, the jth batch mean is just the time-weighted average of the process over the time interval $[T_0 + (j-1)m, T_0 + jm)$, exactly as in Example 11.8.

When the raw output data after deletion form a discrete-time process, $\{Y_i, i = d+1, d+2, \ldots, n\}$, such as the customer delays in a queue or the cost per period of an inventory system, then we form k batches of size $m = (n-d)/k$ and compute the batch means as

$$\bar{Y}_j = \frac{1}{m}\sum_{i=(j-1)m+1}^{jm} Y_{i+d}$$

for $j = 1, 2, \ldots, k$ (assuming k divides $n - d$ evenly, otherwise round down to the nearest integer). That is, the batch means are formed as shown here:

$$\underbrace{Y_1, \ldots, Y_d}_{\text{deleted}}, \underbrace{Y_{d+1}, \ldots, Y_{d+m}}_{\bar{Y}_1}, \underbrace{Y_{d+m+1}, \ldots, Y_{d+2m}}_{\bar{Y}_2}, \ldots, \underbrace{Y_{d+(k-1)m+1}, \ldots, Y_{d+km}}_{\bar{Y}_k}$$

Starting with either continuous-time or discrete-time data, the variance of the sample mean is estimated by

$$\frac{S^2}{k} = \frac{1}{k}\sum_{j=1}^{k}\frac{(\bar{Y}_j - \bar{Y})^2}{k-1} = \frac{\sum_{j=1}^{k}\bar{Y}_j^2 - k\bar{Y}^2}{k(k-1)} \tag{11.39}$$

where $\bar{Y}$ is the overall sample mean of the data after deletion. As was discussed in Section 11.2, the batch means $\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_k$ are not independent; however, if the batch size is sufficiently large, successive batch means will be approximately independent, and the variance estimator will be approximately unbiased.

Unfortunately, there is no widely accepted and relatively simple method for choosing an acceptable batch size m (or equivalently choosing a number of batches k). But there are some general guidelines that can be culled from the research literature:

- Schmeiser [1982] found that for a *fixed total sample size* there is little benefit from dividing it into more than $k = 30$ batches, even if we could do so and still retain independence between the batch means. Therefore, there is no reason to consider numbers of batches much greater than 30, no matter how much raw data are available. He also found that the performance of the confidence interval, in terms of its width and the variability of its width, is poor for fewer than 10 batches. Therefore, a number of batches between 10 and 30 should be used in most applications.
- Although there is typically autocorrelation between batch means at all lags, the lag-1 autocorrelation $\rho_1 = \text{corr}(\bar{Y}_j, \bar{Y}_{j+1})$ is usually studied to assess the dependence between batch means. When the lag-1 autocorrelation is nearly 0, then the batch means are treated as independent. This approach is based on the observation that the autocorrelation in many stochastic processes decreases as the lag increases. Therefore, all lag autocorrelations should be smaller (in absolute value) than the lag-1 autocorrelation.
- The lag-1 autocorrelation between batch means can be estimated as described shortly. However, the autocorrelation should not be estimated from a small number of batch means (such as the $10 \le k \le 30$ recommended above); there is bias in the autocorrelation estimator. Law and Carson [1979] suggest estimating the lag-1 autocorrelation from a large number of batch means based on a smaller batch size (perhaps $100 \le k \le 400$). When the autocorrelation between these batch means is approximately 0, then the autocorrelation will be even smaller if we rebatch the data to between 10 and 30 batch means based on a larger batch size. Hypothesis tests for 0 autocorrelation are available, as described next.
- If the *total sample size is to be chosen sequentially*, say to attain a specified precision, then it is helpful to allow the batch size and number of batches to grow as the run length increases. It can be shown that a good strategy is to allow the number of batches to increase as the square root of the sample size after first finding a batch size at which the lag-1 autocorrelation is approximately 0. Although we will not discuss this point further, an algorithm based on it can be found in Fishman and Yarberry [1997]; see also Steiger and Wilson [2002].

Given these insights, we recommend the following general strategy:

1. Obtain output data from a single replication and delete as appropriate. Recall our guideline: collecting at least 10 times as much data as is deleted.
2. Form up to $k = 400$ batches (but at least 100 batches) with the retained data, and compute the batch means. Estimate the sample lag-1 autocorrelation of the batch means as

$$\hat{\rho}_1 = \frac{\sum_{j=1}^{k-1} (\bar{Y}_j - \bar{Y})(\bar{Y}_{j+1} - \bar{Y})}{\sum_{j=1}^{k} (\bar{Y}_j - \bar{Y})^2}$$

3. Check the correlation to see whether it is sufficiently small.
   (a) If $\hat{\rho}_1 \le 0.2a$, then rebatch the data into $30 \le k \le 40$ batches, and form a confidence interval using $k - 1$ degrees of freedom for the t distribution and Equation (11.39) to estimate the variance of $\bar{Y}$.
   (b) If $\hat{\rho}_1 > 0.2$, then extend the replication by 50% to 100% and go to Step 2. If it is not possible to extend the replication, then rebatch the data into approximately $k = 10$ batches, and form the confidence interval, using $k-1$ degrees of freedom for the t distribution and Equation (11.39) to estimate the variance of $\bar{Y}$.

4. As an additional check on the confidence interval, examine the batch means (at the larger or smaller batch size) for independence, using the following test. (See, for instance, Alexopoulos and Seila [1998].) Compute the test statistic

$$C = \sqrt{\frac{k^2 - 1}{k - 2}}\left(\hat{\rho}_1 + \frac{(\bar{Y}_1 - \bar{Y})^2 + (\bar{Y}_k - \bar{Y})^2}{2\sum_{j=1}^{k} (\bar{Y}_j - \bar{Y})^2} \right)$$

If $C < z_\beta$ then accept the independence of the batch means, where β is the Type I error level of the test (such as 0.1, 0.05, 0.01). Otherwise, extend the replication by 50% to 100% and go to Step 2. If it is not possible to extend the replication, then rebatch the data into approximately $k = 10$ batches, and form the confidence interval, using k-1 degrees of freedom for the t distribution and Equation (11.39) to estimate the variance of $\bar{Y}$.

This procedure, including the final check, is conservative in several respects. First, if the lag-1 autocorrelation is substantially negative then we proceed to form the confidence interval anyway. A dominant negative correlation tends to make the confidence interval wider than necessary, which is an error, but not one that will cause us to make incorrect decisions. The requirement that $\hat{\rho}_1 < 0.2$ at $100 \le k \le 400$ batches is pretty stringent and will tend to force us to get more data (and therefore create larger batches) if there is any hint of positive dependence. And finally, the hypothesis test at the end has a probability of β of forcing us to get more data when none are really needed. But this conservatism is by design; the cost of an incorrect decision is typically much greater than the cost of some additional computer run time.

The batch-means approach to confidence-interval estimation is illustrated in the next example.

Example 11.19

Reconsider the *M/G/1* simulation of Example 11.8, except that the mean service time is changed from 9.5 minutes to 7 minutes (implying a long-run server utilization of 0.7). Suppose that we want to estimate the steady-state expected delay in queue, w_Q, by a 95% confidence interval. To illustrate the method of batch means, assume that one run of the model has been made, simulating 3000 customers after the deletion point. We then form batch means from $k = 100$ batches of size $m = 30$ and estimate the lag-1 autocorrelation to be $\hat{\rho}_1 = 0.346 > 0.2$. Thus, we decide to extend the simulation to 6000 customers after the deletion point, and again we estimate the lag-1 autocorrelation. This estimate, based on $k = 100$ batches of size $m = 60$, is $\hat{\rho}_1 = 0.004 < 0.2$.

Having passed the correlation check, we rebatch the data into $k = 30$ batches of size $m = 200$. The point estimate is the overall mean

$$\bar{Y} = \frac{1}{6000}\sum_{j=1}^{6000} \bar{Y}_j = 9.04$$

minutes. The variance of $\bar{Y}$, computed from the 30 batch means, is

$$\frac{S^2}{k} = \frac{\sum_{j=1}^{30} \bar{Y}_j^2 - 30\bar{Y}^2}{30(29)} = 0.604$$

Thus, a 95% confidence interval is given by

$$\bar{Y} - t_{0.025,29}\sqrt{0.604} \le w_Q \le \bar{Y} + t_{0.025,29}\sqrt{0.604}$$

or

$$7.45 = 9.04 - 2.04(0.777) \le w_Q \le 9.04 + 2.04(0.777) = 10.63$$

Thus, we assert with 95% confidence that true mean delay in queue, w_Q, is between 7.45 and 10.63 minutes. If these results are not sufficiently precise for practical use, the run length should be increased to achieve greater precision.

As a further check on the validity of the confidence interval, we can apply the correlation hypothesis test. To do so, we compute the test statistic from the $k = 30$ batches of size $m = 200$ used to form the confidence interval. This gives

$$C = -0.31 < 1.96 = z_{0.05}$$

confirming the lack of correlation at the 0.05 significance level. Notice that, at this small number of batches, the estimated lag-1 autocorrelation appears to be slightly negative, illustrating our point about the difficulty of estimating correlation with small numbers of observations.

11.5.6 Quantiles

Constructing confidence intervals for quantile estimates in a steady-state simulation can be tricky, especially if the output process of interest is a continuous-time process, such as $L_Q(t)$, the number of customers in queue at time t. In this section, we outline the main issues.

Taking the easier case first, suppose that the output process from a single replication (after appropriate deletion of initial data) is $Y_{d+1}, \ldots, Y_n$. To be concrete, Y_i might be the delay in queue of the ith customer. Then the point estimate of the pth quantile can be obtained as before, either from the histogram of the data or from the sorted values. Of course, only the data after the deletion point are used. Suppose we make R replications and let $\hat{\theta}_i$ be the quantile estimate from the ith. Then the R quantile estimates, $\hat{\theta}_1, \ldots, \hat{\theta}_R$, are independent and identically distributed. Their average is

$$\bar{\theta}_\cdot = \frac{1}{R}\sum_{i=1}^{R} \hat{\theta}_i$$

It can be used as the point estimator of θ; and an approximate confidence interval is

$$\hat{\theta} \pm t_{\alpha/2, R-1} \frac{S}{\sqrt{R}}$$

where S^2 is the usual sample variance of $\hat{\theta}_1, \ldots, \hat{\theta}_R$.

What if only a single replication is obtained? Then the same reasoning applies if we let $\hat{\theta}_i$ be the quantile estimate from *within* the ith batch of data. This requires sorting the data, or forming a histogram, within each batch. If the batches are large enough, then these within-batch quantile estimates will also be approximately i.i.d.

When we have a continuous-time output process, then, in principle, the same methods apply. However, we must be careful not to transform the data in a way that changes the problem. In particular, we cannot first form batch means—as we have done throughout this chapter—and then estimate the quantile from these batch means. The p quantile of the *batch means* of $L_Q(t)$ is *not* the same as the p quantile of $L_Q(t)$ itself. Thus, the quantile point estimate must be formed from the histogram of the raw data—either from each run, if we make replications, or within each batch, if we make a single replication.

