Proceedings of the 1986 Winter Simulation Conference
J. Wilson, J. Henriksen, S. Roberts (eds.)

STATISTICAL DESIGN AND ANALYSIS

W. David Kelton
Department of Management Sciences
School of Management
University of Minnesota
Minneapolis, Minnesota 55455

ABSTRACT

After a general simulation model is built, coded, verified, and validated, it is used to learn about the system(s) under study, requiring careful prior design of the runs and appropriate analysis of their output. This paper surveys methods for effective and efficient design and analysis of simulation experiments.

1. INTRODUCTION

Most computer simulations may be classified as being either <u>deterministic</u> or <u>stochastic</u>.

In a deterministic simulation, there are no random components or inputs; an example would be a differential or difference equation model with no random components that is being simulated (as opposed to being mathematically solved) due to its complexity. Deterministic simulations are simpler from the analysis point of view since a single run of the simulation, under a fixed set of operating and parametric conditions, will always produce the same output. This mode of study is akin to "traditional" computer work where repeating a run is redundant.

On the other hand, stochastic simulations involve as part of their input the generation of random numbers which in turn propagate through to the output, making it random as well. Thus, stochastic simulations will, in general, not produce the same output values if run repeatedly with independent random number streams (called <u>replicating</u>). For this reason, it is important to realize that it is generally <u>not</u> sufficient to go the "usual" programming route of debugging the program, running it once, and accepting the resulting output as "the answer." The purpose of this paper is to provide an introduction and overview of the statistical methods that can be brought to bear in stochastic simulation.

As a simple example which will be carried throughout, consider a single-server queueing system with exponential interarrival times and service times with means $1/\lambda = 1.0$ minute and $1/\mu = 0.5$ minute, respectively. (This is usually called the <u>M/M/1 queue</u>.) The system begins empty and idle, and runs until the 1000th customer has completed his delay in queue (exclusive of service time). The output measure of performance is the average delay in queue,

$$\bar{D}(1000) = \sum_{i=1}^{1000} D_i \,/\, 1000$$

Where D_i represents the delay in queue of the ith individual customer.

This model was simulated, and resulted in a value of $\bar{D}(1000) = 0.501$ minutes. Since this particular result depended on the particular random numbers generated for this simulation (which are determined by the generator and seed values specified), we might have obtained a different result with different (but just as valid) random numbers. Indeed, Table 1 shows the results of ten independent replications of this model, using separate random number streams for each replication. As can be seen, there is considerable variability from replication to replication, and had we stopped with just a single run of the system (and the average delay of 0.501), we would have been somewhat mislead about the system's expected performance. Even worse, we might have obtained (with other random number generator seeds) a value from the table such as 0.378 or 0.544, which would be quite misleading. The danger is that we would have no way of knowing this unless we made multiple replications.

Table 1: M/M/1 Queue Replications

Replication	$\bar{D}(1000)$
1	0.501
2	0.449
3	0.540
4	0.460
5	0.406
6	0.485
7	0.456
8	0.378
9	0.408
10	0.544

For simulations such as this one, which involve specific starting and stopping conditions as part of the model, the above approach of replicating the whole model independently is an appropriate technique for the basis of a valid statistical analysis. For other types of simulations, however, replication may not be a fruitful tack. The formalization of what to do with the replication results, as well as possible approaches to steady-state simulations' statistical analysis, is taken up below in Section 3.

The remainder of the paper is divided into two main sections, along the lines of statistical aspects concerning the input (Section 2) to simulations and those involving the output (Section 3). Section 4 contains some general conclusions. The purpose of this paper is to draw attention to these kinds of problems in simulation, rather than to provide a complete compendium of problems, goals, and methods. On the input side, Kelton (1984) contains additional information and references. The reader is referred to Law (1983) and Welch (1983) for comprehensive treatments of the statistical aspects of output data analysis, and for many references to the simulation methodological literature; additional classifications of types of simulations, as well as further examples, can be found in Kelton (1983, 1985).

2. INPUT -- EXPERIMENTAL DESIGN

In this section we describe some of the statistical concerns that enter into the simulation process on the input side. In more traditional laboratory-type experimentation, these concerns would be covered by the term experimental design, used in the traditional sense. In simulation experiments, however, more general kinds of "design" must be done; this is discussed in Section 2.1.

