SIMSTAT--A SIMPLE STATISTICAL PACKAGE TO SUPPORT SIMULATION

Susan L. Solomon
Eastern Washington University

Proceedings of the 1982
Winter Simulation Conference
Highland * Chao * Madrigal, Editors

82CH1844-0/82/0000-0307 $00.75 © 1982 IEEE

ABSTRACT

This paper discusses a package of thirteen programs written in conversational FORTRAN which facilitate many aspects of data collection and analysis associated with modeling of waiting line systems. While the package was developed for pedagogical purposes, it may also be used by simulation practitioners or managers with a knowledge of elementary statistics. Most statistics are nonparametric.

The package includes one program (DATCOL) to minimize the effort of data collection and to calculate sample performance measures; one program to compute descriptive statistics, sort data, average each adjacent m values and select every n-th value; one program to perform Chi-Square or Kolmogorov-Smirnov goodness of fit tests; one program to produce performance measures based on common analytical waiting line models; one program to generate uniform, Poisson, negative exponential or normal random variates; three programs to evaluate the random number stream using, respectively, Yule's test, the gap test and the runs-up-and-down test; one program to extract individual performance observations from the image of GPSS output stored on disk; one program to compute the Spearman rank correlation coefficient; one program to plot the data values; one program to apply three steady state heuristics to the data; and one program to perform the Mann-Whitney-Wilcoxon test or the Kruskal-Wallis "H" test of differences in alternative configurations.

Topic and Problem

At first glance simulation may seem to be a topic whose essence can be conveyed quickly to students: its purpose is to facilitate the search for better decisions when simpler modeling methods are inadequate; its mechanism is the substitution of random numbers for real events. While one cannot quarrel with these statements, they leave the student without even the minimal practical experience and understanding gained by solving a two-variable linear programming contrived homework problem. Since simulation is customized model building, it takes considerably greater time, effort and care to comprehend and execute than do analytical methods such as LP. Ironically, the difficulty of conceiving and executing a simulation causes instructors to spend their scarce allotted time on theory rather than the practice which is essential to a firm grasp of the subject.

Even when it is possible to devote an entire course to simulation, one quarter or one semester is typically not sufficient to cover a survey of popular simulation application areas, the theoretical underpinnings of simulation in mathematics and statistics, the philosophy of model building, the statistical tests which are desirable to perform in conjunction with simulation, rudiments of a simulation programming language, validation and verification of models, and selection and costing of alternatives. The far more desirable additional step of assisting individuals or small groups of students to confirm their understanding by carrying through the entire simulation process for systems chosen by the students is too much even to contemplate, in a one-quarter, one-night-a-week class environment!

Simulation classes at the upper-division and graduate levels tend to focus on one of three subsets of what there is to know about simulation: philosophy of modeling and applications surveys, simulation programming, or mathematical and statistical foundations of simulation. Large universities may be able to afford the luxury of a two-course sequence in simulation, in which all of these items can be covered, but even then, the burden of number crunching precludes the individual student from actually executing all aspects of a simulation model of a real-world system.

Textbooks tend to fall into the three categories of simulation coursework enumerated above. An example of a philosophy and applications survey text might be Julian Reitman's Computer Simulation Applications (Wiley, 1971); an example of a simulation programming text might be Thomas J. Schriber's Simulation Using GPSS (Wiley, 1974); and an example of a mathematical/statistical simulation text might be George S. Fishman's Concepts and Methods in Discrete Event Simulation (Wiley, 1973).

Further impediments in the path of the business student who seeks a working knowledge of simulation in business are the level of the math

Topic and Problem (continued)

and stat used in the texts, the nature of the applications discussed, and the programming capability assumed. In each category the material is better suited for majors in engineering and computer science than business. A new text, Computer Simulation in Business, by Hugh J. Watson (Wiley, 1981), has taken a significant step toward overcoming these obstacles, yet Watson asserts that simulation programming and full-scale projects must be relegated to a second course beyond the one in which his text is used.