11.6 SUMMARY

This chapter emphasized the idea that a stochastic discrete-event simulation is a statistical experiment. Therefore, before sound conclusions can be drawn on the basis of the simulation-generated output data,

a proper statistical analysis is required. The purpose of the simulation experiment is to obtain estimates of the performance measures of the system under study. The purpose of the statistical analysis is to acquire some assurance that these estimates are sufficiently precise for the proposed use of the model.

A distinction was made between terminating simulations and steady-state simulations. Steady-state simulation output data are more difficult to analyze, because the simulation analyst must address the problem of initial conditions and the choice of run length. Some suggestions were given regarding these problems, but unfortunately no simple, complete, and satisfactory solution exists. Nevertheless, simulation analysts should be aware of the potential problems, and of the possible solutions—namely, deletion of data and increasing of the run length. More advanced statistical techniques (not discussed in this text) are given in Alexopoulos and Seila [1998], Bratley, Fox, and Schrage [1996], and Law and Kelton [2000].

The statistical precision of point estimators can be measured by a standard-error estimate or by a confidence interval. The method of independent replications was emphasized. With this method, the simulation analyst generates statistically independent observations, and thus standard statistical methods can be employed. For steady-state simulations, the method of batch means was also discussed.

The main point is that simulation output data contain some amount of random variability; without some assessment of its size, the point estimates cannot be used with any degree of reliability.

REFERENCES

ALEXOPOULOS, C., AND A. F. SEILA [1998], "Output Data Analysis," Chapter 7 in *Handbook of Simulation*, J. Banks, ed., Wiley, New York.

BRATLEY, P., B. L. FOX, AND L. E. SCHRAGE [1996], *A Guide to Simulation*, 2d ed., Springer-Verlag, New York.

FISHMAN, G. S., AND L. S. YARBERRY [1997], "An Implementation of the Batch Means Method," *INFORMS Journal on Computing*, Vol. 9, pp. 296–310.

GOLDSMAN, D., L. SCHRUBEN, AND J. J. SWAIN [1994], "Tests for Transient Means in Simulated Time Series," *Naval Research Logistics*, Vol. 41, pp. 171–187.

KELTON, W. D. [1986], "Replication Splitting and Variance for Simulating Discrete-Parameter Stochastic Processes," *Operations Research Letters*, Vol. 4, pp. 275–279.

KELTON, W. D. [1989], "Random Initialization Methods in Simulation," *IIE Transactions*, Vol. 21, pp. 355–367.

KELTON, W. D., AND A. M. LAW [1983], "A New Approach for Dealing with the Startup Problem in Discrete Event Simulation," *Naval Research Logistics Quarterly*, Vol. 30, pp. 641–658.

KLEIJNEN, J. P. C. [1987], *Statistical Tools for Simulation Practitioners*, Dekker, New York.

LAW, A. M. [1977], "Confidence Intervals in Discrete Event Simulation: A Comparison of Replication and Batch Means," *Naval Research Logistics Quarterly*, Vol. 24, pp. 667–78.

LAW, A. M. [1980], "Statistical Analysis of the Output Data from Terminating Simulations," *Naval Research Logistics Quarterly*, Vol. 27, pp. 131–43.

LAW, A. M., AND J. S. CARSON [1979], "A Sequential Procedure for Determining the Length of a Steady-State Simulation," *Operations Research*, Vol. 27, pp. 1011–1025.

LAW, A. M., AND W. D. KELTON [2000], *Simulation Modeling and Analysis*, 3d ed., McGraw-Hill, New York.

NELSON, B. L. [2001], "Statistical Analysis of Simulation Results," Chapter 94 in *Handbook of Industrial Engineering*, 3d ed., G. Salvendy, ed., Wiley, New York.

SCHMEISER, B. [1982], "Batch Size Effects in the Analysis of Simulation Output," *Operations Research*, Vol. 30, pp. 556–568.

SCHRUBEN, L. [1980], "Detecting Initialization Bias in Simulation Output," *Operations Research*, Vol. 30, pp. 569–590.

SNELL, M., AND L. SCHRUBEN [1985], "Weighting Simulation Data to Reduce Initialization Effects," *IIE Transactions*, Vol. 17, pp. 354–363.

STEIGER, N. M., AND J. R. Wilson [2002], "An Improved Batch Means Procedure for Simulation Output Analysis," *Management Science*, Vol. 48, pp. 1569–1586.

WELCH, P. D. [1983], "The Statistical Analysis of Simulation Results," in *The Computer Performance Modeling Handbook*, S. Lavenberg, ed., Academic Press, New York, pp. 268–328.

EXERCISES

1. Suppose that, in Example 11.14, the simulation analyst decided to investigate the bias by using batch means over a batching interval of 2000 minutes. By definition, a batch mean for the interval $[(j-1)\,2000,\ j(2000))$ is defined by

$$Y_j = \frac{1}{2000}\int_{(j-1)2000}^{j(2000)} L_Q(t)\,dt$$

   (a) Show algebraically that such a batch mean can be obtained from two adjacent batch means over the two halves of the interval.
   (b) Compute the seven averaged batch means for the intervals [0, 2000), [2000, 4000), ... for the $M/G/1$ simulation. Use the data ($\bar{Y}_j$) in Table 11.6 (ignoring $\bar{Y}_{15} = 8.76$).
   (c) Draw plots of the type used in Figures 11.4 and 11.5. Does it still appear that deletion of the data over [0, 2000) (the first "new" batch mean) is sufficient to remove most of the point-estimator bias?

2. Suppose, in Example 11.14, that the simulation analyst could only afford to run 5 independent replications (instead of 10). Use the batch means in Table 11.5 for replications 1 to 5 to compute a 95% confidence interval for mean queue length L_Q. Investigate deletion of initial data. Compare the results from using 5 replications with those from using 10 replications.

3. In Example 11.7, suppose that management desired 95% confidence in the estimate of mean system time w and that the error allowed was $\varepsilon = 0.4$ minute. Using the same initial sample of size $R_0 = 4$ (given in Table 11.1), figure out the required total sample size.

4. Simulate the dump-truck problem in Example 3.4. At first, make the run length $T_E = 40$ hours. Make four independent replications. Compute a 90% confidence interval for mean cycle time, where a cycle time for a given truck is the time between its successive arrivals to the loader. Investigate the effect of different initial conditions (all trucks initially at the loader queue, versus all at the scale, versus all traveling, versus the trucks distributed throughout the system in some manner).

5. Consider an (M, L) inventory system, in which the procurement quantity, Q, is defined by

$$Q = \begin{cases} M - I & \text{if } I < L \\ 0 & \text{if } I \geq L \end{cases}$$

   where I is the level of inventory on hand plus on order at the end of a month, M is the maximum inventory level, and L is the reorder point. M and L are under management control, so the pair (M, L) is called the inventory policy. Under certain conditions, the analytical solution of such a model is possible, but the computational effort can be prohibitive. Use simulation to investigate an (M, L) inventory system with the following properties: The inventory status is checked at the end of each month. Backordering is allowed at a cost of \$4 per item short per month. When an order arrives, it will first be used to relieve the backorder. The lead time is given by a uniform distribution on the interval (0.25, 1.25) months. Let the beginning inventory level stand at 50 units, with no orders outstanding. Let the holding cost be \$1 per unit in inventory per month. Assume that the inventory position is reviewed each month. If an order is placed, its cost is \$60 + \$5Q, where \$60 is the ordering cost and \$5 is the cost of each item. The time between demands is exponentially distributed with a mean of 1/15 month. The sizes of the demands follow this distribution:

Demand	*Probability*
1	1/2
2	1/4
3	1/8
4	1/8

(a) Make four independent replications, each of run length 100 months preceded by a 12-month initialization period, for the $(M, L) = (50, 30)$ policy. Estimate long-run mean monthly cost with a 90% confidence interval.

(b) Using the results of part (a), estimate the total number of replications needed to estimate mean monthly cost within \$5.

6. Reconsider Exercise 6, except that, if the inventory level at a monthly review is zero or negative, a rush order for Q units is placed. The cost for a rush order is $\$120+\$12Q$, where \$120 is the ordering cost and \$12 is the cost of each item. The lead time for a rush order is given by a uniform distribution on the interval (0.10, 0.25) months.

(a) Make four independent replications for the (M, L) policy, and estimate long-run mean monthly cost with a 90% confidence interval.

(b) Using the results of part (a), estimate the total number of replications needed to estimate mean monthly cost within \$5.

7. Suppose that the items in Exercise 6 are perishable, with a selling price given by the following data:

On the Shelf (Months)	*Selling Price*
0-1	\$10
1-2	5
>2	0

Thus, any item that has been on the shelf greater than 2 months cannot be sold. The age is measured at the time the demand occurs. If an item is outdated, it is discarded, and the next item is brought forward. Simulate the system for 100 months.

(a) Make four independent replications for the $(M, L) = (50, 30)$ policy, and estimate long-run mean monthly cost with a 90% confidence interval.

(b) Using the results of part (a), estimate the total number of replications needed to estimate mean monthly cost within \$5.

At first, assume that all the items in the beginning inventory are fresh. Is this a good assumption? What effect does this "all-fresh" assumption have on the estimates of long- run mean monthly cost? What can be done to improve these estimates? Carry out a complete analysis.

8. Consider the following inventory system:

(a) Whenever the inventory level falls to or below 10 units, an order is placed. Only one order can be outstanding at a time.

(b) The size of each order is Q. Maintaining an inventory costs \$0.50 per day per item in inventory. Placing an order incurs a fixed cost, \$10.00.

(c) Lead time is distributed in accordance with a discrete uniform distribution between zero and 5 days.

(d) If a demand occurs during a period when the inventory level is zero, the sale is lost at a cost of \$2.00 per unit.

(e) The number of customers each day is given by the following distribution:

Number of Customers per Day	*Probability*
1	0.23
2	0.41
3	0.22
4	0.14

(f) The demand on the part of each customer is Poisson distributed with a mean of 3 units.

(g) For simplicity, assume that all demands occur at noon and that all orders are placed immediately thereafter.

Assume further that orders are received at 5:00 P.M., or after the demand that occurred on that day. Consider the policy having $Q = 20$. Make five independent replications, each of length 100 days, and compute a 90% confidence interval for long-run mean daily cost. Investigate the effect of initial inventory level and existence of an outstanding order on the estimate of mean daily cost. Begin with an initial inventory of $Q + 10$ and no outstanding orders.

9. 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.

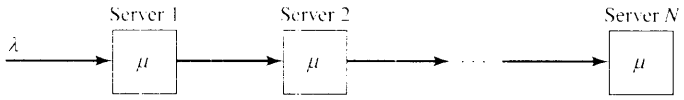
Any cards left after 2 weeks are sold for \$0.25. The card-shop owner is not sure how many cards can be sold, but thinks it is somewhere (i.e., uniformly distributed) between 200 and 400. Suppose that the card-shop owner decides to order 300 cards. Estimate the expected total profit with an error of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

10. A very large mining operation has decided to control the inventory of high-pressure piping by a "periodic review, order up to M" policy, where M is a target level. The annual demand for this piping is normally distributed, with mean 600 and variance 800. This demand occurs fairly uniformly over the year. The lead time for resupply is Erlang distributed of order $k = 2$ with its mean at 2 months. The cost of each unit is \$400. The inventory carrying charge, as a proportion of item cost on an annual basis, is expected to fluctuate normally about the mean 0.25 (simple interest), with a standard deviation of 0.01. The cost of making a review and placing an order is \$200, and the cost of a backorder is estimated to be \$100 per unit backordered. Suppose that the inventory level is reviewed every 2 months, and let $M = 337$.