Although the focus of this paper is on stochastic simulation, it is interesting to note that many experimental design methods can be used as well in deterministic simulations. If there are many parameters in a deterministic simulation model whose effects are of interest then a screening design (see Section 2.4) could be used to try to identify the important ones. The use of experimental design in deterministic simulations is, in a way, easier than in stochastic simulations, since we don't have to be concerned about noise, replication, significance of effects estimates, etc.

The remainder of the section discusses the use of traditional experimental design in simulation.

2.1 Simulation Experimental Design

There are several issues in designing statistically valid and effective simulation issues that do not come up in laboratory experiments. Some of these are discussed below.

Run Conditions. A simulation must be started and stopped somehow, and the decisions on how these tasks are done can have a dramatic effect on the simulation's results, and thus on the ensuing decisions. In our example, we chose to start the system out empty and idle and terminate it after 1000 customers had completed their delays in queue. The choice of these conditions is really a modeling issue, in the sense that different start/stop rules really define a different model. More on starting and stopping simulations will appear in Section 3 in connection with the difference between terminating and steady-state simulations.

What to Watch. In our example simulation, the output variable of interest was the average delay in queue of the 1000 customers. Most real simulations consume large amounts of computer time (not to mention modeling and coding time on the part of the analysts), so we would usually want to get more than just a single number out of a simulation run. The decision about "what to watch" as the simulation proceeds must be made before the simulation is run (unless we are saving a complete "trace" of the simulation, which typically is a very large file if the simulation is complicated or long). Returning to the example, we could also have observed the maximum of the 1000 delays, or the time-average number in queue, given by

$$\bar{Q}(T(1000)) = \int_0^{T(1000)} Q(t)\,dt \;/\; T(1000),$$

where $Q(t)$ is the number of customers in queue at time t, and $T(1000)$ is the time (on the simulation clock) required to observe the desired 1000 delays; other output measures of interest would be the maximum value of $Q(t)$, the utilization of the server (i.e., proportion of time busy), and the value of $T(1000)$ itself. In addition, we often would like to know the value of a quantile, a value that "cuts off" a certain proportion of observed values. For example, it could be of interest in our example to determine a value above which it is only 5% probable that the queue will ever grow; this would be the upper 95% quantile of the distribution of the maximum value of $Q(t)$.

Random Number Allocation. In some simulations, it is possible to reduce the variability of the output by exploiting the ability to control the random number generator. Such variance reduction techniques (the subject of a subsequent tutorial in this conference, by Professor Russell Cheng) usually require devoting particular random numbers from the generator to particular purposes, and this requires design, often of the simulation code itself, beforehand. In particular, one must decide whether to use the methods of common random numbers or antithetic variates in designing the simulation code as well as the runs. See Schruben and Margolin (1978) for more on random number allocation.

Identifying Factors. In laboratory experiments, attention is usually restricted to varying the factors that would be controllable in the real world. Other uncontrollable factors are usually not considered, simply because the experiment can't control them either. In simulation experiments, however, all factors are "controllable" by simply changing the input parameters. In our example, the mean interarrival time $1/\lambda$ would not usually be controllable in the real world (or in a physical experiment with the corresponding real-world system), but there is no reason why we couldn't vary it in the simulation. This would provide a means for investigating hypothetical "what if" questions that could not be investigated in a physical experiment. Thus, we should be more broad-minded about what constitutes a "factor" in simulation experiments.

Determining Factor Combinations. Before doing the simulation, we must decide at what levels to set the factors, and in what combinations. This is where "traditional" experimental design can help, and is the subject of the rest of this section. A comprehensive treatment of experimental design in general is Box, Hunter, and Hunter (1978).

2.2 Factorial Designs

Once the run conditions, output measures, random number allocation scheme, and factors to be studied have been determined, it remains to decide exactly which variants (in terms of particular combinations of factors) will be considered.

For each factor, we must decide on how many different values it will take on; these are called levels of the factor in traditional experimental design parlance. For numerical factors (such as the arrival rate $1/\lambda$), there may be a theoretically unlimited number of values possible; for other non-numerical factors (such as the queue discipline or whether there is a capacity on the queue length) there may be only a few that physically make sense. In any case, we must decide how many levels each factor will assume in the experimentation, and what those levels will be (called the coding of the factors). These issues will be discussed in Sections 2.3 and 2.4 below.

