Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

SIMSTAT: A TOOL FOR SIMULATION ANALYSIS

Warren E. Blaisdell

MC² Analysis Systems
667 Pinewoods Ave.
Troy, NY 12180, U.S.A.

Jorge Haddock

Department of Decision Sciences & Engineering Systems
Rensselaer Polytechnic Institute
Troy, NY 12180, U.S.A.

ABSTRACT

SIMSTAT is an interactive graphical software tool that performs statistical analysis on simulation input and output data. It is designed to work seamlessly with most simulation languages, such as GPSS/H, SIMAN, ProModelPC, and SLAMSYSTEM. Simulation analysis is an essential part of every simulation study, but is often neglected or under-used because there has been no easy way to accomplish it. SIMSTAT bridges the technology gap by providing essential analysis techniques in a user-friendly environment.

1 INTRODUCTION

Most discrete event simulations are based on modeling the random behavior of systems. Because of the random nature of the input data, the output data is also random. And, due to this stochastic nature of the input and output variables, the simulation can be considered a sampling experiment where the simulation output values are observed for a given set of random numbers. Therefore, it is essential to employ valid statistical methods in order to reach legitimate conclusions.

In general, the purpose of simulation output analysis is to determine point and interval estimates for one or more system parameters. This information can then be used to compare several system configurations to determine the best.

1.1 Two Common Problems in Simulation Output

There are two common problems that arise in simulation output that prevent the use of standard classical statistics. These problems are autocorrelation (or non-independence), and initialization bias (or non-stationary, moving average behavior). Both of these conditions violate the IID (independent and identically distributed) assumption.

1.1.1 Autocorrelation

The easiest way to understand the autocorrelation problem is to consider a simple queueing system -- a bank. A bank has several tellers with one collective waiting line. Therefore, the amount of time that a person has to wait for a teller is directly influenced by the amount of time that the person just in front of him/her must wait. So, if we are concerned about the average time spent waiting for a teller, our sequential observations will be autocorrelated. Additionally, the degree of autocorrelation will depend on the traffic intensity and the rate of service.

This is a problem since standard point and interval estimation of the mean assumes that the data is independent. The simplest method of dealing with autocorrelation is by batching the waiting time data. Batching is a process where groups or "batches" of k consecutive observations are treated as single observations. This results in a new set of data in which the autocorrelation effect is usually alleviated, depending on the selection of k.

1.1.2 Initialization Bias

To understand the initialization bias problem we can consider a factory. As the factory is first built and operation begins, the system is empty. Then a "warm-up" period is encountered where the system progresses into a stable, "steady state" behavior.

If we are interested in the average time it takes for a part to be produced, this warm-up phase will, in general have a significantly different average production time than will the steady state. It is for this reason that the initialization phase causes a bias in the resulting processing time. If an analysis of the steady state behavior is desired, this initialization phase must be removed from the output in order to obtain accurate results.

The practical procedures to deal with this problem are Welch's graphical method (Law & Kelton 1991), and the Schruben, Singh and Tierney statistical test (Schruben, Singh, and Tierney 1983). Both of these procedures are based on the concept of truncating a certain number of observations from the beginning of the data, and are valuable to help ensure that the results accurately describe the steady state. Both of these procedures are also available in SIMSTAT.

1.2 Importance of proper statistical analysis

This is what just a few of the experts say about analysis:

> "The output variables [from a simulation] are estimates that contain random error, and therefore a proper statistical analysis is required. Such a philosophy is in contrast to the analyst who makes a single run and draws an inference from that single data point."
>
> Jerry Banks & John S. Carson, II (1984)

> "The results from a single run of a good simulation model often have a specious plausibility which can entrap the novice and the unwary into a quite unwarranted confidence in their accuracy. Simulations carried out without adequate statistical support are not merely useless but dangerous."
>
> Paul Bratley, Bennett L. Fox & Linus E. Schrage (1983)

> "...lack of definitive output data analyses appears to have been a major shortcoming of most simulation studies."
>
> Averill M. Law & W. David Kelton (1991)

> "I see a real need for tools that help in designing and analyzing simulation experiments. This area grossly lags the development of software for modeling."
>
> Lee Schruben (1987)