(a) Make five independent replications, each of run length 100 months, to estimate long-run mean monthly cost by means of a 90% confidence interval.

(b) Investigate the effects of initial conditions. Calculate an appropriate number of monthly observations to delete to reduce initialization bias to a negligible level.

11. Consider some number, say N, of $M/M/1$ queues in series. The $M/M/1$ queue, described in Section 6.4, has Poisson arrivals at some rate λ customers per hour, exponentially distributed service times with mean $1/\mu$, and a single server. (Recall that "Poisson arrivals" means that interarrival times are exponentially distributed.) By $M/M/1$ queues in series, it is meant that, upon completion of service at a given server, a customer joins a waiting line for the next server. The system can be shown as follows:



All service times are exponentially distributed with mean $1/\mu$, and the capacity of each waiting line is assumed to be unlimited. Assume that $\lambda = 8$ customers per hour and $1/\mu = 0.1$ hour. The measure of performance is response time, which is defined to be the total time a customer is in the system.

(a) By making appropriate simulation runs, compare the initialization bias for $N = 1$ (i.e., one $M/M/1$ queue) to $N = 2$ (i.e., two $M/M/1$ queues in series). Start each system with all servers idle and no customers present. The purpose of the simulation is to estimate mean response time.

(b) Investigate the initialization bias as a function of N, for $N = 1, 2, 3, 4$, and 5.

(c) Draw some general conclusions concerning initialization bias for "large" queueing systems when at time 0 the system is assumed to be empty and idle.

12. Jobs enter a job shop in random fashion according to a Poisson process at a stationary overall rate, two every 8-hour day. The jobs are of four types. They flow from work station to work station in a fixed order, depending on type, as shown next. The proportions of each type are also shown.

Type	*Flow through Stations*	*Proportion*
1	1, 2, 3, 4	0.4
2	1, 3, 4	0.3
3	2, 4, 3	0.2
4	1, 4	0.1

Processing times per job at each station depend on type, but all times are (approximately) normally distributed with mean and s.d. (in hours) as follows:

	Station			
Type	*1*	*2*	*3*	*4*
1	(20, 3)	(30, 5)	(75, 4)	(20, 3)
2	(18, 2)		(60, 5)	(10, 1)
3		(20, 2)	(50, 8)	(10, 1)
4	(30, 5)			(15, 2)

Station i will have c_i workers ($i = 1, 2, 3, 4$). Each job occupies one worker at a station for the duration of a processing time. All jobs are processed on a first-in–first-out basis, and all queues for waiting jobs are assumed to have unlimited capacity. Simulate the system for 800 hours, preceded by a 200-hour initialization period. Assume that $c_1 = 8$, $c_2 = 8$, $c_3 = 20$, $c_4 = 7$. Based on $R = 5$ replications, compute a 97.5% confidence interval for average worker utilization at each of the four stations. Also, compute a

95% confidence interval for mean total response time for each job type, where a total response time is the total time that a job spends in the shop.

13. Change Exercise 12 to give priority at each station to the jobs by type. Type 1 jobs have priority over type 2, type 2 over type 3, and type 3 over type 4. Use 800 hours as run length, 200 hours as initialization period, and $R = 5$ replications. Compute four 97.5% confidence intervals for mean total response time by type. Also, run the model without priorities and compute the same confidence intervals. Discuss the trade-offs when using *first in, first out* versus a priority system.

14. Consider a single-server queue with Poisson arrivals at rate $\lambda = 10.82$ per minute and normally distributed service times with mean 5.1 seconds and variance 0.98 seconds^2. It is desired to estimate the mean time in the system for a customer who, upon arrival, finds i other customers in the system—that is, to estimate

$$w_i = E(W \mid N = i) \qquad \text{for } i = 0, 1, 2, \ldots$$

where W is a typical system time and N is the number of customers found by an arrival. For example, w_0 is the mean system time for those customers who find the system empty, w_1 is the mean system time for those customers who find one other customer present upon arrival, and so on. The estimate $\hat{w}_i$ of w_i will be a sample mean of system times taken over all arrivals who find i in the system. Plot $\hat{w}_i$ vs i. Hypothesize and attempt to verify a relation between w_i and i.

(a) Simulate for a 10-hour period with empty and idle initial conditions.
(b) Simulate for a 10-hour period after an initialization of one hour. Are there observable differences in the results of (a) and (b)?
(c) Repeat parts (a) and (b) with service times exponentially distributed with mean 5.1 seconds.
(d) Repeat parts (a) and (b) with deterministic service times equal to 5.1 seconds.
(e) Find the number of replications needed to estimate $w_0, w_1, \ldots, w_6$ with a standard error for each of at most 3 seconds. Repeat parts (a)–(d), but using this number of replications.

15. At Smalltown U., there is one specialized graphics workstation for student use located across campus from the computer center. At 2:00 A.M. one day, six students arrive at the workstation to complete an assignment. A student uses the workstation for 10 ± 8 minutes, then leaves to go to the computer center to pick up graphics output. There is a 25% chance that the run will be OK and the student will go to sleep. If it is not OK, the student returns to the workstation and waits until it becomes free. The roundtrip from workstation to computer center and back takes 30 ± 5 minutes. The computer becomes inaccessible at 5:00 A.M. Estimate the probability, p, that at least five of the six students will finish their assignment in the 3-hour period. First, make $R = 10$ replications, and compute a 95% confidence interval for p. Next, work out the number of replications needed to estimate p within $\pm.02$, and make this number of replications. Recompute the 95% confidence interval for p.

16. Four workers are spaced evenly along a conveyor belt. Items needing processing arrive according to a Poisson process at the rate 2 per minute. Processing time is exponentially distributed, with mean 1.6 minutes. If a worker becomes idle, then he or she takes the first item to come by on the conveyor. If a worker is busy when an item comes by, that item moves down the conveyor to the next worker, taking 20 seconds between two successive workers. When a worker finishes processing an item, the item leaves the system. If an item passes by the last worker, it is recirculated on a loop conveyor and will return to the first worker after 5 minutes.

Management is interested in having a balanced workload—that is, management would like worker utilizations to be equal. Let ρ_i be the long-run utilization of worker i, and let ρ be the average utilization of all workers. Thus, $\rho = (\rho_1 + \rho_2 + \rho_3 + \rho_4)/4$. According to queueing theory, ρ can be estimated

by $\rho = \lambda/c\mu$, where $\lambda = 2$ arrivals per minute, $c = 4$ servers, and $1/\mu = 1.6$ minutes is the mean service time. Thus, $\rho = \lambda/c\mu = (2/4)1.6 = 0.8$; so, on the average, a worker will be busy 80% of the time.

(a) Make 5 independent replications, each of run length 40 hours preceded by a one hour initialization period. Compute 95% confidence intervals for ρ_1 and ρ_4. Draw conclusions concerning workload balance.

(b) Based on the same 5 replications, test the hypothesis $H_0 : \rho_1 = 0.8$ at a level of significance $\alpha = 0.05$. If a difference of $\pm.05$ is important to detect, determine the probability that such a deviation is detected. In addition, if it is desired to detect such a deviation with probability at least 0.9, figure out the sample size needed to do so. (See any basic statistics textbook for guidance on hypothesis testing.)

(c) Repeat (b) for $H_0 : \rho_4 = 0.8$.

(d) From the results of (a)–(c), draw conclusions for management about the balancing of workloads.

17. At a small rock quarry, a single power shovel dumps a scoop full of rocks at the loading area approximately every 10 minutes, with the actual time between scoops modeled well as being exponentially distributed, with mean 10 minutes. Three scoops of rocks make a pile; whenever one pile of rocks is completed, the shovel starts a new pile.

The quarry has a single truck that can carry one pile (3 scoops) at a time. It takes approximately 27 minutes for a pile of rocks to be loaded into the truck and for the truck to be driven to the processing plant, unloaded, and return to the loading area. The actual time to do these things (altogether) is modeled well as being normally distributed, with mean 27 minutes and standard deviation 12 minutes.

When the truck returns to the loading area, it will load and transport another pile if one is waiting to be loaded; otherwise, it stays idle until another pile is ready. For safety reasons, no loading of the truck occurs until a complete pile (all three scoops) is waiting.

The quarry operates in this manner for an 8-hour day. We are interested in estimating the utilization of the trucks and the expected number of piles waiting to be transported if an additional truck is purchased.

18. Big Bruin, Inc. plans to open a small grocery store in Juneberry, NC. They expect to have two checkout lanes, with one lane being reserved for customers paying with cash. The question they want to answer is: how many grocery carts do they need?

During business hours (6 A.M.–8 P.M.), cash-paying customers are expected to arrive at 8 per hour. All other customers are expected to arrive at 9 per hour. The time between arrivals of each type can be modeled as exponentially distributed random variables.

The time spent shopping is modeled as normally distributed, with mean 40 minutes and standard deviation 10 minutes. The time required to check out after shopping can be modeled as lognormally distributed, with (a) mean 4 minutes and standard deviation 1 minute for cash-paying customers; (b) mean 6 minutes and standard deviation 1 minute for all other customers.

We will assume that every customer uses a shopping cart and that a customer who finishes shopping leaves the cart in the store so that it is available immediately for another customer. We will also assume that any customer who cannot obtain a cart immediately leaves the store, disgusted.

The primary performance measures of interest to Big Bruin are the expected number of shopping carts in use and the expected number of customers lost per day. Recommend a number of carts for the store, remembering that carts are expensive, but so are lost customers.

19. Develop a simulation model of the total time in the system for an $M/M/1$ queue with service rate $\mu = 1$; therefore, the traffic intensity is $\rho = \lambda/\mu = \lambda$, the arrival rate. Use the simulation, in conjunction with

the technique of plotting ensemble averages, to study the effect of traffic intensity on initialization bias when the queue starts empty. Specifically, see how the initialization phase T_0 changes for $\rho = 0.5, 0.7, 0.8, 0.9, 0.95$.

20. The average waiting data from 10 replication of a queuing system are

Replication	*Average Waiting Time*
1	1.77
2	2.50
3	1.87
4	3.22
5	3.00
6	2.11
7	3.12
8	3.49
9	2.39
10	3.49

Determine 90% confidence interval for the average waiting time.

21. Consider Example 6. If it is required to estimate the average waiting time with an absolute error of 0.25 and confidence level of 90%, determine the number of replications required.

22. In a queuing simulation with 20 replications, 90% confidence interval for average queue length is found to be in the range 1.72–2.41. Determine the probability that the average queue length is less than 2.75.

23. Collect papers dealing with simulation output analysis and study the tools used.

12

Comparison and Evaluation of Alternative System Designs

Chapter 11 dealt with the precise estimation of a measure of performance for one system. This chapter discusses a few of the many statistical methods that can be used to compare two or more system designs on the basis of some performance measure. One of the most important uses of simulation is the comparison of alternative system designs. Because the observations of the response variables contain random variation, statistical analysis is needed to discover whether any observed differences are due to differences in design or merely to the random fluctuation inherent in the models.