Just as we must decide what to watch in the simulation model itself, we should also ask what kind of information we want from the experimental design. Typically, a good design will provide information on the effect of each of the factors alone (called the main effects of the factors), as well as possible joint action between different factors (called interactions between the factors). Depending on the "richness" of the design (basically a function of how many simulation runs we are willing to make), varying amounts of such information can be obtained from the design. A more ambitious goal would be to learn something about the functional dependence of the simulation output measure(s) on the factors; this is called a response surface, and usually requires a reasonably large number of simulation runs.

Finally, all of the design information discussed in the preceding paragraph (main effects, interactions, response surfaces) are possible for each of the simulation outputs (or responses). For example, the effects of the arrival rate and the service rate on both the average delay in queue and the maximum length of the queue could be studied.

2.3 2^k Factorial Designs

A useful and economical starting point for designing simulation experiments is to allow each factor to assume only two levels. What these levels are is sometimes a murky issue; typical advice is to use prior information about the system to set the levels at what is felt to be "high" and "low" values for the factor.

Although it may seem limiting to allow only two levels per factor, the reason for this is economy. If there are k different factors, each of which can assume two levels, then the number of factor combinations is 2^k (whence the name of this technique). For example, if there are five factors (a modest number for a complex simulation), then there are 32 different combinations, each corresponding to a different simulation model that must be run (and perhaps replicated). If we instead wanted to study each factor at four levels, then the number of combinations would instead be $4^5 = 1024$. Thus, limiting to two the number of levels each factor can assume allows a higher number of factors to be considered, which is usually desirable, at least in the beginning stages of simulation experimentation.

Returning to our example, suppose we want to study the effects of three factors: The mean interarrival time $1/\lambda$, the mean service time $1/\mu$, and the number of customers m whose delays are to be observed (originally, we took m = 1000). For each factor, we "code" the values using a "+" or "-" sign; Table 2 gives the coding for the three factors.

Table 2: Factor Coding for M/M/1 Queue

		Factor		
		$1/\lambda$	$1/\mu$	m
Level	-	1.0	0.5	100
	+	1.1	0.8	1000

Notice that the (-,-,+) factor combination is the one originally used. Now, we would like to learn about what happens if the mean interarrival time is increased to 1.1, the mean service time is increased to 0.8, and the length of the simulation is decreased to 100 customer delays.

The first four columns of Table 3 spell out exactly what each of the factor levels will be for each of the $2^3 = 8$ factor combinations, or runs of the experiment. This is called the design matrix for the experiment, and is important for setting up the simulations as well as for the later analysis of the design's results.

Table 3: Design Matrix and Results for M/M/1 Queue

Run	Factor 1/λ	Factor 1/μ	Factor m	Response $\bar{D}(m)$
1	-	-	-	0.681
2	+	-	-	0.805
3	-	+	-	0.674
4	+	+	-	1.256
5	-	-	+	0.449
6	+	-	+	0.574
7	-	+	+	2.463
8	+	+	+	2.091

The final column in Table 3 gives the results of the average delay in queue of the m customers for each factor combination. (Note that run 5 -- the original configuration -- has a response of 0.449, different from the 0.501 originally obtained, again displaying the randomness of output from stochastic simulations.)

To study the effect of changing a factor from its "-" level to its "+" level, we simply look at the average response when that factor is at its "+" level, minus the average response when the factor is at its "-" level. For factor 1 ($1/\lambda$), this is

$$(0.805 + 1.256 + 0.574 + 2.091)/4 - (0.681 + 0.674 + 0.449 + 2.463)/4,$$

which is 0.115. The interpretation is that the average, or main effect of changing the mean interarrival time from 1.0 minute to 1.1 minutes was to increase the average delay in queue (of m customers) by 0.115 minute. Similarly, the main effect of changing the mean service time (from 0.5 minute to 0.8 minute) was to increase average delays by 0.994 minute, and the main effect of changing the number of customers from 100 to 1000 was to increase average delays by 0.540 minute.

It is also possible from designs of this type to study how two (or more) factors interact with each other; for this we refer the reader to Box, Hunter, and Hunter (1978).

