11. INSTRUCTIONAL APPLICATIONS 2

Chaired by Paula Goolkasian, *University of North Carolina*

A method for evaluating the effectiveness of educational software

STEVE COHEN, RICHARD CHECHILE, GEORGE SMITH,
FRANK TSAL, and GLEN BURNS
Tufts University, Medford, Massachusetts

A method for executing a detailed evaluation of educational software is described. Several issues are discussed, including the use of an experimental setting versus a field study, and the design of assessment instruments suited to evaluation of educational software. The instruments include a test of remedial skills, a 103-item test of conceptual understanding, and a system for recording students' use of the software. The method was used to evaluate ConStatS, a program for teaching conceptual understanding of probability and statistics. Preliminary results of the evaluation are presented.

The past several years have seen a good deal of research that has helped characterize how technology can serve education. Much, though not all, of this work has fallen into either of two categories. On the one hand, there is the bigger picture, which includes the meta-analytic studies of Kulik and Kulik (1991) and the framework offered by Kozma and Bangert-Downs (1987) for studying software applied to learning and teaching. Such studies offer guidelines and expectations, which do not necessarily translate well into concrete concerns and choices for the instructor with fairly well-defined discipline-specific teaching concerns. On the other hand, there is a host of small, fairly individualized studies on the effect of specific educational technologies in local settings. Many of the smaller studies lack a robust enough design to permit transfer of results.

Welsh (1993), Duncan (1993), Ransdell (1993), and Castellan (1993) have offered a set of guidelines for improving evaluation of educational software. Their papers emphasize the value of evaluation results that permit instructors to make well-informed decisions about the effective use of technology for specific educational ends. The present study deals with many of the points raised in those papers. In particular, we present a method for

evaluating technology that has been developed over the past 2 years. This method is currently being used to evaluate ConStatS, a program developed at the Tufts University Curricular Software Studio for teaching introductory probability and statistics.¹ As we present the method, we will make reference to this evaluation. We believe that all or parts of this method can be useful for evaluating educational technology in general.

Where Does ConStatS Fit Into Educational Technology?

To aid the description of just how ConStatS fits into the broad range of products that loosely define educational technology, we offer the following taxonomy for describing how technology might serve education. Technology may (1) address remedial problems, (2) improve efficiency, and (3) help students gain a deep understanding of a particular topic or domain. This taxonomy is heuristic. It is not complete (for instance, technology may be used to support evaluation), and a designated piece of technology might address more than one objective. However, the classification provides a way of thinking that can help guide the design of a new educational technology and assist in establishing methods and goals for evaluation. ConStatS is designed to help students reach the third objective, a deep conceptual understanding of introductory probability and statistics. Hence, one principal goal of the evaluation was to determine how well that objective has been met.

ConStatS is curricular and educational software, designed with a distinct educational goal and pedagogical philosophy. The following two theses, which have served

The research in this article was supported in part by the Fund for the Improvement of Postsecondary Education (FISPE), Grant 116AH70624. The authors would like to thank Durwood Marshall for statistical consultation, and Barbara Alarie, Paula Fisher, Christine Sossaman, and Joe Debold for document preparation support. Correspondence concerning this article may be sent to S. Cohen, Curricular Software Studio, Tufts University, Medford, MA 02155 (e-mail: SCohen@Jade.Tufts.edu).

as points of departure for the evaluation, also embody the purpose of the software.

1. An active, experimental style of learning is indispensable for achieving a substantial conceptual understanding of statistics—or of any other subject matter built around a comparably abstract theory.

2. If carefully designed with the preceding aim in mind, curricular software can enable a much wider range of students to engage in this style of learning and hence to reach levels of understanding currently being achieved only by exceptional students.

ConStatS consists of nine independent programs that cover about the first 10 weeks of an introductory statistics class. The students are always in control; not only do they choose what to do next, but nothing occurs on the screen except through their choices. Each screen presents several choices, posed as questions. These choices are those that have to be made to determine some result that will appear next—as, for example, the choice of data range and the number and type of intervals in order to draw a histogram. Students unsure of which choice to make can access a Why or Help option for any question. Why options typically offer one or two sentences that present students with a reason for making a choice. Help options provide extensive explanations and instructions on how to proceed. By forcing students into an experimental and generative style of learning, which only the superior students seem to use on their own, we expect that a much larger fraction of students will develop a deeper understanding of concepts taught in an introductory statistics course. Figure 1 shows an experiment screen from ConStatS, where students can change the number of class intervals used to construct the histogram on the left and see how the display of the distribution changes. Each time a student tries a new number of intervals it is considered an experiment.