2 OVERVIEW OF SIMSTAT

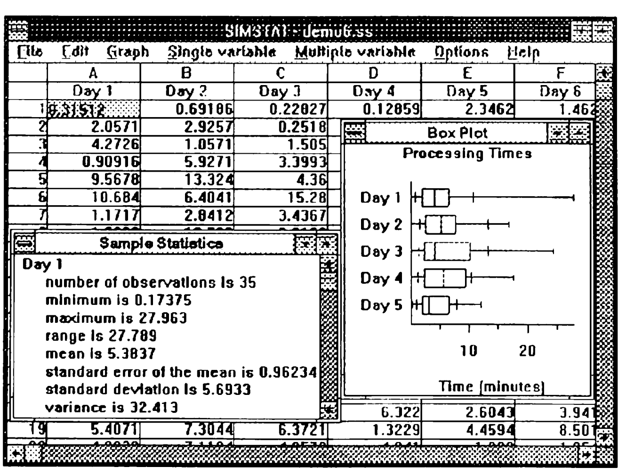

Figure 1: SIMSTAT

2.1 The Environment

SIMSTAT features easy to use pull down menus and is fully integrated into the Windows 3.0 and 3.1 environments. Data is maintained in a spreadsheet format for ease of examination and analysis. SIMSTAT takes advantage of the Windows Clipboard to allow the transferring of data and graphics directly to other Windows word processing, spreadsheet, and graphics packages. SIMSTAT takes full advantage of the extended memory and memory swapping features of Windows. Printing can be accomplished with a large variety of dot matrix, laser, ink jet, and color printers, as well as plotters. Additionally, SIMSTAT has an extensive on-line help facility to provide quick answers to problems that arise.

2.2 The Data

Data can be entered directly, or can be easily imported from GPSS/H (using the BPUTPIC command to generate either columns or tagged lines output files), SIMAN (binary output files), ProModelPC (Duration Reports), SLAM (plot data files), and other simulation languages (column aligned ASCII files).

SIMSTAT has no fixed limit on the amount of data that can be held in a spreadsheet, rather, the limit is only determined by the memory configuration of your system.

Data can be "marked" - to allow a versatile analysis to be done. This is beneficial to iteratively examine and determine the initialization phase of a system. Data can also be cut and pasted to and from different columns in the spreadsheet. This ability is necessary if the simulation does not yield its output data in an organized fashion.

2.3 Graphical Analysis

SIMSTAT has many graphics capabilities. To view the data, simple plots are available, as well as cumulative average plots and moving average plots. To assist in viewing the distribution of the data, SIMSTAT provides probability plots (normal, exponential, and uniform), frequency and cumulative frequency histograms, pie charts, box plots, as well as overlapping histograms of the data with theoretical distributions (for Beta, exponential Gamma,, lognormal, normal, triangular, uniform, and Weibull distributions), Also, SIMSTAT has bar charts for comparing multiple system designs, autocorrelation plots for checking independence, and Welch's method for graphically evaluating the warm-up phase.

2.4 Statistical Analysis

SIMSTAT calculates the sample statistics: mean, variance, minimum, maximum, range, standard error of the mean, standard deviation, coefficient of variation, skewness, and kurtosis. The hypothesis testing capabilities allow the testing of the mean and variance of one or two variables. Also included is the ability to perform a multiple means comparison using Fisher's Least Significant Difference method. SIMSTAT has six methods of calculating confidence intervals: replication, batch means, regenerative, spectral, autoregressive, and standardized time series. Finally, SIMSTAT includes two methods for performing variance reduction: antithetic variates and common random numbers. For further information on the technical details of these procedures, see Law and Kelton (1991) and Blaisdell (1991)

3. THE SIMULATION PROCESS

The best approach to understanding the power and flexibility of SIMSTAT is to see how it can be integrated into the simulation process. Although there is no fixed structure for the simulation process, the following will be used: 1. Define Problem, 2. Collect Data, 3. Create Model, 4. Verify/Debug Model, 5. Validate Model, 6. Analyze Model, 7. Present Results. For the sake of completeness, each step will include a brief description.

3.1 Define Problem

The first stage in every simulation study is to concisely define the problem, and can be accomplished in three parts. First, this involves a clear understanding of the system to be modeled as well as the bounds of the alternative systems to be evaluated. Second, there must be a determination of key system parameters, which are to be used as a basis for evaluation. Finally, it must be decided that the technique of simulation is indeed the best approach to the solution of this problem.

3.2 Collect Data

This step could easily be the most time consuming effort in the entire project. In order to accurately model any system, a thorough knowledge of all system parameters must be obtained. This may involve long periods of simply observing the system and recording what is seen. This will aid in recognizing the essential and non-essential portions of the system, and eventually in deciding which functions will need to be included in the model.

3.3 Create Model

The model creation step can be broken into two parts. First, the data that has been collected must be converted into probability distributions Then, the model must be written in some simulation language such as GPSS/H, or perhaps just in a programming language such as FORTRAN or C. Any alternative systems must also be codified.