This example, in addition to demonstrating how factorial designs can be applied in simulation, illustrates two important shortcomings as well. First, the results obtained are in general dependent on the particular levels we chose for the factor coding (except in the unlikely situation that the output response is really a linear function of the factors). The second (and probably more serious, at least in stochastic simulation) defect is that the above analysis was based on a single replication of each model variant. (Does it even make intuitive physical sense that increasing the mean interarrival time from 1.0 to 1.1 -- thereby making arrivals further apart -- should make the queue more congested, as evidenced by the positive main effect estimate of 0.115?) A better tack would be to replicate the entire experiment some number (say, n) times, and obtain n independent estimates of each main effect, interaction, etc. These could then be averaged across replications of the design to obtain much more stable effects estimates.

2.4 Screening Designs

Many simulations involve a large number of parameters, operational rules, and structural features that are potentially interesting factors in a course of simulation experimentation. The 2^k factorial designs just discussed may not be practical for such situations. For example, a simulation with ten factors (not an unreasonably sized model) would require $2^{10} = 1024$ separate factor combinations; if the design were replicated five times (a modest amount of data collection from the statistician's point of view), then we would have to make some 5120 separate simulation runs.

This being clearly unworkable for large simulations, a single run of which may be quite expensive, we must either reduce the number of factors (which may be undesirable), or turn to a different type of experimental design. Fractional factorial designs have been developed that call for only a fraction (half, fourth, eighth, sixteenth, ...) of the 2^k runs in the "full" factorial designs discussed in Section 2.3; see Box, Hunter, and Hunter (1978). Basically, in carrying out a fractional rather than a full design, we give up the ability to estimate higher-order interactions (which may not be of much interest anyway) and receive in return a more modest requirement in terms of the number of runs. This is particularly useful for "screening out" factors which appear to have little or no effect on the responses, reducing the number of important factors for more intensive study later. Further types of screening designs, specifically in the simulation context, are discussed by Smith and Mauro (1982).

3. OUTPUT -- STATISTICAL ANALYSIS

A carefully designed course of simulation experimentation sets the stage for an appropriate and effective analysis of the output from these simulations. Many of the ideas of "standard" statistics can be used in the analysis of simulation output data, but care must be taken to apply them appropriately. Failure to do so can result in serious errors of interpretation, rendering the simulation effort not just useless, but actually harmful. As was demonstrated in Table 1, the output from stochastic simulations is variable, or random, and thus requires some sort of statistical analysis for its proper use and interpretation. Also, the experimental design example from Section 2.3 illustrated the questionability of making a decision on the basis of single runs of systems. (The problem in that example is that increasing the time between successive arrivals appeared to lead to an increase in congestion -- as measured by average delay in queue -- rather than the more likely decrease.)

In this section we will mention the main threads of problems, goals, and solution approaches, without attempting to provide a complete compendium of the (by now) extensive literature on the statistical analysis of simulation output data; again, the reader is referred to Kelton (1983, 1985) for previous presentations in prior years' Winter Simulation Conferences, and to Law (1983), and Welch (1983) for more complete surveys of the subject.

3.1 Time Frames of Simulations

Traditionally, dynamic simulations have been classified as being either terminating or steady-state.

In a terminating simulation, the model is defined relative to specific starting and stopping conditions, and the output responses are thus also defined relative to these beginning and ending rules. In our M/M/1 queue example, we said that the starting conditions were that the system was empty and that the server was idle, and the terminating conditions were that 1000 customers had completed their delays in queue. As we saw in Section 2.3, changing one of these (specifically, altering the 1000 to 100) led to a change in model behavior, and really constituted a different model (parametrically if not structurally). Establishing appropriate start/stop conditions for a simulation model greatly simplifies the statistical analysis problem (see Sections 3.2 and 3.3 below), but is by no means obvious in many applications. This is really a modeling issue, and should be made, so far as possible, in accordance with the way the corresponding real-world system actually is thought to start and stop.

On the other hand, a steady-state simulation has as its goal the estimation of long-run (or steady-state) system performance measures. A steady-state simulation of the M/M/1 queue might seek to estimate the long-run expected average delay in queue, or the steady-state expected proportion of time the server is busy. Notice that no initial conditions or terminal conditions for the simulation are specified, which creates a real difficulty for the simulator, since such obviously must be specified in order to run the simulation. In particular, the performance measures are specified in a way that is independent of the initial conditions used in the simulations, and over a theoretically infinite period of time. The reality is, however, that in a simulation we must both start and stop, and it does matter how these tasks are done. These problems indicate that steady-state simulations are much more difficult and costly to carry out than are terminating simulations, and the statistical analysis problem is also much more difficult (see Section 3.3). It is probably good practical advice to avoid doing a steady-state simulation if possible, but there are situations in which a steady-state simulation is really the appropriate goal (for example, in telecommunications systems).