Field Testing and Multiple Classes

The evaluation was designed as a field study rather than as an experiment or a set of experiments. From the outset, it became clear that instructors willing to participate in the evaluation could not reach consensus on how to include ConStatS in their courses. Seven different faculty members who teach introductory statistics at Tufts participated in the evaluation, and each had particular beliefs about how to use ConStatS. In some cases, these beliefs were motivated by pedagogical concerns, in others by the unique way each discipline uses statistics, and finally by the points at which each part of the software fit into existing curricula. For instance, several faculty members participating in the evaluation added a computer laboratory for ConStatS, but another professor simply dropped a lecture per week in order to have a ConStatS lab. The Tufts classes participating in the study (there were two classes from outside universities participating as well) came from three departments: Biology, Psychology, and Economics. Class sizes ranged from 12 to 110 students. Three other Tufts classes served as control groups.

Figure 1. A screen designed to permit students to experiment with class intervals on histograms. If students are not sure which option to use, they can select Why or Help.

Maintaining experimental control in just one course is very difficult. Since ConStatS covers such a large fraction of the course, it seemed almost inevitable that the kinds of confounds Duncan (1993) mentioned would surface and undermine experimental conditions and internal validity. It became clear that it would be much better to use the program in a variety of settings and then examine the various outcomes to learn which settings proved most or least effective, and why. We sought to achieve a range of outcomes. Thus we could avoid the problem of the developer as evaluator (someone with a deep familiarity and commitment to the product achieving artificially high results), yet leave a place for considering how much improvement the dedicated instructor with a deep belief in the technology might expect. We could also balance traditional outcome-based assessment (i.e., whether or not ConStatS helps students develop a deeper understanding of statistics) with the role of the educator as evaluator described by Cronbach (1980). Finally, the field-based approach points to the need for instruments and methods to help one keep track of different classes and sites. Concerns range from making sure that the sites use the software, to implementation strategies, to backgrounds and profiles of students. A large part of this evaluation method has been designed to address these concerns.

Defining Outcomes: Detailed Content Analysis

At least two principal criticisms have been levied against outcome-based evaluations of educational software: (1) evaluators ignore principles of test construction (Duncan, 1993), and (2) it is not clear that content addressed by the test is a fair and complete representation of what is in the software (Megarry, 1979).

The principal outcome assessment instrument that we developed was a short-answer essay test of conceptual un-

derstanding. The test had to meet several criteria. It had to (1) cover the broad range of concepts contained in the software, (2) help us understand which parts of the software were helping students learn and which parts needed to be improved, and (3) test conceptual understanding. To be certain that the test covered the range of topics addressed by the software, we first developed a database of the comprehension points in ConStatS. We defined each program as a series of screens, considered the experiments students could perform on each screen, and then defined the comprehension points we expected the students to learn. Each comprehension point was specified as either *explicit* (a concept the experiment was designed to illustrate) or *implicit* (a subtle result the students might or might not notice). This exercise not only helped us define the domain to be tested, but also designated responsibility to each part of the software for teaching specific concepts.

Two methodological issues are worth noting. First, ConStatS lends itself nicely to a division into sets of distinct screens. For instance, Figure 1 shows a ConStatS screen. Users cannot modify this screen by adding new elements (multiple graphs, windows with statistics, etc.) as they might be able to in a tool-based package. If students could systematically add elements to the screen, at some point it would not be clear exactly what features they were interacting with. Second, creating a database of concepts is not the only appropriate method for defining a domain. For instance, in an evaluation of the effect of computer simulation on teaching problem solving, Woodward, Carnine, and Gersten, (1988) used a tree diagram to represent the problem-solving strategies students might use. How best to represent content is in part a function of the kinds of outcomes emphasized in the evaluation.

The content analysis performed on ConStatS yielded over 1,000 comprehension points. However, many of the points were similar. To reduce the assessment to a manageable task, we combined similar comprehension points into clusters and defined a unifying concept for each cluster. Table 1 shows an edited cluster for understanding the role that intervals play in displaying data on histograms. The original cluster had 34 comprehension points. While the principal screen for experimenting with intervals (displayed previously) resides in the Displaying Data program (Program 1), students can experiment with class intervals in other ConStatS programs as well. The cluster in Table 1 indicates the program and the screen and whether or not the comprehension point is explicitly taught (denoted by an X) or implicitly taught (denoted by an I).