Unique Features of this Approach

The key to this comprehensive yet compact simulation course is the SIMSTAT statistical package developed by this author over a two-year period. The package consists of thirteen programs written in conversational FORTRAN. The FORTRAN programs use format-free input statements and accept data entered from a terminal or from a disk file. With minimal introduction to system job control language the student can cause the output of one program to be written on the disk to serve as the input for the next program. The programs begin with a statement about their objective and the circumstances in which they should be used; prompt the student for data needed to complete the analysis; report computed and critical values of Z, Chi-Square and other relevant statistics from internally stored tables as needed; and warn the user when sample sizes fall below recommended levels for particular tests. The edit program extracts performance data from the image of standard GPSS output written onto a disk file, performance data which would otherwise have to be rekeyed laboriously from hard-copy printouts on many systems, and makes the data available on disk for statistical analysis. The only data entry performed by the student occurs when the original observations of interarrival time and service time for the individual project system have been gathered. All other data manipulation and calculation functions are automated.

The random number generation program also makes it possible for the instructor to write individual data sets under student computer identification numbers so that each student can analyze a different homework or exam problem.

Benefits of this Approach

The one-quarter, one-evening-per-week simulation course to be described has numerous benefits, including:

1. each student or pair of students gets experience, satisfaction, knowledge, confidence and perspective from performing every aspect of a real modeling project, with the instructor's guidance;
2. other students learn from the ongoing modeling experiences of their classmates;
3. the tedium and errors induced by large amounts of data transcription and manipulation manually are virtually eliminated;
4. students are prompted to do the right thing statistically without much review of classical statistics or extensive discussion of nonparametric statistics;
5. students may be given individualized homework assignments and examination questions using data generated by the computer and placed in the student's personal data file;
6. each student is told at the end of the quarter the billable cost of the computer time s/he used, so that the substantial dollar magnitude of a simulation project makes some lasting impression; and
7. students get much more excited about examining and improving a real system whose sites they can visit (perhaps even a site where they are employed) than they would if dry textbook cases or contrived short problems were assigned for analysis.

Level of Students

This course has been designed for students who have completed a minimum of one course in computer usage and one course in statistics. A second course in statistics and a survey course in quantitative methods are suggested but not required. In a typical section of 10-15 students about 50% are Seniors, 25% are Juniors, and 25% are graduate students.

DISCUSSION

Content

I have been teaching simulation for the past ten years. During that time I have found it frustrating that there were few textbooks which covered what I believe to be the three essential components of a simulation course--philosophy and applications, simulation programming and statistical methods--at the level and to the extent that I wanted them covered. There was no intersection between these two groups of desirable items. The problem was intensified when I switched from a large university on the semester system to a smaller university on the quarter system. Not only was the time factor critical because students had only ten weeks to learn to produce a workable model, but the students tended not to be as well prepared as those in the larger institution. While it would have been my

Discussion (continued)

preference to rely on an existing text, I felt forced to develop my own materials, especially given my conviction that a student learns simulation only by actually doing simulation, from beginning to end, in a real-world context of the student's choice.

Finding journal articles covering simulation philosophy and applications has been no great problem. Publications such as Decision Science, Interfaces, and Simuletter have been placed on reserve in our library and give the student a reasonable notion of the scope of simulation applications--from corporate modeling to inventory planning, to waiting line analysis. The student can digest these materials at a superficial level without very much class time devoted to their discussion.

Most students have at least an introduction to a general purpose programming language such as FORTRAN or BASIC and know the elements of terminal usage for file storage and retrieval. There is no choice about the need to teach certain elements of a simulation programming language, and GPSS is clearly the simplest to present. After about five years of experience it became obvious that approximately two dozen GPSS blocks were most often needed in the small queuing simulations assigned to students. Other blocks are introduced only insofar as a particular student described a situation in which only that block would perform the desired task satisfactorily. The summary sheet giving block types and field alternatives which appears in the user's manual for most GPSS compilers is duplicated and distributed to the students, and the user's manual itself placed on library reserve for student reference. An effort is made to restrict the set of GPSS blocks covered to those which are available in even the most rudimentary versions of GPSS. About 30% of the ten-week quarter has been devoted to GPSS programs of increasing difficulty, with one predecessor week covering various language alternatives for simulation modeling in different situations.