The comparison of two system designs is computationally easier than the simultaneous comparison of multiple (more than two) system designs. Section 12.1 discusses the case of two system designs, using two possible statistical techniques: *independent sampling* and *correlated sampling*. Correlated sampling is also known as the *common random numbers* (CRN) technique; simply put, the same random numbers are used to simulate both alternative system designs. If implemented correctly, CRN usually reduces the variance of the estimated difference of the performance measures and thus can provide, for a given sample size, more precise estimates of the mean difference than can independent sampling. Section 12.2 extends the statistical techniques of Section 12.1 to the comparison of multiple (more than two) system designs, using the Bonferroni approach to confidence-interval estimation, screening, and selecting the best. The Bonferroni approach is limited to twenty or fewer system designs, but Section 12.3 describes how a large number of complex system designs can sometimes be represented by a simpler metamodel. Finally, for comparison and evaluation of a very large number of system designs that are related in a less structured way, Section 12.4 presents optimization via simulation.

12.1 COMPARISON OF TWO SYSTEM DESIGNS

Suppose that a simulation analyst desires to compare two possible configurations of a system. In a queueing system, perhaps two possible queue disciplines, or two possible sets of servers, are to be compared. In a supply-chain inventory system, perhaps two possible ordering policies will be compared. A job shop could have two possible scheduling rules; a production system could have in-process inventory buffers of various capacities. Many other examples of alternative system designs can be provided.

The method of replications will be used to analyze the output data. The mean performance measure for system i will be denoted by $\theta_i (i = 1,2)$. If it is a steady-state simulation, it will be assumed that deletion of data, or other appropriate techniques, have been used to ensure that the point estimators are approximately unbiased estimators of the mean performance measures, θ_i. The goal of the simulation experiments is to obtain point and interval estimates of the difference in mean performance, namely $\theta_1 - \theta_2$. Three methods of computing a confidence interval for $\theta_1 - \theta_2$ will be discussed, but first an example and a general framework will be given.

Example 12.1

A vehicle-safety inspection station performs three jobs: (1) brake check, (2) headlight check, and (3) steering check. The present system has three stalls in parallel; that is, a vehicle enters a stall, where one attendant makes all three inspections. The current system is illustrated in Figure 12.1(a). Using data from the existing system, it has been assumed that arrivals occur completely at random (i.e., according to a Poisson process) at an average rate of 9.5 per hour and that the times for a brake check, a headlight check, and a steering check are normally distributed with means of 6.5, 6, and 5.5 minutes, respectively, all having standard deviations of approximately 0.5 minute. There is no limit on the queue of waiting vehicles.

An alternative system design is shown in Figure 12.1(b). Each attendant will specialize in a single task, and each vehicle will pass through three work stations in series. No space is allowed for vehicles between the brake and headlight check, or between the headlight and steering check. Therefore, a vehicle in the brake or headlight check must move to the next attendant, and a vehicle in the steering check must exit before the next vehicle can move ahead. The increased specialization of the inspectors suggests that mean inspection times for each type of check will decrease by 10%: to 5.85, 5.4, and 4.95 minutes, respectively, for the brake, headlight, and steering inspections. The Safety Inspection Council has decided to compare the two systems on the basis of mean response time per vehicle, where a response time is defined as the total time from a vehicle arrival until its departure from the system.

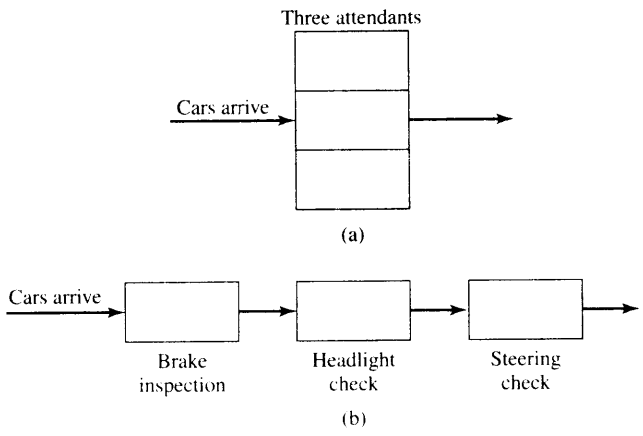


Figure 12.1 Vehicle safety inspection station and a possible alternative design.

When comparing two systems, such as those in Example 12.1, the simulation analyst must decide on a run length $T_E^{(i)}$ for each model ($i = 1, 2$), and a number of replications R_i to be made of each model. From replication r of system i, the simulation analyst obtains an estimate Y_{ri} of the mean performance measure, θ_i. In Example 12.1, Y_{ri} would be the average response time observed during replication r for system i ($r = 1, \ldots, R_i$; $i = 1, 2$). The data, together with the two summary measures, the sample means $\overline{Y}_i$, and the sample variances S_i^2, are exhibited in Table 12.1. Assuming that the estimators Y_{ri} are (at least approximately) unbiased, it follows that

$$\theta_1 = E(Y_{r1}),\ r = 1, \ldots, R_1;\ \theta_2 = E(Y_{r2}),\ r = 1, \ldots, R_2$$

In Example 12.1, the Safety Inspection Council is interested in a comparison of the two system designs, so the simulation analyst decides to compute a confidence interval for $\theta_1 - \theta_2$, the difference between the two mean performance measures. The confidence interval is used to answer two questions: (1) How large is the mean difference, and how precise is the estimator of mean difference? (2) Is there a significant difference between the two systems? This second question will lead to one of three possible conclusions:

1. If the confidence interval (c.i.) for $\theta_1 - \theta_2$ is totally to the left of zero, as shown in Figure 12.2(a), then there is strong evidence for the hypothesis that $\theta_1 - \theta_2 < 0$, or equivalently $\theta_1 < \theta_2$.

Table 12.1 Simulation Output Data and Summary Measures for Comparing Two Systems

System	Replication 1	2	...	R_i	Sample Mean	Sample Variance
1	Y_{11}	Y_{21}	...	$Y_{R1}1$	$\overline{Y}_1$	S_1^2
2	Y_{12}	Y_{22}	...	$Y_{R2}2$	$\overline{Y}_2$	S_2^2

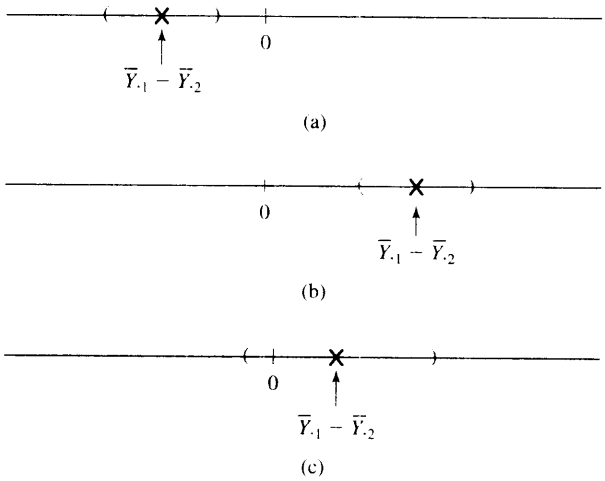


Figure 12.2 Three confidence intervals that can occur in the comparing of two systems.

In Example 12.1, $\theta_1 < \theta_2$ implies that the mean response time for system 1 (the original system) is smaller than for system 2 (the alternative system).

2. If the c.i. for $\theta_1 - \theta_2$ is totally to the right of zero, as shown in Figure 12.2(b), then there is strong evidence that $\theta_1 - \theta_2 > 0$, or equivalently, $\theta_1 > \theta_2$.

In Example 12.1, $\theta_1 > \theta_2$ can be interpreted as system 2 being better than system 1, in the sense that system 2 has smaller mean response time.

3. If the c.i. for $\theta_1 - \theta_2$ contains zero, then, in the data at hand, there is no strong statistical evidence that one system design is better than the other.

Some statistics textbooks say that the weak conclusion $\theta_1 = \theta_2$ can be drawn, but such statements can be misleading. A "weak" conclusion is often no conclusion at all. Most likely, if enough additional data were collected (i.e., R_i increased), the c.i. would shift, and definitely shrink in length, until conclusion 1 or 2 would be drawn. In addition to one of these three conclusions, the confidence interval provides a measure of the precision of the estimator of $\theta_1 - \theta_2$.

In this chapter, a two-sided $100(1-\alpha)\%$ c.i. for $\theta_1 - \theta_2$ will always be of the form

$$\bar{Y}_{\cdot 1} - \bar{Y}_{\cdot 2} \pm t_{\alpha/2,\nu}\,\text{s.e.}(\bar{Y}_{\cdot 1} - \bar{Y}_{\cdot 2}) \tag{12.1}$$

where $\bar{Y}_{\cdot i}$ is the sample mean performance measure for system i over all replications

$$\bar{Y}_{\cdot i} = \frac{1}{R_i}\sum_{r=1}^{R_i} Y_{ri} \tag{12.2}$$

and ν is the degrees of freedom associated with the variance estimator, $t_{\alpha/2,\nu}$ is the $100(1 - \alpha/2)$ percentage point of a t distribution with ν degrees of freedom, and s.e.$(\cdot)$ represents the standard error of the specified point estimator. To obtain the standard error and the degrees of freedom, the analyst uses one of three statistical techniques. All three techniques assume that the basic data, Y_{ri} of Table 12.1, are approximately normally distributed. This assumption is reasonable provided that each Y_{ri} is itself a sample mean of observations from replication r (which is indeed the situation in Example 12.1).

By design of the simulation experiment, $Y_{r1}(r = 1, \ldots, R_1)$ are independently and identically distributed (i.i.d.) with mean θ_1 and variance σ_1^2 (say). Similarly, $Y_{r2}(r = 1, \ldots, R_2)$ are i.i.d. with mean θ_2 and variance σ_2^2 (say). The three techniques for computing the confidence interval in (12.1), which are based on three different sets of assumptions, are discussed in the following subsections.

There is an important distinction between *statistically significant* differences and *practically significant* differences in systems performance. Statistical significance answers the following question: Is the observed difference $\bar{Y}_1 - \bar{Y}_2$ larger than the variability in $\bar{Y}_1 - \bar{Y}_2$? This question can be restated as: Have we collected enough data to be confident that the difference we observed is real, or just chance? Conclusions 1 and 2 imply a statistically significant difference, while Conclusion 3 implies that the observed difference is not statistically significant (even though the systems may indeed be different). Statistical significance is a function of the simulation experiment and the output data.

Practical significance answers the following question: Is the true difference $\theta_1 - \theta_2$ large enough to matter for the decision we need to make? In Example 12.1, we may reach the conclusion that $\theta_1 > \theta_2$ and decide that system 2 is better (smaller expected response time). However, if the actual difference $\theta_1 - \theta_2$ is very small—say, small enough that a customer would not notice the improvement— then it might not be worth the cost to replace system 1 with system 2. Practical significance is a function of the actual difference between the systems and is independent of the simulation experiment.