The clustering method produced a set of testable comprehension points, yet still maintained a formal connection between the comprehension points and the various screens and options. Ultimately we defined 103 comprehension points, each of them treated as a dependent measure. This put us in a position to assess how well ConStatS helps students learn each of the 103 concepts, and to understand which parts of the program are responsible for success or failure. It is likely that some form of cluster-

Table 1
Set of Comprehension Points Defining a Conceptual Cluster:
Understanding the Role of Intervals in the Display
of Data on Histograms

Program	Screen		How does changing the number of class intervals affect the way data is displayed on histograms?
1	10.1	I	The heights of the data in the intervals illustrate how the data are distributed over the range of values.
1	12	X	The number of intervals on a histogram is arbitrary.
2	6	I	The data points in an interval are not necessarily ordered or equal.

Note—X, the comprehension point is *explicitly* taught; I, comprehension point is *implicitly* taught.

ing will be needed in almost any evaluation of educational software.

Test Construction

For each comprehension point, we constructed a question that tested conceptual understanding. All 103 questions were subject to the following criteria: (1) the statistical concept was included in the software, (2) the question was appropriate for an introductory course in statistics, and (3) the question assessed conceptual understanding. The questions were reviewed against these criteria first by internal members of the evaluation team, and then by two outside statistical consultants and professors of quantitative methods. The 103 questions were divided into three tests that covered similar but not identical content. Each test had about 6 questions on the following topics: displaying data, descriptive statistics, transformations, bivariate representation and regression, and probability distributions and sampling.

Although some questions emphasized one of the comprehension points included in the cluster, most were focused on the overall concept defined by the cluster. In either case, the questions were constructed so that if a student answered correctly, we could be fairly confident that they understood the range of related points in the cluster. (If there had been fewer total items to test, we would have selectively tested several subordinate concepts. This kind of testing might be useful as a validity measure for transfer of knowledge.) The questions typically forced students to transfer knowledge into new or counterintuitive situations and exposed the difference between a declarative or procedural understanding and a deep conceptual understanding. We were able to use the test results to evaluate the patterns of mistakes and understand what students were thinking when they answered incorrectly.

Analyzing the patterns of mistakes offers a unique look into how much attention a topic deserves in the curriculum. To use the example of changing intervals again, while more students who used ConStatS were able to discern the effect of a different number of intervals, the students who were mistaken often made the same error.

These students often believed that they were looking at different data, and they drew very different conclusions. The patterns of errors offer a kind of qualitative loss function which suggests what kinds of errors students might make if they do not understand a concept. It may be that the concept of class intervals does not appear, at face value, to be an important topic. The nature of the mistakes may offer a reason to believe otherwise.

In addition to the test of conceptual understanding, we developed a short, 10-question test of the minimum mathematical skills necessary for using ConStatS. Although there is very little math in ConStatS, students need to work with fractions, ratios, decimals, and variables. Students without the minimum skills could easily represent a different population than would those who have no remedial problems. This test was constructed in the same manner as the comprehension test, first by going through ConStatS and determining the basic mathematical skills needed for one to use the software and then by creating a question to test the skill.

Tracing Student Use

Perhaps the most ambitious and novel component of the evaluation is a system we developed and installed in the software to record, or trace, how students use ConStatS. The traces offer a method of additional control when one is studying the use of curricular software in a field setting, as well as a real opportunity to understand the kinds of experimental styles of learning employed by students.

The traces are embedded in the program code to account for all the time each student, or pair of students, spends with each ConStatS program. Table 2 shows a template for the traces with specific trace information that came from the class interval experiment screen in Figure 1. The Time field indicates the number of seconds since the last trace was recorded. The Program field indicates which ConStatS program is being used (Program 1, Displaying Data), and the Screen is Number 12, with four options. The Interaction field indicates the specific activity—that is, the student is performing an experiment. The user has selected Option 3, changed the number of intervals, and then clicked “Go On.” The type of information recorded in the Parameter field may be different for each kind of experiment. In this instance, we record the number of intervals selected by the student. Had the interaction provided a message for the student to read, the length of the message would have been recorded in the Text field.