SIMSTAT can be used to fit the data to theoretical probability distributions (Beta, exponential, Gamma, lognormal, normal, triangular, uniform and Weibull). Figure 2 shows overlapping histograms of data that was fit to a normal distribution using maximum likelihood estimation. Once the distribution has been fitted, the goodness of fit can be tested using the chi-squared or the Kolmogorov-Smirnov test feature. SIMSTAT can also be used to generate empirical distributions when theoretical distributions are not appropriate. For the case when data collection is not possible, SIMSTAT also provides functionality to graph up to ten overlapping probability distributions to aid in visual distribution selection.

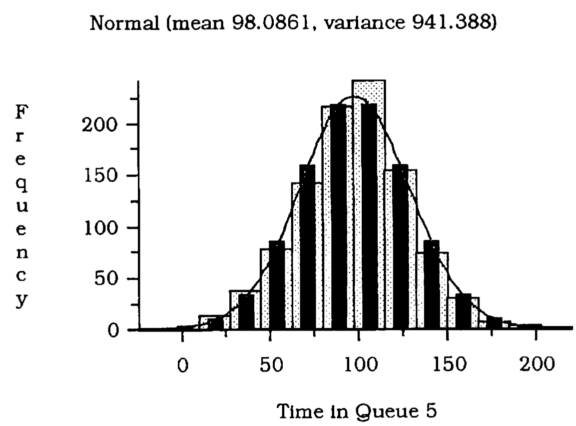

Figure 2: Fitting Distributions to Data in SIMSTAT

3.4 Verify/Debug Model

This is the tedious process of making sure that no typographic, syntax, and logic errors have been made in the model.

3.5 Validate Model

This step may be closely tied to the Model Verification step. Additionally, there should be an evaluation of some trial runs of the model to make sure that model is an adequate representation of the actual system. SIMSTAT can be used to test how appropriately the random samples used in the simulation fit the data that was collected. SIMSTAT can also be used to perform some hypothesis testing on the mean and variance of the key output variables. The data that was collected should be closely reflected in the behavior of the model. For a more in-depth study of model validation, see Sargent (1991).

3.6 Analyze Model

This step is essential. If the model is not analyzed properly, invalid conclusions may result. The simulationist must decide on the number of runs of the model and on the alternative systems as well as the length of each run. For a more thorough examination of the analysis process see Seila (1991) and Law (1983).

To illustrate the use of SIMSTAT for this step, we will limit our analysis to the two problems discussed above in Section 1.1. Figure 3 shows the processing time data from a simulation of 1000 parts and five runs displayed using Welch's method with a moving average window of 15. We can visually eliminate an initialization period of the first 200 parts. However, if we additionally wish to test this using the Schruben, Singh & Tierney (1983) test, we find that at the 90% confidence level there is still initialization bias present after the deletion of the first 200 and 300 parts. Not until 400 parts are deleted does the bias disappear (at the 90% level). Therefore, it may be necessary to run the simulation for more than 1000 parts to be sure that steady state has indeed been reached.

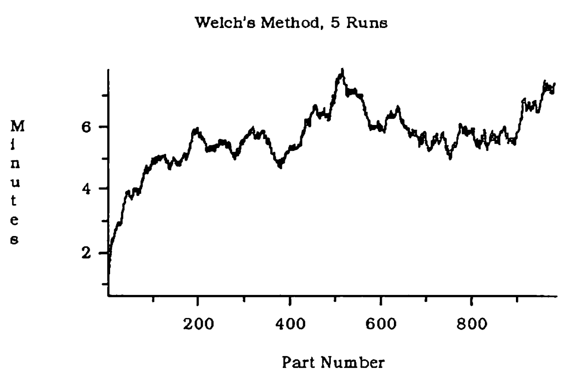

Figure 3: Welch's Method for Determining Initialization Bias in SIMSTAT

Figure 4 shows the strong autocorrelation structure that may exist in simulation output data, such as the bank problem discussed in Section 1.1. The data was batched into batches of size 15. The resulting autocorrelation structure of the batched data is shown in Figure 5. It is obvious that batching the data has had a significant effect on the autocorrelation. SIMSTAT performs the necessary calculations to batch the data, but does not automatically select a batch size.