Confidence intervals do not answer the question of practical significance directly. Instead, they bound (with probability $1 - \alpha$) the true difference $\theta_1 - \theta_2$ within the range

$$\bar{Y}_{\cdot 1} - \bar{Y}_{\cdot 2} - t_{\alpha 2, \nu} \text{ s.e.}(\bar{Y}_{\cdot 1} - \bar{Y}_{\cdot 2}) \leq \theta_1 - \theta_2 \leq \bar{Y}_{\cdot 1} - \bar{Y}_{\cdot 2} + t_{\alpha 2, \nu} \text{ s.e.}(\bar{Y}_{\cdot 1} - \bar{Y}_{\cdot 2})$$

Whether a difference within these bounds is practically significant depends on the particular problem.

12.1.1 Independent Sampling with Equal Variances

Independent sampling means that different and independent random number streams will be used to simulate the two systems. This implies that all the observations of simulated system 1, namely $\{Y_{r1}, r = 1, \ldots, R_1\}$, are statistically independent of all the observations of simulated system 2, namely $\{Y_{r2}, r = 1, \ldots, R_2\}$. By Equation (12.2) and the independence of the replications, the variance of the sample mean, $\bar{Y}_{\cdot i}$, is given by

$$V(\bar{Y}_{\cdot i}) = \frac{V(Y_{ri})}{R_i} = \frac{\sigma_i^2}{R_i}, \quad i = 1, 2$$

For independent sampling, $\bar{Y}_{\cdot 1}$ and $\bar{Y}_{\cdot 2}$ are statistically independent; hence,

$$\begin{aligned} V(\bar{Y}_{\cdot 1} - \bar{Y}_{\cdot 2}) &= V(\bar{Y}_{\cdot 1}) + V(\bar{Y}_{\cdot 2}) \\ &= \frac{\sigma_1^2}{R_1} + \frac{\sigma_2^2}{R_2} \end{aligned} \tag{12.3}$$

In some cases, it is reasonable to assume that the two variances are equal (but unknown in value); that is, $\sigma_1^2 = \sigma_2^2$. The data can be used to test the hypothesis of equal variances; if rejected, the method of Section 12.1.2 must be used. In a steady-state simulation, the variance σ_i^2 decreases as the run length $T_E^{(i)}$ increases; therefore, it might be possible to adjust the two run lengths, $T_E^{(1)}$ and $T_E^{(2)}$, to achieve at least approximate equality of σ_1^2 and σ_2^2.

If it is reasonable to assume that $\sigma_1^2 = \sigma_2^2$ (approximately), a two-sample-t confidence-interval approach can be used. The point estimate of the mean performance difference is

$$\bar{Y}_{\cdot 1} - \bar{Y}_{\cdot 2} \tag{12.4}$$

with $\bar{Y}_{\cdot i}$ given by Equation (12.2). Next, compute the sample variance for system i by

$$\begin{aligned} S_i^2 &= \frac{1}{R_i - 1} \sum_{r=1}^{R_i} (Y_{ri} - \bar{Y}_{\cdot i})^2 \\ &= \frac{1}{R_i - 1} \left(\sum_{r=1}^{R_i} Y_{ri}^2 - R_i \bar{Y}_{\cdot i}^2 \right) \end{aligned} \tag{12.5}$$

Note that S_i^2 is an unbiased estimator of the variance σ_i^2. By assumption, $\sigma_1^2 = \sigma_2^2 = \sigma^2$ (say), so a pooled estimate of σ^2 is obtained by

$$S_p^2 = \frac{(R_1 - 1)S_1^2 + (R_2 - 1)S_2^2}{R_1 + R_2 - 2}$$

which has $v = R_1 + R_2 - 2$ degrees of freedom. The c.i. for $\theta_1 - \theta_2$ is then given by Expression (12.1) with the standard error computed by

$$\text{s.e.}(\bar{Y}_1 - \bar{Y}_2) = S_p\sqrt{\frac{1}{R_1} + \frac{1}{R_2}} \tag{12.6}$$

This standard error is an estimate of the standard deviation of the point estimate, which, by Equation (12.3), is given by $\sigma\sqrt{1/R_1 + 1/R_2}$.

In some cases, the simulation analyst could have $R_1 = R_2$, in which case it is safe to use the c.i. in Expression (12.1) with the standard error taken from Equation (12.6), even if the variances (σ_1^2 and σ_2^2) are not equal. However, if the variances are unequal and the sample sizes differ, it has been shown that use of the two-sample-t c.i. could yield invalid confidence intervals whose true probability of containing $\theta_1 - \theta_2$ is much less than $1 - \alpha$. Thus, if there is no evidence that $\sigma_1^2 = \sigma_2^2$, and if $R_1 \neq R_2$, the approximate procedure in the next subsection is recommended.

12.1.2 Independent Sampling with Unequal Variances

If the assumption of equal variances cannot safely be made, an approximate $100(1 - \alpha)\%$ c.i. for $\theta_1 - \theta_2$ can be computed as follows. The point estimate and sample variances are computed by Equations (12.4) and (12.5). The standard error of the point estimate is given by

$$\text{s.e.}(\bar{Y}_1 - \bar{Y}_2) = \sqrt{\frac{S_1^2}{R_1} + \frac{S_2^2}{R_2}} \tag{12.7}$$

with degrees of freedom, v, approximated by the expression

$$v = \frac{(S_1^2/R_1 + S_2^2/R_2)^2}{[(S_1^2/R_1)^2/(R_1 - 1)] + [(S_2^2/R_2)^2/(R_2 - 1)]} \tag{12.8}$$

rounded to an integer. The confidence interval is then given by Expression (12.1), using the standard error of Equation (12.7). A minimum number of replications $R_1 \geq 6$ and $R_2 \geq 6$ is recommended for this procedure.

12.1.3 Common Random Numbers (CRN)

CRN means that, for each replication, the same random numbers are used to simulate both systems. Therefore, R_1 and R_2 must be equal, say $R_1 = R_2 = R$. Thus, for each replication r, the two estimates, Y_{r1} and Y_{r2}, are no longer independent, but rather are correlated. However, independent streams of random numbers are used on different replications, so the pairs (Y_{r1}, Y_{s2}) are mutually independent when $r \neq s$. (For example, in Table 12.1, the observation Y_{11} is correlated with Y_{12}, but Y_{11} is independent of all other observations.) The purpose of using CRN is to induce a positive correlation between Y_{r1} and Y_{r2} (for each r) and thus to achieve a variance reduction in the point estimator of mean difference, $\bar{Y}_1 - \bar{Y}_2$. In general, this variance is given by

$$\begin{aligned} V(\bar{Y}_1 - \bar{Y}_2) &= V(\bar{Y}_1) + V(\bar{Y}_2) - 2\text{cov}(\bar{Y}_1, \bar{Y}_2) \\ &= \frac{\sigma_1^2}{R} + \frac{\sigma_2^2}{R} - \frac{2\rho_{12}\sigma_1\sigma_2}{R} \end{aligned} \tag{12.9}$$

where ρ_{12} is the correlation between Y_{r1} and Y_{r2}. [By definition, $\rho_{12} = \text{cov}(Y_{r1}, Y_{r2})/\sigma_1\sigma_2$, which does not depend on r.]

Now compare the variance of $\bar{Y}_1 - \bar{Y}_2$ arising from the use of CRN [Equation (12.9), call it V_{CRN} to the variance arising from the use of independent sampling with equal sample sizes [Equation (12.3) with $R_1 = R_2 = R$, call it V_{IND}]. Notice that

$$V_{CRN} = V_{IND} - \frac{2\rho_{12}\sigma_1\sigma_2}{R} \tag{12.10}$$

If CRN works as intended, the correlation ρ_{12} will be positive; hence, the second term on the right side of Equation (12.9) will be positive, and, therefore,

$$V_{CRN} < V_{IND}$$

That is, the variance of the point estimator will be smaller with CRN than with independent sampling. A smaller variance (for the same sample size R) implies that the estimator based on CRN is more precise, leading to a shorter confidence interval on the difference, which implies that smaller differences in performance can be detected.

To compute a $100(1 - \alpha)\%$ c.i. with correlated data, first compute the differences

$$D_r = Y_{r1} - Y_{r2} \tag{12.11}$$

which, by the definition of CRN, are i.i.d.; then compute the sample mean difference as

$$\bar{D} = \frac{1}{R}\sum_{r=1}^{R} D_r \tag{12.12}$$

(Thus, $\bar{D} = \bar{Y}_1 - \bar{Y}_2$.) The sample variance of the differences $\{D_r\}$ is computed as

$$\begin{aligned} S_D^2 &= \frac{1}{R-1}\sum_{r=1}^{R}(D_r - \bar{D})^2 \\ &= \frac{1}{R-1}\left(\sum_{r=1}^{R} D_r^2 - R\bar{D}^2\right) \end{aligned} \tag{12.13}$$

which has degrees of freedom $v = R - 1$. The $100(1 - \alpha)\%$ c.i. for $\theta_1 - \theta_2$ is given by Expression (12.1), with the standard error of $\bar{Y}_1 - \bar{Y}_2$ estimated by

$$\text{s.e.}(\bar{D}) = \text{s.e.}(\bar{Y}_1 - \bar{Y}_2) = \frac{S_D}{\sqrt{R}} \tag{12.14}$$

Because $S_D/\sqrt{R}$ of Equation (12.14) is an estimate of $\sqrt{V_{CRN}}$ and Expression (12.6) or (12.7) is an estimate of $\sqrt{V_{IND}}$, CRN typically will produce a c.i. that is shorter for a given sample size than the c.i. produced by independent sampling if $\rho_{12} > 0$. In fact, the expected length of the c.i. will be shorter with use of CRN if $\rho_{12} > 0.1$, provided $R > 10$. The larger R is, the smaller ρ_{12} can be and still yield a shorter expected length [Nelson 1987].

For any problem, there are many ways of implementing common random numbers. It is never enough to simply use the same seed on the random-number generator(s). Each random number used in one model for some purpose should be used for the same purpose in the second model—that is, the use of the random numbers must be synchronized. For example, if the ith random number is used to generate a service time at

work station 2 for the 5th arrival in model 1, the ith random number should be used for the very same purpose in model 2. For queueing systems or service facilities, synchronization of the common random numbers guarantees that the two systems face identical work loads: both systems face arrivals at the same instants of time, and these arrivals demand equal amounts of service. (The actual service times of a given arrival in the two models may not be equal; they could be proportional if the server in one model were faster than the server in the other model.) For an inventory system, in comparing of different ordering policies, synchronization guarantees that the two systems face identical demand for a given product. For production or reliability systems, synchronization guarantees that downtimes for a given machine will occur at exactly the same times and will have identical durations, in the two models. On the other hand, if some aspect of one of the systems is totally different from the other system, synchronization could be inappropriate—or even impossible to achieve. In summary, those aspects of the two system designs that are sufficiently similar should be simulated with common random numbers in such a way that the two models "behave" similarly; but those aspects that are totally different should be simulated with independent random numbers.