Some simulations may not fit into either the terminating or steady-state mold. For example, we might alter the arrival process in our example M/M/1 model to allow for peaks and troughs of arrival rates, perhaps in a periodic or cyclical way. In this type of model, it could conceivably be of interest to carry out a terminating-type simulation defined across some number of complete cycles of the arrival pattern. We might also want to do a steady-state-type analysis in which system performance is measured over a large number (theoretically infinite) number of arrival cycles. In the latter case, care should be taken that it makes sense to think of some sort of steady state's being reached; for example, in a model with an upward trend in the arrival rate cycle, it would probably be expected that congestion would grow without bound, rendering a steady-state simulation meaningless.

3.2 Statistical Methods for Terminating Simulations

We have already seen the basic methodology for providing information for appropriate statistical analysis of terminating simulations -- replication of the entire simulation. Perhaps the most important idea here is that an entire simulation run provides only a sample of size one. To be sure, a single replications produces many numbers (e.g., 1000 separate delays in queue), but these numbers go together in concert to determine the final output measure of performance. In order to obtain additional data for the purpose of statistical analysis, it is necessary to replicate the entire simulation some number of times.

In general, let X_j represent the output measure from the j^{th} of a total of n replications of the model, and let

$$\bar{X}(n) = \sum_{j=1}^{n} X_j / n$$

and

$$s^2(n) = \sum_{j=1}^{n} (X_j - X(n))^2 / (n - 1)$$

be the sample mean and variance of the replication data. These statistics provide the basis of several usual statistical descriptors, such as a hypothesis test for the expected value of the output measure, $E(X_j)$, or a confidence interval for $E(X_j)$. The latter is particularly useful, and can be formed (at confidence level $1 - \alpha$) as

$$\bar{X}(n) \pm t_{n-1,1-\alpha/2}\, s(n)/\sqrt{n},$$

where $t_{n-1,1-\alpha/2}$ is the upper $1-\alpha/2$ critical point of Student's t distribution with n-1 degrees of freedom. The validity of this interval depends on the assumption that the X_j's have a normal probability distribution, a fairly reasonable assumption in most (but certainly not all) cases. Carrying this computation out for the n = 10 replications from Table 1 (where $X_1 = 0.501$, $X_2 = 0.449$, etc.) we get a 95% (i.e., $\alpha = 0.05$) confidence interval of 0.463 ± 0.040, which provides an easily interpreted statement of how accurately we feel we have estimated the expected average delay in queue of the first 1000 customers to this system.

There are several drawbacks to this methodology. First, the normality assumption may not be reasonable, rendering the above confidence interval invalid; in this case, a nonparametric method might be considered. Secondly, the interval in a given case might be too wide to be of any practical use; more replications (i.e., higher n) can remedy this, but only at the obvious cost in computing. Finally, it may be difficult to choose a confidence level appropriate for the application; higher confidence is certainly desirable, but this would have the undesirable side-effect of increasing the interval width.

The replication approach provides the basis for appropriate statistical analysis (of many types, in addition to confidence intervals), and should always be done in a terminating simulation. Law (1980) considers additional statistical aspects of terminating simulations.

3.3 Statistical Methods for Steady-State Simulations

Finding an appropriate, reliable, and efficient method for statistical analysis of steady-state simulations is much more difficult, and has been the subject of a substantial amount of recent research; see Law (1983). We will only briefly indicate here some of the possibilities.

Replication could be used for a steady-state simulation as well as for a terminating simulation, but one must take care to start and stop the simulation so that steady-state conditions are observed. Naturally, this involves long simulation runs, but it is usually difficult to decide just how long is long enough. Furthermore, some attempt should be made to initialize the simulation in a way that is thought to be reasonably representative of steady state conditions. While it is easy to say all this, it is often difficult to carry it out for a complex model. If replication is used in steady-state simulations, however, it is very important that attention be given to these matters to avoid biasing the simulation results; this difficulty is known as the initial transient or startup problem in the simulation literature.