There is debate among simulation practitioners about the importance of statistical analysis in the several phases of model building and execution. Some senior people in the field believe that statistical analysis does not prove much but does demand a great deal of effort. My view is that statistical analysis is vital to the credibility of a simulation project. Most students have been exposed to introductory statistics including probability theory, theoretical distributions, sampling, estimation and hypothesis testing, but many have forgotten the concepts and mechanics and most have never applied these methods except to end-of-chapter problems. Most have no knowledge of nonparametric statistics except, perhaps, for Chi-Square tests. There are two extremely time consuming aspects of remedying student deficiencies in statistics to enable the activities supporting simulation. First, class time is very scarce, and second, the time available to evening students to complete homework assignments and projects is about equally scarce.

Over the past two years I have developed a twelve-program package which I have dubbed SIMSTAT (Simple Statistics to Support Simulation) to facilitate competent and efficient student application of statistics to various phases of simulation. Competency is encouraged because each program invites the student initially to ask for a brief description of the nature and purpose of the techniques used. The programs warn the student when sample size limitations are violated and report computed and critical values of appropriate statistics for the student data at the .05 level of significance. Efficiency is pursued in several ways. First, a student who is familiar with reading and writing disk files needs to type in only the original observations of interarrival time and service time for the project system. From that point all generation of homework data, calculation, summarization or modification is performed by the computer at the student's direction, and new data files may be written by certain programs in the package for analysis by other programs. A student who is frightened by the notion of disk files may manually enter data for any and all programs, although most students gain enough confidence to discard this crutch early in the quarter.

The first program in the SIMSTAT package is DATCOL, Data Collection and Reduction. DATCOL accepts as input the service start and end time for all transactions including those in the initial queue. It also accepts arrival time for transactions arriving after the observation period began. DATCOL calculates, prints and writes on disk each interarrival time, service time, waiting time, transit time, number in queue and server idle time, as well as summary statistics for these performance measures.

The second program in the SIMSTAT package which is used by the students is DESCST, Descriptive Statistics. Students process their data files for interarrival time and service time separately; the program reports an echo check of the original data so that data entry errors can be caught and fixed, a printout of the original data in sorted order to facilitate goodness of fit tests, the mean, variance and standard deviation of the data set. By examining the sorted data list, the summary statistics and the discrete or continuous character of the data, the student can make some preliminary guesses about which theoretical distributions, if any, might characterize the data. The student then calls the program GOFFIT, Goodness of Fit, and can check the data for conformity to a uniform, Poisson, negative exponential, Normal, or any desired empirical distribution. The program calculates any probability values needed from those distributions and even provides a method for determining class boundaries for a negative exponential test which yield the often-stipulated expected frequencies of five in each class, given the sample size. Both Chi-Square and Kolmogorov-Smirnov tests are included, but the student is encouraged to use Chi-Square to analyze real-world data because of imprecise knowledge of population parameters and to reserve the K-S test for later analysis of random number streams.

If the data prove to be approximately negative

Discussion (continued)

exponentially distributed, the student can call QMODEL to determine performance measures such as average waiting time and transit time, average number in the queue and system, and the probability of n transactions in the system, for several common analytical queuing formula contexts such as single and multiple servers, finite population and finite waiting area.

As a precursor to simulation model building the student is then introduced to random number generation via the RANGEN program, which invites the student to enter a seed and generate any desired quantity of psuedo-random digits with a mid-product generator which is easy to conceptualize. Inverse transformations for the uniform, Poisson, negative exponential and Normal distributions are included. The K-S goodness of fit test may be applied to show the consequences of insufficient sample size and mediocre performance of the mid-product generator in producing the desired distribution. The student proceeds to use GAP, the gap test of spread between consecutive random digits; YULES, Yule's test of sums of two, three or four random digits; and RUNUPD, the runs-up-and-down test, to evaluate the random number generator further.