Implementation of common random numbers is model dependent, but certain guidelines can be given that will make CRN more likely to yield a positive correlation. The purpose of the guidelines is to ensure that synchronization occurs:

1. Dedicate a random-number stream to a specific purpose, and use as many different streams as needed. (Different random-number generators, or widely spaced seeds on the same generator, can be used to get two different, nonoverlapping streams.) In addition, assign independently chosen seeds to each stream at the beginning of each replication. It is not sufficient to assign seeds at the beginning of the first replication and then let the random-number generator merely continue for the second and subsequent replications. If simulation is conducted in this manner, the first replication will be synchronized, but subsequent replications might not be.
2. For systems (or subsystems) with external arrivals: As each entity enters the system, the next interarrival time is generated, and then immediately all random variables (such as service times, order sizes, etc.) needed by the arriving entity and identical in both models are generated in a fixed order and stored as attributes of the entity, to be used later as needed. Apply guideline 1: Dedicate one random-number stream to these external arrivals and all their attributes.
3. For systems having an entity performing given activities in a cyclic or repeating fashion, assign a random-number stream to this entity. (Example: a machine that cycles between two states: up–down–up–down–.... Use a dedicated random-number stream to generate the uptimes and downtimes.)
4. If synchronization is not possible, or if it is inappropriate for some part of the two models, use independent streams of random numbers for this subset of random variates.

Unfortunately, there is no guarantee that CRN will always induce a positive correlation between comparable runs of the two models. It is known that if, for each input random variate X, the estimators Y_{r1} and Y_{r2} are increasing functions of the random variate X (or both are decreasing functions of X), then ρ_{12} will be positive. The intuitive idea is that both models (i.e., both Y_{r1} and Y_{r2}) respond in the same direction to each input random variate, and this results in positive correlation. This increasing or decreasing nature of the response variables (called *monotonicity*) with respect to the input random variables is known to hold for certain queueing systems (such as the $GI/G/c$ queues), when the response variable is customer delay, so some evidence exists that common random numbers is a worthwhile technique for queueing simulations. (For simple queues, customer delay is an increasing function of service times and a decreasing function of interarrival times.) Wright and Ramsay [1979] reported a negative correlation for certain inventory simulations, however. In summary, the guidelines recently described should be followed, and some reasonable notion that the response variable of interest is a monotonic function of the random input variables should be evident.

Example 12.1: Continued

The two inspection systems shown in Figure 12.1 will be compared by using both independent sampling and CRN, in order to illustrate the greater precision of CRN when it works.

Each vehicle arriving to be inspected has four input random variables associated with it:

A_n = interarrival time between vehicles n and $n + 1$

$S_n^{(1)}$ = brake inspection time for vehicle n in model 1

$S_n^{(2)}$ = headlight inspection time for vehicle n in model 1

$S_n^{(3)}$ = steering inspection time for vehicle n in model 1

For model 2 (of the proposed system), mean service times are decreased by 10%. When using independent sampling, different values of service (and interarrival) times would be generated for models 1 and 2 by using different random numbers. But when using CRN, the random number generator should be used in such a way that exactly the same values are generated for $A_1, A_2, A_3, \ldots$ in both models. For service times, however, we do not want the same service times in both models, because the mean service time for model 2 is 10% smaller, but we do want strongly correlated service times. There are at least two ways to do this:

1. Let $S_n^{(i)}(i = 1, 2, 3; n = 1, 2, \ldots)$ be the service times generated for model 1; then use $S_n^{(i)} - 0.1E(S_n^{(i)})$ as the service times in model 2. In words, we take each service time from model 1 and subtract 10% of its true mean.
2. Recall that normal random variates are usually produced by first generating a standard normal variate and then using Equation (8.29) to obtain the correct mean and variance. Therefore, the service times for, say, a brake inspection could be generated by

$$E(S_n^{(1)}) + \sigma Z_n^{(1)} \tag{12.15}$$

where $Z_n^{(1)}$ is a standard normal variate, $\sigma = 0.5$ minute, but $E(S_n^{(1)}) = 6.5$ minutes for model 1 and $E(S_n^{(1)}) = 5.85$ minutes (10% less) for model 2. The other two inspection times would be generated in a similar fashion. To implement (synchronized) common random numbers, the simulation analyst would generate identical $Z_n^{(i)}$ sequences ($i = 1, 2, 3$; $n = 1, 2, \ldots$) in both models and then use the appropriate version of Equation (12.15) to generate the inspection times.

For the synchronized runs, the service times for a vehicle were generated at the instant of arrival (by guideline 2) and stored as an attribute of the vehicle, to be used as needed. Runs were also made with non-synchronized common random numbers, in which case one random number stream was used as needed.

Table 12.2 gives the average response time for each of $R = 10$ replications, each of run length $T_E = 16$ hours. It was assumed that two cars were present at time 0, waiting to be inspected. Column 1 gives the outputs from model 1. Model 2 was run with independent random numbers (column 2I) and with common random numbers without synchronization (column 2C*) and with synchronization (column 2C). The purpose of the simulation is to estimate mean difference in response times for the two systems.

For the two independent runs (1 and 2I), it was assumed that the variances were not necessarily equal, so the method of Section 12.1.2 was applied. Sample variances and the standard error were computed by Equations (12.5) and (12.7), yielding

$$S_1^2 = 118.9, \qquad S_{2I}^2 = 244.3$$

and

$$\text{s.e.}(\bar{Y}_1 - \bar{Y}_{2I}) = \sqrt{\frac{118.9}{10} + \frac{244.3}{10}} = 6.03$$

Table 12.2 Comparison of System Designs for the Vehicle-Safety Inspection System

	Average Response Time for Model				*Observed Differences*	
Replication	*1*	*2I*	*2C**	*2C*	$D_{1,2C^*}$	$D_{1,2C}$
1	29.59	51.62	56.47	29.55	−26.88	0.04
2	23.49	51.91	33.34	24.26	−9.85	−0.77
3	25.68	45.27	35.82	26.03	−10.14	−0.35
4	41.09	30.85	34.29	42.64	6.80	−1.55
5	33.84	56.15	39.07	32.45	−5.23	1.39
6	39.57	28.82	32.07	37.91	7.50	1.66
7	37.04	41.30	51.64	36.48	−14.60	0.56
8	40.20	73.06	41.41	41.24	−1.21	−1.04
9	61.82	23.00	48.29	60.59	13.53	1.23
10	44.00	28.44	22.44	41.49	21.56	2.51
Sample mean	37.63	43.04			−1.85	0.37
Sample variance	118.90	244.33			208.94	1.74
Standard error	6.03				4.57	0.42

with degrees of freedom, ν, equal to 17, as given by Equation (12.8). The point estimate is $\bar{Y}_1 - \bar{Y}_{2I} = -5.4$ minutes, and a 95% c.i. [Expression (12.1)] is given by

$$-5.4 \pm 2.11(6.03)$$

or

$$-18.1 \leq \theta_1 - \theta_2 \leq 7.3 \tag{12.16}$$

The 95% confidence interval in Inequality (12.16) contains zero, which indicates that there is no strong evidence that the observed difference, −5.4 minutes, is due to anything other than random variation in the output data. In other words, it is not statistically significant. Thus, if the simulation analyst had decided to use independent sampling, no strong conclusion would be possible, because the estimate of $\theta_1 - \theta_2$ is quite imprecise.

For the two sets of correlated runs (1 and 2C*, and 1 and 2C), the observations are paired and analyzed as given in Equations (12.11) through (12.14). The point estimate when not synchronizing the random numbers is given by Equation (12.12) as

$$\bar{D} = -1.9 \text{ minutes}$$

the sample variance by S_D^2 (with $\nu = 9$ degrees of freedom), and the standard error by s.e.$(\bar{D}) = 4.6$. Thus, a 95% c.i. for the true mean difference in response times, as given by expression (12.1), is

$$-1.9 \pm 2.26(4.6)$$

or

$$-12.3 < \theta_1 - \theta_2 < 8.5 \tag{12.17}$$

Again, no strong conclusion is possible, because the confidence interval contains zero. Notice, however, that the estimate of $\theta_1 - \theta_2$ is slightly more precise than that in Inequality (12.16), because the length of the interval is smaller.

When complete synchronization of the random numbers was used, in run 2C, the point estimate of the mean difference in response times was

$$\bar{D} = 0.4 \text{ minute}$$

the sample variance was $S_D^2 = 1.7$ (with $\nu = 9$ degrees of freedom), and the standard error was s.e.$(\bar{D}) = 0.4$. A 95% c.i. for the true mean difference is given by

$$-0.50 < \theta_1 - \theta_2 < 1.30 \tag{12.18}$$

The confidence interval in Inequality (12.18) again contains zero, but it is considerably shorter than the previous two intervals. This greater precision in the estimation of $\theta_1 - \theta_2$ is due to the use of synchronized common random numbers. The short length of the interval in Inequality (12.18) suggests that the true difference, $\theta_1 - \theta_2$, is close to zero. In fact, the upper bound, 1.30, indicates that system 2 is at most 1.30 minutes faster, in expectation. If such a small difference is not practically significant, then there is no need to look further into which system is truly better.

As is seen by comparing the confidence intervals in inequalities (12.16), (12.17), and (12.18), the width of the confidence interval is reduced by 18% when using nonsynchronized common random numbers, by 93% when using common random numbers with full synchronization. Comparing the estimated variance of $\bar{D}$ when using synchronized common random numbers with the variance of $\overline{Y_1} - \overline{Y_2}$ when using independent sampling shows a variance reduction of 99.5%, which means that, to achieve precision comparable to that achieved by CRN, a total of approximately $R = 209$ independent replications would have to be made.

The next few examples show how common random numbers can be implemented in other contexts.

Example 12.2: The Dump-Truck Problem, Revisited

Consider Example 3.4 (the dump-truck problem), shown in Figure 3.7. Each of the trucks repeatedly goes through three activities: loading, weighing, and traveling. Assume that there are eight trucks and that, at time 0, all eight are at the loaders. Weighing time per truck on the single scale is uniformly distributed between 1 and 9 minutes, and travel time per truck is exponentially distributed, with mean 85 minutes. An unlimited queue is allowed before the loader(s) and before the scale. All trucks can be traveling at the same time. Management desires to compare one fast loader against the two slower loaders currently being used. Each of the slow loaders can fill a truck in from 1 to 27 minutes, uniformly distributed. The new fast loader can fill a truck in from 1 to 19 minutes, uniformly distributed. The basis for comparison is mean system response time, where a response time is defined as the duration of time from a truck arrival at the loader queue to that truck's departure from the scale.

To implement synchronized common random numbers, a separate and distinct random number stream was assigned to each of the eight trucks. At the beginning of each replication (i.e., at time 0), a new and independently chosen set of eight seeds was specified, one seed for each random number stream. Thus, weighing times and travel times for each truck were identical in both models, and the loading time for a given truck's ith visit to the fast loader was proportional to the loading time in the original system (with two slow loaders). Implementation of common random numbers without synchronization (e.g., using one random number stream to generate all loading, weighing, and travel times as needed) would likely lead to a given random number being used to generate a loading time in model 1 but a travel time in model 2, or vice versa, and from that point on the use of a random number would most likely be different in the two models.