To get around the initial transient problem, several methods have been developed which call for only a single "long" run, so that the transient period will have to be passed through only once. This is really a single "replication," and we thus lose the independent observations we got from multiple replications. The difficulty comes in obtaining an estimate of the variance of the point estimator, and several methods have been proposed. Batch means breaks the single output record into some number of "batches" of contiguous points, whose averages are then taken to be independent; they aren't really independent, and so one usually takes the batches to be quite large (hundreds or thousands of individual points) in an attempt to reduce the correlation between the batches. Time series modeling fits a relatively simple time series model (e.g., ARMA) to the output record and uses the estimates of the parameters to obtain a variance estimate. Spectral analysis is a way of estimating the autocorrelation in the output record to estimate the desired variance. The regenerative method makes a special probabilistic assumption about the process being simulated, and uses this to obtain a more rigorously justified confidence interval. Standardized time series makes a weaker assumption about the process, and also yields a rigorous method for statistical analysis that is in addition general and simple to apply.

No method for steady-state statistical analysis is really cheap and simple. Probably the best one can do in practice is to become familiar with the available methods, and attempt to use an appropriate technique for a given problem.

3.4 Additional Statistical Problems, Goals, and Methods

The above discussion focuses on the estimation of a single output measure from a single simulation model. Usually, our goals are quite a bit more ambitious than this, including simultaneous estimation of several output measures from the same model, comparison of several alternative model designs, selection of one or several "best" designs from among the alternatives, and optimization of design by means of simulation. These subjects will be taken up in a subsequent tutorial at the conference, by Professor Peter Glynn.

There also exists in simulation the opportunity to "control" the randomness, due to the controllable nature of computer-generated random numbers. We can take advantage of this ability to reduce the instability, or variance, of the simulation output. The result is that we get more accurate estimates for a given computing budget, or are required to do less computing to attain a desired degree of accuracy. Such variance reduction techniques are the subject of another subsequent tutorial at the conference, by Professor Russell Cheng.

4. CONCLUSIONS

The purpose here has been to call attention to the fact that statistical analysis has an important role to play in a simulation project. There has been an understandable tendency to ignore these aspects, since they are to some extent "outside" the modeling and coding effort, which typically consumes substantial resources. However, unless some attempt is made to recognize and estimate the uncertainty in a stochastic simulation's output, the very real possibility exists that all the work that went into modeling and coding will be wasted.

REFERENCES

Box, G.E.P., Hunter, W.G., and Hunter, J.S. (1978). Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building. Wiley, New York.

Kelton, W.D. (1983). Simulation analysis. In: Proceedings of the 1983 Winter Simulation Conference (S. Roberts, J. Banks, and B. Schmeiser, eds.). Institute of Electrical and Electronics Engineers, San Francisco, California, 159-168.

Kelton, W.D. (1984). Input data collection and analysis. In: Proceedings of the 1984 Winter Simulation Conference (S. Sheppard, U. Pooch, and D. Pegden, eds.). Institute of Electrical and Electronics Engineers, San Francisco, California, 91-95.

Kelton, W.D. (1985). Analysis of simulation output data. In: Proceedings of the 1985 Winter Simulation Conference (D. Gantz, G. Blais, and S. Solomon, eds.). Institute of Electrical and Electronics Engineers, San Francisco, California, 33-35.

Law, A.M. (1980). Statistical analysis of the output data from terminating simulations. Naval Research Logistics Quarterly 27, 131-143.

Law, A.M. (1983). Statistical analysis of simulation output data. Operations Research 31, 983-1029.

Schruben, L.W. and Margolin, B.H. (1978). Pseudorandom number assignment in statistically designed simulation and distribution sampling experiments. Journal of the American Statistical Association 73, 504-520.

Smith, D.E. and Mauro, C.A. (1982). Factor screening in computer simulation. Simulation, 49-54.

Welch, P.D. (1983). The statistical analysis of simulation results. In: The Computer Performance Modeling Handbook (S.S. Lavenberg, ed.). Academic Press, New York, 268-328.

AUTHOR'S BIOGRAPHY

W. DAVID KELTON is an Associate Professor in the Department of Management Sciences, School of Management, at the University of Minnesota. He received a B.A. in Mathematics and M.S. and Ph.D. degrees in Industrial Engineering from the University of Wisconsin - Madison. His current research interests are in simulation methodology, statistical inference on stochastic processes, queueing, and quality control. He is a member of ASA, DSI, ORSA, and TIMS.

W. David Kelton
Department of Management Sciences
School of Management
University of Minnesota
Minneapolis, Minnesota 55455
(612) 624-8503