The remaining programs in the SIMSTAT package pertain to statistical analysis of output of the simulation model written in GPSS. Before this analysis can be undertaken, the individual observations of performance created by a PRINT block must be extracted from the standard GPSS output format as written onto disk rather than onto the printer. This program, SIGN, searches the area where SAVEVALUE 1 prints, with a plus sign, and retrieves the contents of the SAVEVALUE, some performance measure such as transit time. The first test of these output observations is for serial correlation using SPEAR, a Spearman rank correlation test for association between consecutive values. If serial correlation is present, the student calls DESCST again to average 2, 3, or any desired m adjacent observations, or to select every n-th observation. DESCST writes a new data file on disk and this data file becomes the focus for further study unless the number of averages becomes too small. Should that occur, the programs notify the student that it is necessary to extend the original simulation run and expand the original data file of output observations. This process continues until the autocorrelation level is satisfactory.

Then the student calls two programs whose mission is to establish the presence or absence of steady state. PLOT enables a student to see the output values in the order of their occurrence, a portrait which can highlight where transients occur. STEADY implements three popular heuristics for detection of the onset of steady state. As a result of examining the output from these two programs, the student may elect to truncate some segment of the simulation data stream.

Once the student has proceeded to this stage, it may be assumed that the simulation output data are "clean," that is, free of autocorrelation and transients; and they are held until one or two postulated alternative configurations of the system have been modeled and their output likewise sanitized.

The final program in the SIMSTAT package is NANOVA, nonparametric analysis of variance, which performs either the Mann-Whitney-Wilcoxon test if there are only two alternative systems under consideration, or the Kruskal-Wallis H test of three or more alternatives. When the Kruskal-Wallis test is used, it is accompanied by Dunn's test of multiple comparisons to pinpoint where the significant differences, if any, may lie. After this is accomplished, the student knows whether or not the difference between or among the alternative configurations is statistically significant, and can then cost out the alternatives and prepare a report giving recommendations to management.

Presentation

Classroom presentation to date has assumed that there was no text to which the students might refer directly for guidance. Thus, an abundance of dittoed handouts with worked-out examples rather than here-today-gone-tomorrow transparencies are used. The complete project materials contained in the Appendix are placed on two-hour reserve in the library for additional reference. The computer programs in the SIMSTAT package are expressly designed to assist the student who is not sure what to do statistically, or why. Their terminal output prompts the student toward accurate responses. Since the FORTRAN source code is directly available to the student, it may be scrutinized and transported via cards, tape or paper copy when the student graduates.

Considerable class time is saved by this efficient use of supplemental teaching materials. Not only does this enable the compression of well in excess of one quarter's worth of simulation topics into a single quarter but leaves time available to assist individual students in the last half hour of class--students who often have no access to the instructor or terminals at any other time during the week. (Our main campus is twenty miles away from the minimal downtown facility occupied by the evening program.)

Another advantage of this personalized approach to coursework is that each student has ample practical statistical and programming experience, with no incentive to collaborate or cheat because no two student workloads are exactly the same. Thus, there is no compelling reason to put lengthy, tedious problem material on exams; the exams can focus the attention of the student back to the concepts and purposes of the practice undertaken.

Every student's term project is an experiential exercise to whose development other class members are continually exposed through formal and informal discussion. Homework mini-case series are based on student term projects in prior years.

Although I choose not to do it because our computer is down frequently and there is heavy demand for terminals in the evening, students could be given exams containing problems to be solved by computer during the examination period. Take-home exams are also a possibility since it is easy to create individual exam problems for each student, just as regular homework can be personalized.

Student simulation projects have included the underground shuttle train between the Capitol and the House and Senate office buildings, the rest-room facilities for a world's fair, a campus student health care service, several hospital operating room suites, a remote job entry computer station and the copiers in a large office building, as well as the traditional banks, gas stations, post offices, and professors' offices.