A taxonomy of seven interaction categories, with additional subcategories, defines how a student spends his/

Table 3
Taxonomy for Capturing Student Use of Educational Software

Taxonomy	Examples of the Interaction
Navigation	Moving to a new or previous experiment screen
Experiment	Changing intervals
Information retrieval	Asking for help
Result survey	Time spent reviewing an experimental result
Screen survey	Time spent reviewing a screen before making a choice
Watching a process	Step-by-step building a histogram
Animation	Watching a simulated coin flip during a sampling experiment

her time with the program (see Table 3). In a field study setting, records of student use offer a simple check on whether or not the program(s) have actually been used. Were they used for 30 min or 2 h? In the ConStatS evaluation, the participating instructors agreed ahead of time to use at least seven of the nine programs. In some cases, not all seven programs were used, and in other instances, programs were used only briefly. Woodward et al. (1988) indicate that research on the effectiveness of computer simulation is plagued by short intervention—students often work through the simulation only once. The traces offer a way to monitor all these variables in a field setting.

The traces also offer an opportunity to examine experimental styles of learning. For instance, how do students go about changing the number of intervals? Do they try more than one experiment and a range of different sized intervals? Many instructors offer specific assignments for students to use with ConStatS, but typically the assignments permit students a fair amount of flexibility once they arrive at an experiment screen. Finally, trace files can become large and difficult to manage. They range from about 10,000 bytes to 250,000 bytes per student per session with ConStatS.

Additional Techniques and Instruments

In addition to the techniques already described, we also used the following instruments and methods to supplement the evaluation: (1) a 7-point survey of course content to assess which topics in ConStatS were emphasized in the lectures, homework, and textbook; (2) a brief survey of students to determine whether they felt the software was useful and used well by the instructor; and (3) an open line of communication during the evaluation for comments and suggestions by students, instructors, and teaching assistants. We have not integrated the results from the use of these instruments with the results presented in the next section.

Preliminary Results

We are currently beginning the 3rd year of this 3-year evaluation. The evaluation took place in the fall 1992 semester and the spring and summer semesters of 1993. Classes at three universities participated. Seven classes, with a total of 327 students, participated as experimental groups, and three classes (from Tufts), with a total of 63

Table 2
A Template for Recording, or Tracing, Each Interaction a User Has With a Program

Time	Program	Screen	#Options	Option	Interaction
22.23	1	12	4	3	Experiment
Text	Parameters				
0	4				

Note—Each heading represents a field (see text for explanations).

students, participated as control groups. Two hundred and twenty-six experimental subjects came from the introductory statistics course for psychology at Tufts. Other subjects were enrolled in statistics courses for biology, education, child study, and economics. Two of the control group classes (54 students) were taught by a member of the team that designed ConStatS. All control groups used tool-based statistics software for mainframe and personal computers, but did not use ConStatS.

Questions on the comprehension test were scored on a 5-point scale: 1 = *no understanding*; 2 = *little understanding*; 3 = *moderate understanding*; 4 = *good understanding*; 5 = *excellent understanding*. We computed the average results for each question and then computed difference scores by subtracting the results of the control group from those of the experimental group.

The difference scores showed that the experimental group did better on 92 of the 103 questions. The 10 questions on which the experimental group showed the greatest improvement over the control group involved transformations (z scores and linear transformations), probability, and the concepts of deviation and sensitivity of summary measures. Although it is not clear why the largest gains appeared in these topics, the results point to the value of a detailed evaluation. Clearly these topics were not being well understood, and curricular software can help improve conceptual understanding on many of the concepts.

An item analysis revealed six questions with $r > .5$ for overall success on the comprehension test. Five of the six questions addressed concepts and representations of probability. Although it is not clear what underlies this correlation, it is a surprise. There is a shift toward analyzing data, rather than teaching probability, in introductory statistics classes. We are currently examining this result in more detail.

The traces were very useful for understanding differences between experimental classes on subscales of the test. Large differences between two Tufts experimental classes on the probability subscale (61% vs. 45%) became understandable when the traces revealed that students in one class used the program fairly actively and extensively, whereas the other class barely used the program at all.