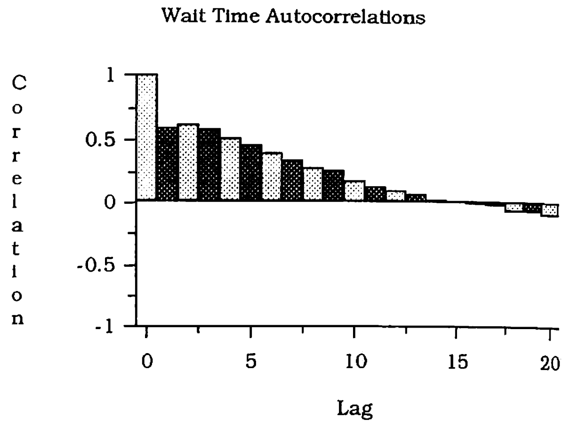

Figure 4: SIMSTAT Autocorrelation Plot for Wait Time Data

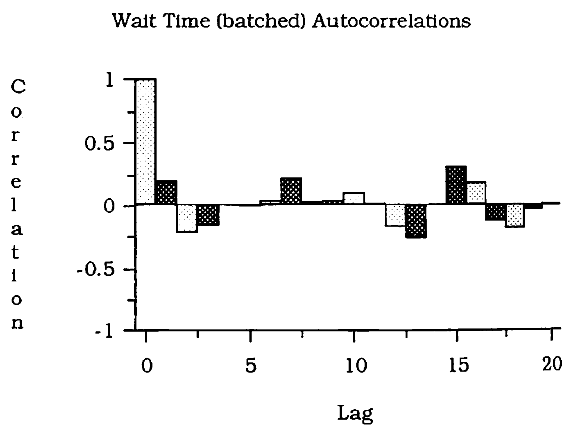

Figure 5: SIMSTAT Autocorrelation Plot for Batched Wait Time Data

3.7 Present Results

Once the analysis is completed, a report must be written to present the conclusions of the study. If a Windows word processing program is used (such as Word for Windows) to write the report, each graphical and statistical output window in SIMSTAT can be copied to the Window's Clipboard and from there be pasted directly into the report. This makes the report writing quite painless.

4 SUMMARY

Proper analysis is an essential facet of every valid simulation study. SIMSTAT is an easy to use graphical and statistical tool designed to work together with most simulation languages to provide the necessary analysis. SIMSTAT is also designed to work together with Windows based word processing programs to generate effective simulation summaries.

5 THE FUTURE

SIMSTAT is constantly being updated to include the recent trends in simulation analysis. Some of the features that are currently being planned for future versions of SIMSTAT include: experimental design, response surface methods, nonparametric hypothesis testing and confidence intervals for mean, variance and independence, the Fishman procedure for determining an optimal batch size, and simulation optimization.

REFERENCES

Banks, J. and J. S. Carson, II (1984). *Discrete-Event System Simulation*, Prentice-Hall, Inc., Englewood Cliffs, NJ.

Blaisdell, W. E. (1991). *SIMSTAT for Windows 3.0 User's Manual.* Troy, NY: MC^2 Analysis Systems.

Bratley, P., B. L. Fox and L.E. Schrage (1983). *A Guide to Simulation*, Springer-Verlag, NY.

Law, A. M. (1983). Statistical Analysis of Simulation Output Data. *Operations Research*, 31, pp. 983-1029.

Law, A. M. and W. D. Kelton (1991). *Simulation Modeling and Analysis,* 2nd ed., McGraw-Hill, NY.

Sargent, R. G. (1991). Simulation Model Verification and Validation. *Proceedings of the 1991 Winter Simulation Conference*, pp. 37-47.

Schruben, L. W. (1987). Using Simulation to Solve Problems: A Tutorial on the Analysis of Simulation Output. *Proceedings of the 1987 Winter Simulation Conference*

Schruben, L. W., H. Singh, and L. Tierney (1983). Optimal Tests for Initialization Bias in Simulation Output. *Operations Research*, 31, pp. 1167-1178.

Seila, A. F. (1991). Output Analysis for Simulation. *Proceedings of the 1991 Winter Simulation Conference*, pp. 28-36.

AUTHOR BIOGRAPHIES

WARREN E. BLAISDELL is a partner at MC^2 Analysis Systems and is in charge of the development of SIMSTAT. He received a B.S. in mathematics from Worcester Polytechnic Institute, and he received a M.S. in operations research and statistics from Rensselaer Polytechnic Institute. His areas of research include statistical computing, simulation analysis, and regenerative sampling.

JORGE HADDOCK is an associate professor of Industrial Engineering and Operations Research in the Department of Decision Sciences and Engineering Systems at Rensselaer Polytechnic Institute. He holds a BSCE from the University of Puerto Rico, a MSMgtE from Rensselaer, and a PhD in Industrial Engineering from Purdue University. His primary research interests involve modeling of manufacturing/production and inventory control systems, as well as the design and implementation of simulation modeling and analysis tools. He received the Outstanding Young Industrial Engineer Award from IIE in 1990 and the Martin Luther King, Jr. Faculty/Staff Award at Rensselaer in 1992.