Table 12.3 Comparison of System Designs for the Dump Truck Problem

Replication	*Average Response Time for Model*			*Differences,* $D_{1,2C}$
	1 (*2 Loaders*)	*2I* (*1 Loader*)	*2C* (*1 Loader*)	
1	21.38	29.01	24.30	−2.92
2	24.06	24.70	27.13	−3.07
3	21.39	26.85	23.04	−1.65
4	21.90	24.49	23.15	−1.25
5	23.55	27.18	26.75	−3.20
6	22.36	26.91	25.62	−3.26
Sample mean	22.44	26.52		−2.56
Sample variance	1.28	2.86		0.767
Sample standard deviation	1.13	1.69		0.876

Six replications of each model were run, each of run length $T_E = 40$ hours. The results are shown in Table 12.3. Both independent sampling and CRN were used, to illustrate the advantage of CRN. The first column (labeled model 1) contains the observed average system response time for the existing system with two loaders. The columns labeled 2I and 2C are for the alternative design having one loader; the independent sampling results are in 2I, and the CRN results are in the column labeled 2C. The rightmost column, labeled $D_{1,2C}$, lists the observed differences between the runs of model 1 and model 2C.

For independent sampling assuming unequal variances, the following summary statistics were computed by using Equations (12.2), (12.5), (12.7), (12.8), and (12.1) and the data (in columns 1 and 2I) in Table 12.3:

Point Estimate: $\bar{Y}_1 - \bar{Y}_{2I} = 22.44 - 26.52 = -4.08$ minutes

Sample variances: $S_1^2 = 1.28$, $S_{2I}^2 = 2.86$

Standard Error: $\text{s.e.}(\bar{Y}_1 - \bar{Y}_2) = (S_1^2 / R_1 + S_{2I}^2 / R_2)^{1/2} = 0.831$

Degrees of freedom: $\nu = 8.73 \approx 9$

95% c.i. for $\theta_1 - \theta_2$: $-4.08 \pm 2.26(0.831)$ or -4.08 ± 1.878

$$-5.96 \le \theta_1 - \theta_2 \le -2.20$$

For CRN, implemented by the use of synchronized common random numbers, the following summary statistics were computed by using Equations (12.12), (12.13), (12.14), and (12.1) plus the data (in columns 1 and 2C) in Table 12.3:

Point Estimate: $\bar{D} = \bar{Y}_1 - \bar{Y}_{2C} = -2.56$ minutes

Sample variance: $S_D^2 = 0.767$

Standard Error: $\text{s.e.}(\bar{D}) = S_D / \sqrt{R} = 0.876 / \sqrt{6} = 0.358$

Degrees of freedom: $\nu = R - 1 = 5$

95% c.i. for $\theta_1 - \theta_2$: $-2.56 \pm 2.57(0.358)$ or -2.56 ± 0.919

$$-3.48 \le \theta_1 - \theta_2 \le -1.641$$

By comparing the c.i. widths, we see that the use of CRN with synchronization reduced c.i. width by 50%. This reduction could be important if a difference of as much as, say, 5.96 is considered practically

significant, but a difference of at most 3.48 is not. Equivalently, if equal precision were desired, independent sampling would require approximately four times as many observations as would CRN: approximately 24 replications of each model instead of six.

To illustrate how CRN can fail when not implemented correctly, consider the dump-truck model again. There were eight trucks, and each was assigned its own random number stream. For each of the six replications, eight seeds were randomly chosen, one seed for each random number stream. Therefore, a total of 48 (6 times 8) seeds were specified for the correct implementation of common random numbers. When the authors first developed and ran this example, eight seeds were specified at the beginning of the first replication only; on the remaining five replications the random numbers were generated by continuing down the eight original streams. Since comparable replications with one and two loaders required different numbers of random variables, only the first replications of the two models were synchronized. The remaining five were not synchronized. The resulting confidence interval for $\theta_1 - \theta_2$ under CRN was approximately the same length as, or only slightly shorter than, the confidence interval under independent sampling. Therefore, CRN is quite likely to fail in reducing the standard error of the estimated difference unless proper care is taken to guarantee synchronization of the random number streams on all replications.

Example 12.3

In Example 2.5, two policies for replacing bearings in a milling machine were compared. The bearing-life distribution, assumed discrete in Example 2.5 (Table 2.22), is now more realistically assumed to be continuous on the range from 950 to 1950 hours, with the first column of Table 2.22 giving the midpoint of 10 intervals of width 100 hours. The repairperson delay-time distribution of Table 2.23 is also assumed continuous, in the range from 2.5 to 17.5 minutes, with interval midpoints as given in the first column. The probabilities of each interval are given in the second columns of Tables 2.22 and 2.23.

The two models were run by using CRN and, for illustrative purposes, by using independent sampling, each for $R = 10$ replications. The purpose was to estimate the difference in mean total costs per 10,000 bearing hours, with the cost data given in Example 2.5. The estimated total cost for the two policies is given in Table 12.4.

Table 12.4 Total Costs for Alternative Designs of Bearing Replacement Problem

Replication r	*Total Cost for Policy*			*Difference in Total Cost*
	2	*II*	*IC*	$D_{IC,2}$
1	13,340	17,010	17,556	4,216
2	12,760	17,528	17,160	4,400
3	13,002	17,956	17,808	4,806
4	13,524	17,920	18,012	4,488
5	13,754	18,880	18,200	4,446
6	13,318	17,528	17,936	4,618
7	13,432	17,574	18,350	4,918
8	14,208	17,954	19,398	5,190
9	13,224	18,290	17,612	4,388
10	13,178	17,360	17,956	4,778
Sample mean	13,374	17,800		4,624
Sample variance	160,712	276,188		87,353

Policy 1 was to replace each bearing as it failed. Policy 2 was to replace all three bearings whenever one bearing failed. Policy 2 was run first, and then policy 1 was run, using independent sampling (column 1I), and using CRN (column 1C). The 95% confidence intervals for mean cost difference are as follows:

Independent sampling: \$4426 ± 439
CRN: \$4625 ± 211

(The computation of these confidence intervals is left as an exercise for the reader.)

Notice that the confidence interval for mean cost difference when using CRN is approximately 50% of the length of the confidence interval based on independent sampling. Therefore, for the same computer costs, (i.e., for $R = 10$ replications), CRN produces estimates that are twice as precise in this example. If CRN were used, the simulation analyst could conclude with 95% confidence that the mean cost difference between the two policies is between \$4414 and \$4836.

12.1.4 Confidence Intervals with Specified Precision

Section 11.4.2 described a procedure for obtaining confidence intervals with specified precision. Confidence intervals for the *difference* between two systems' performance can be obtained in an analogous manner.

Suppose that we want the error in our estimate of $\theta_1 - \theta_2$ to be less than $\pm\epsilon$ (the quantity ϵ might be a practically significant difference). Therefore, our goal is to find a number of replications R such that

$$H = t_{\alpha/2,\nu}\text{s.e.}(\bar{Y}_1 - \bar{Y}_2) \leq \epsilon \tag{12.19}$$

As in Section 11.4.2, we begin by making $R_0 \geq 2$ replications of each system to obtain an initial estimate of s.e.$(\bar{Y}_1 - \bar{Y}_2)$. We then solve for the total number of replications $R \geq R_0$ needed to achieve the half-length criterion (12.19). Finally, we make an additional $R - R_0$ replications (or a fresh R replications) of each system, compute the confidence interval, and check that the half-length criterion has been attained.

Example 12.1: Continued

Recall that $R_0 = 10$ replications and complete synchronization of the random numbers yielded the 95% confidence interval for the difference in expected response time of the two vehicle-inspection stations in Inequality (12.18); this interval can be rewritten as 0.4 ± 0.90 minutes. Although system 2 appears to have the smaller expected response time, the difference is not statistically significant, since the confidence interval contains 0. Suppose that a difference larger than ±0.5 minute is considered to be practically significant. We therefore want to make enough replications to obtain a $H \leq \epsilon = 0.5$.

The confidence interval used in Example 12.1 was $\bar{D} \pm t_{\alpha/2,R_0-1} S_D / \sqrt{R_0}$, with the specific values $\bar{D} = 0.4$, $R_0 = 10$, $t_{0.025,9} = 2.26$ and $S_D^2 = 1.7$. To obtain the desired precision, we need to find R such that

$$\frac{t_{\alpha/2,R-1} S_D}{\sqrt{R}} \leq \epsilon$$

Therefore, R is the smallest integer satisfying $R \geq R_0$ and

$$R \geq \left(\frac{t_{\alpha/2,R-1} S_D}{\epsilon} \right)^2$$

Since $t_{\alpha/2,R-1} \leq t_{\alpha/2,R_0-1}$, a conservative estimate for R is given by

$$R \geq \left(\frac{t_{\alpha/2,R_0-1} S_D}{\epsilon} \right)^2$$

Substituting $t_{0.025,9} = 2.26$ and $S_D^2 = 1.7$, we obtain

$$R \geq \frac{(2.26)^2(1.7)}{(0.5)^2} = 34.73$$

implying that 35 replications are needed, 25 more than in the initial experiment.

12.2 COMPARISON OF SEVERAL SYSTEM DESIGNS

Suppose that a simulation analyst desires to compare K alternative system designs. The comparison will be made on the basis of some specified performance measure, θ_i, of system i, for $i = 1, 2, \ldots, K$. Many different statistical procedures have been developed that can be used to analyze simulation data and draw statistically sound inferences concerning the parameters θ_i. These procedures can be classified as being either fixed-sample-size procedures or sequential-sampling (or *multistage*) procedures. In the first type, a predetermined sample size (i.e., run length and number of replications) is used to draw inferences via hypothesis tests or confidence intervals. Examples of fixed-sample-size procedures include the interval estimation of a mean performance measure (Section 11.3) and the interval estimation of the difference between mean performance measures of two systems [as by Expression (12.1) in Section 12.1]. Advantages of fixed-sample-size procedures include a known or easily estimated cost in terms of computer time before running the experiments. When computer time is limited, or when a pilot study is being conducted, a fixed-sample-size procedure might be appropriate. In some cases, clearly inferior system designs may be ruled out at this early stage. A major disadvantage is that a strong conclusion could be impossible. For example, the confidence interval could be too wide for practical use, since the width is an indication of the precision of the point estimator. A hypothesis test may lead to a failure to reject the null hypothesis, a weak conclusion in general, meaning that there is no strong evidence one way or the other about the truth or falsity of the null hypothesis.

A sequential sampling scheme is one in which more and more data are collected until an estimator with a prespecified precision is achieved or until one of several alternative hypotheses is selected, with the probability of correct selection being larger than a prespecified value. A two-stage (or multistage) procedure is one in which an initial sample is used to estimate how many additional observations are needed to draw conclusions with a specified precision. An example of a two-stage procedure for estimating the performance measure of a single system was given in Section 11.4.2 and 12.1.4.

The proper procedure to use depends on the goal of the simulation analyst. Some possible goals are the following:

1. estimation of each parameter, θ_i;
2. comparison of each performance measure, θ_i, to a control, θ_1 (where θ_1 could represent the mean performance of an existing system);
3. all pairwise comparisons, $\theta_i - \theta_j$, for $i \neq j$;
4. selection of the best θ_i (largest or smallest).