In Figure 2, the comprehension test results for all classes are plotted as a function of the performances on the minimal skills test. A 10 is the highest score, and 8 indicates an 8 or lower. Improvement in conceptual understanding jumps considerably when minimal skills scores reach 9 and then 10. This trend is also evident when the experimental subjects are separated into two groups—students from Tufts or from outside universities. With respect to a range of outcomes over all experimental classes, students scoring a 10 on the minimal skills test showed a 25%–60% gain in conceptual understanding.

Feedback

The results from the evaluation were used to improve the program and its use, as well as teachers' understand-

Figure 2. Results from the 1st year of the evaluation, as a function of performance on the minimal skills test. The experimental group includes all classes that used the program, both in and outside of Tufts.

ing of impediments to learning probability and statistics. Like Castellan (1993), we recognize the need for cyclic improvement.

One unique form of feedback that we utilized was the patterns of errors. On 54 out of the 103 questions, there was a pattern to the wrong answers. For each of these questions, we went back to the original cluster, recorded the concept, and listed the common misconceptions that the students exhibited. This list was provided to instructors participating in the evaluation. We hoped that this list would help teachers target the misconceptions during lectures, and put students in a better position to use the program effectively. The misconceptions and overall test results were also used to design 15 new experiment screens and several new options on existing screens. These changes, and the effects they do or do not show, will provide a measure of the evaluation method as a means of improving instructional technology and the curriculum.

Final Remarks About the Method

This evaluation method represents an attempt to marry standard multisite program evaluation with techniques used in cognition, knowledge representation, and criterion-based test construction. Together, the methodologies offer both a top-down and a bottom-up way of considering instructional evaluation. The top-down component is preserved by way of means-based comparisons across several sites and classes. Because of the range of student populations and course offerings under the title Introductory Statistics, as well as the variety of technology bases available, it is crucial to acquire results from several settings. Multi-

site results appear to be a necessary condition for the preservation of validity in this kind of evaluation.

The methods used to construct the comprehension test and the trace facility permit a much more fine-grained look at how and why the software was and was not successful. The detailed links between the comprehension test and the software provided precise feedback on what concepts the students were learning and which parts of the program needed to be improved. The trace data have been useful for conditioning comprehension test results on implementation issues such as how long and how often each part of the program was used. These bottom-up methods help bridge the gap between big-picture evaluations and local, often discipline-specific evaluations. Ultimately, this puts instructors in a better position to make detailed and well-targeted choices about where and how to use technology in a curriculum.

REFERENCES

- CASTELLAN, N. J., JR. (1993). Evaluating information technology in teaching and learning. *Behavior Research Methods, Instruments, & Computers*, **25**, 233-237.
- CRONBACH, L. J. (1980). *Toward reform of program evaluation*. San Francisco: Jossey Bass.
- DUNCAN, N. C. (1993). Evaluation of instructional software: Design considerations and recommendations. *Behavior Research Methods, Instruments, & Computers*, **25**, 223-227.
- KOZMA, R., & BANGERT-DOWNS, R. (1987). *Design in context: A conceptual framework for the study of computer software in higher education* (Tech. Rep. No 87-F-001.0) Ann Arbor: University of Michigan, National Center for Research to Improve Postsecondary Teaching and Learning.
- KULIK, J. A., & KULIK, C. L. (1991). Effectiveness of computer based instruction: An updated analysis. *Computers in Human Behavior*, **7**, 75-94.
- MEGARRY, J. (1979). Monitoring. *Simulation/Games for Learning*, **9**, 170-177.
- RANSEDELL, S. (1993). Educational software evaluation research: Balancing internal, external, and ecological validity. *Behavior Research Methods, Instruments, & Computers*, **25**, 228-232.
- WELSH, J. A. (1993). The effectiveness of computerized instruction at the college level: Five suggestions for successful implementation. *Behavior Research Methods, Instruments, & Computers*, **25**, 220-222.
- WOODWARD, J., CARNINE, D., & GERSTEN, R. (1988). Teaching problem solving through computer simulations. *American Educational Research Journal*, **25**, 72-86.

NOTE

1. ConStatS consists of three major parts, each of which contains several distinct programs covering different topic areas.
 - Data representation.* Displaying data, summary statistics, transformations, describing bivariate data.
 - Probability.* Probability measurement and probability distributions.
 - Sampling.* Sampling distributions, sampling errors, a sampling problem.
 The nine programs together provide 15-20 h of curricular material. Additional programs are under development.