The first three goals will be achieved by the construction of confidence intervals. The number of such confidence intervals is $C = K$, $C = K - 1$, and $C = K(K - 1)/2$, respectively. Hochberg and Tamhane [1987] and Hsu [1996] are comprehensive references for such multiple-comparison procedures. The fourth goal requires the use of a type of statistical procedure known as a multiple ranking and selection procedure. Procedures to achieve these and other goals are discussed by Kleijnen [1975, Chapters II and V], who also discusses their relative merit and disadvantages. Goldsman and Nelson [1998] and Law and Kelton [2000]

discuss those selection procedures most relevant to simulation. A comprehensive reference is Bechhofer, Santner, and Goldsman [1995]. The next subsection presents a fixed-sample-size procedure that can be used to meet goals 1, 2, and 3 and is applicable in a wide range of circumstances. Subsections 12.2.2–12.2.3 present related procedures to achieve goal 4.

12.2.1 Bonferroni Approach to Multiple Comparisons

Suppose that C confidence intervals are computed and that the ith interval has confidence coefficient $1 - \alpha_i$. Let $\mathcal{S}_i$ be the statement that the ith confidence interval contains the parameter (or difference of two parameters) being estimated. This statement might be true or false for a given set of data, but the procedure leading to the interval is designed so that statement $\mathcal{S}_i$ will be true with probability $1 - \alpha_i$. When it is desired to make statements about several parameters simultaneously, as in goals 1, 2 and 3, the analyst would like to have high confidence that *all* statements are true simultaneously. The Bonferroni inequality states that

$$P(\text{all statements } \mathcal{S}_i \text{ are true}, i = 1, \ldots, C) \geq 1 - \sum_{j=1}^{C} \alpha_j = 1 - \alpha_E \tag{12.20}$$

where $\alpha_E = \sum_{j=1}^{C} \alpha_j$ is called the overall error probability. Expression (12.20) can be restated as

$$P(\text{one or more statements } \mathcal{S}_i \text{ is false}, i = 1, \ldots C) \leq \alpha_E$$

or equivalently,

$$P(\text{one or more of the } C \text{ confidence intervals does not contain the parameter being estimated}) \leq \alpha_E$$

Thus, α_E provides an upper bound on the probability of a false conclusion. To conduct an experiment that involves making C comparisons, first select the overall error probability, say $\alpha_E = 0.05$ or 0.10. The individual α_j may be chosen to be equal ($\alpha_j = \alpha_E/C$), or unequal, as desired. The smaller the value of α_j, the wider the jth confidence interval will be. For example, if two 95% c.i.'s ($\alpha_1 = \alpha_2 = 0.05$) are constructed, the overall confidence level will be 90% or greater ($\alpha_E = \alpha_1 + \alpha_2 = 0.10$). If ten 95% c.i.'s are constructed (α_i 0.05, $i = 1, \ldots, 10$), the resulting overall confidence level could be as low as 50% ($\alpha_E = \sum_{i=1}^{10} \alpha_i = 0.50$), which is far too low for practical use. To guarantee an overall confidence level of 95%, when 10 comparisons are being made, one approach is to construct ten 99.5% confidence intervals for the parameters (or differences) of interest.

The Bonferroni approach to multiple confidence intervals is based on expression (12.20). A major advantage is that it holds whether the models for the alternative designs are run with independent sampling or with common random numbers.

The major disadvantage of the Bonferroni approach in making a large number of comparisons is the increased width of each individual interval. For example, for a given set of data and a large sample size, a 99.5% c.i. will be $z_{0.0025}/z_{0.025} = 2.807/1.96 = 1.43$ times longer than a 95% c.i. For small sample sizes—say, for a sample of size 5—a 99.5% c.i. will be $t_{0.0025,4}/t_{0.025,4} = 5.598/2.776 = 1.99$ times longer than an individual 95% c.i. The width of a c.i. is a measure of the precision of the estimate. For these reasons, it is recommended that the Bonferroni approach be used only when a small number of comparisons are being made. Twenty or so comparisons appears to be the practical upper limit.

Corresponding to goals 1, 2, and 3, there are at least three possible ways of using the Bonferroni Inequality (12.20) when comparing K alternative system designs:

1. (*Individual c.i.'s*): Construct a $100(1 - \alpha_i)\%$ c.i. for parameter θ_i by using Expression (11.12), in which case the number of intervals is $C = K$. If independent sampling were used, the K c.i.'s would be

mutually independent, and thus the overall confidence level would be $(1-\alpha_1)\times(1-\alpha_2)\times\cdots\times(1-\alpha_C)$, which is larger (but not much larger) than the right side of Expression (12.20). This type of procedure is most often used to estimate multiple parameters of a single system, rather than to compare systems—and, because multiple parameter estimates from the same system are likely to be dependent, the Bonferroni inequality typically is needed.

2. (*Comparison to an existing system*): Compare all designs to one specific design—usually, to an existing system: that is, construct a $100(1-\alpha_i)\%$ c.i. for $\theta_i - \theta_1 (i = 2, 3, \ldots, K)$, using Expression (12.1). (System 1 with performance measure θ_1 is assumed to be the existing system). In this case, the number of intervals is $C = K - 1$. This type of procedure is most often used to compare several competitors to the present system in order to learn which are better.
3. (*All pairwise comparisons*): Compare all designs to each other—that is, for any two system designs $i \neq j$, construct a $100(1-\alpha_{ij})\%$ c.i. for $\theta_i - \theta_j$. With K designs, the number of confidence intervals computed is $C = K(K-1)/2$. The overall confidence coefficient would be bounded below by $1-\alpha_E = 1-\sum\sum_{i\neq j}\alpha_{ij}$ (which follows by Expression (12.20)). It is generally believed that CRN will make the true overall confidence level larger than the right side of Expression (12.20), and usually larger than will independent sampling. The right side of Expression (12.20) can be thought of as giving the worst case (i.e., the lowest possible overall confidence level).

Example 12.4

Reconsider the vehicle-inspection station of Example 12.1. Suppose that the construction of additional space to hold one waiting car is being considered. The alternative system designs are the following:

1. existing system (parallel stations);
2. no space between stations in series;
3. one space between brake and headlight inspection only;
4. one space between headlight and steering inspection only.

Design 2 was compared to the existing setup in Example 12.1. Designs 2, 3, and 4 are series queues, as shown in Figure 12.1(b), the only difference being the number or location of a waiting space between two successive inspections. The arrival process and the inspection times are as given in Example 12.1. The basis for comparison will be mean response time, θ_i, for system i, where a response time is the total time it takes for a car to get through the system. Confidence intervals for $\theta_2 - \theta_1$, $\theta_3 - \theta_1$, and $\theta_4 - \theta_1$ will be constructed, each having an overall confidence level of 95%. The run length T_E has now been set at 40 hours (instead of the 16 hours used in Example 12.1), and the number of replications R of each model is 10. Common random numbers will be used in all models, but this does not affect the overall confidence level, because, as mentioned, the Bonferroni Inequality (12.20) holds regardless of the statistical independence or dependence of the data.

Since the overall error probability is $\alpha_E = 0.05$ and $C = 3$ confidence intervals are to be constructed, let $\alpha_i = 0.05/3 = 0.0167$ for $i = 2, 3, 4$. Then use Expression (12.1) (with proper modifications) to construct $C = 3$ confidence intervals with $\alpha = \alpha_i = 0.0167$ and degrees of freedom $\nu = 10 - 1 = 9$. The standard error is computed by Equation (12.14), because common random numbers are being used. The output data Y_{ri} are displayed in Table 12.5; Y_{ri} is the sample mean response time for replication r on system i ($r = 1, \ldots, 10$; $i = 1, 2, 3, 4$). The differences $D_{ri} = Y_{r1} - Y_{ri}$ are also shown, together with the sample mean differences, $\bar{D}_i$, averaged over all replications as in Equation (12.12), the sample variances $S_{D_i}^2$, and the standard error. By Expression (12.1), the three confidence intervals, with overall confidence coefficient at least $1 - \alpha_E$, are given by

$$\bar{D}_i - t_{\alpha_i/2,R-1}\,\text{s.e.}(\bar{D}_i) \leq \theta_1 - \theta_i \leq \bar{D}_i + t_{\alpha_i/2,R-1}\,\text{s.e.}(\bar{D}_i), \quad i = 2, 3, 4$$

Table 12.5 Analysis of Output Data for Vehicle Inspection System When Using CRN

	Average Response Time for System Design				*Observed Difference with System Design 1*		
Replication,	*1,*	*2,*	*3,*	*4.*			
r	Y_{r1}	Y_{r2}	Y_{r3}	Y_{r4}	D_{r2}	D_{r3}	D_{r4}
1	63.72	63.06	57.74	62.63	0.66	5.98	1.09
2	32.24	31.78	29.65	31.56	0.46	2.59	0.68
3	40.28	40.32	36.52	39.87	−0.04	3.76	0.41
4	36.94	37.71	35.71	37.35	−0.77	1.23	−0.41
5	36.29	36.79	33.81	36.65	−0.50	2.48	−0.36
6	56.94	57.93	51.54	57.15	−0.99	5.40	−0.21
7	34.10	33.39	31.39	33.30	0.71	2.71	0.80
8	63.36	62.92	57.24	62.21	0.44	6.12	1.15
9	49.29	47.67	42.63	47.46	1.62	6.66	1.83
10	87.20	80.79	67.27	79.60	6.41	19.93	7.60
Sample mean, $\bar{D}_i$					0.80	5.686	1.258
Sample standard deviation, S_{D_i}					2.12	5.338	2.340
Sample variance, $S_{D_i}^2$					4.498	28.498	5.489
Standard error, $S_{D_i}/\sqrt{R}$					0.671	1.688	0.741

The value of $t_{\alpha_1/2,R-1} = t_{0.0083,9} = 2.97$ is obtained from Table A.5 by interpolation. For these data, with 95% confidence, it is stated that

$$-1.19 \le \theta_1 - \theta_2 \le 2.79$$
$$0.67 \le \theta_1 - \theta_3 \le 10.71$$
$$-0.94 \le \theta_1 - \theta_4 \le 3.46$$

The simulation analyst has high confidence (at least 95%) that all three confidence statements are correct. Notice that the c.i. for $\theta_1 - \theta_2$ again contains zero; thus, there is no statistically significant difference between design 1 and design 2, a conclusion that supports the previous results in Example 12.1. The c.i. for $\theta_1 - \theta_3$ lies completely above zero and so provides strong evidence that $\theta_1 - \theta_3 > 0$—that is, that design 3 is better than design 1 because its mean response time is smaller. The c.i. for $\theta_1 - \theta_4$ contains zero, so there is no statistically significant difference between designs 1 and 4.

If the simulation analyst now decides that it would be desirable to compare designs 3 and 4, more simulation runs would be necessary, because it is not formally correct to decide which confidence intervals to compute after the data have been examined. On the other hand, if the simulation analyst had decided to compute all possible confidence intervals (and had made this decision before collecting the data, Y_{ri}), the number of confidence intervals would have been $C = 6$ and the three c.i.'s would have been $t_{0.0042,9}/t_{0.0083,9} \approx 3.32/2.97 = 1.12$ times (or 12%) longer. There is always a trade-off between the number of intervals (C) and the width of each interval. The simulation analyst should carefully consider the possible conclusions before running the simulation experiments and choose those runs and analyses that will provide the most useful information. In particular, the number of confidence intervals computed should be as small as possible—preferably, 20 or less.