From **Brunelli, Lina & Cicchitelli, Giuseppe (editors). Proceedings of the First Scientific Meeting (of the IASE).** Università di Perugia (Italy), 1994. Pages 237-245. Copyright holder: University of Perugia. Permission granted by Dipartmento di Scienze Statistiche to the IASE to make this book freely available on the Internet. This pdf file is from the IASE website at http://www.stat.auckland.nz/~iase/publications/proc1993. Copies of the complete Proceedings are available for 10 Euros from the ISI (International Statistical Institute). See http://isi.cbs.nl/sale-iase.htm for details.

DESIGNING CURRICULAR SOFTWARE FOR CONCEPTUALIZING STATISTICS

Steve Cohen, George Smith, Richard Chechile and Robert Cook
Curricular Software Studio, Arena User Area
Tufts University Medford, MA 02155, USA

1. Introduction

Over the past decade, software has found its way into the curriculum for a variety of educational purposes. This has been especially true in the case of introductory statistics courses. There are at least two principal uses for technology in such introductory courses. The first is to free students of excessive computational responsibilities, thereby enabling them to spend more time analysing data and assessing results. The second is to help students to get a deeper understanding of the fundamental concepts taught in an introductory course, thereby putting the students in a better position to use powerful tool-based software and to interpret results. In a sense, both of these uses have the same aim, namely to improve the general quantitative reasoning skills of the students. In many respects this is the greatest challenge of an introductory statistics course, or of any course with a large quantitative component.

For years, mainframe tool-based software was used to serve both purposes. However, the promise of helping students to get a deeper understanding of fundamental concepts seemed specifically suited to the more interactive, easy-to-use programs that run on personal computers. A link between interactivity and deep understanding could be observed by watching how the small fraction of superior students in an introductory statistics class mastered concepts. Rather than simply memorising formulae, they would perform thought experiments, mentally clarifying and dissecting the concepts. They might look at a computational formula for a sample variance and wonder what would happen if the deviations were not squared. Posing these kinds of questions and discovering both answers and new questions seems to be the hallmark characteristic of students who leave an introductory statistics course with a facility for quantitative reasoning. Software that puts students in a position to pose questions and execute thought experiments on a computer might accordingly enable a much larger number of students to develop a deeper understanding of statistical concepts.

2. An initial design

In 1987, Tufts University's Curricular Software Studio, with support from FIPSE (Fund for the Improvement of Post-secondary Education), began to develop a set of microcomputer based programs, running under Microsoft Windows, that would permit students to explore concepts in introductory statistics. We decided that the programs needed to have the following design features and functions. Each program would cover a focused topic, such as Displaying Data, that typically takes one or two weeks in an introductory course. The programs would take advantage of a point-and-click graphical user interface that would allow students to access options and to execute commands easily. The pull-down menu options at the top of the program would contain a number of functions that would permit students to manipulate graphs, to examine data, and perform experiments. Students would have instant access to a variety of display options. The programs would offer discipline-specific data sets and random variables, so that students in the various discipline-specific statistics courses could encounter generic statistical concepts in a context with which they were familiar. After a year of development, we introduced seven programs into introductory statistics courses at Tufts, covering topics ranging from Displaying Data to Sampling. Each had a design similar to the Displaying Data program shown in Fig. 1.

We had very little success with this software. Only a small fraction of students, those that typically perform thought experiments and come away with a deep understanding without the aid of software, actually used the programs for an extended period of time. A certain number of the students would point and click, though without much purpose. Perhaps they expected that at some juncture a graph, table or result would appear on the screen that would help forge a useful picture of what had transpired. These students seemed willing to explore, but they were not capable of posing questions and executing experiments. Hence, they typically were not in a position to interpret a result. Finally, the majority of students did very little work with the programs. After a short time, they simply dismissed the opportunity to use the programs, mostly because it was not clear to them just what they were supposed to do.

We used the spring semester to examine our discouraging results and to reflect on the kinds of redesign that might prove useful. The software was originally written with the idea of giving students a simple-to-use platform for exploring a topic. The programs provided students with an environment rich in functionality, but with too little structure to permit those with sparse knowledge of the topic to pose questions or to explore successfully and intelligently. For instance, students using the Displaying

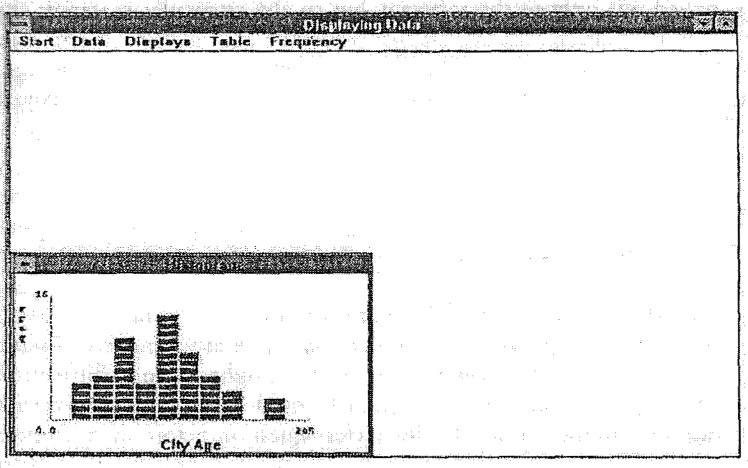

Figure 1

Data program shown in Fig. 1 might create the histogram shown in the bottom left hand corner. After they had created the histogram, we hoped they would perform the following four operations:

- Formulate a question about the histogram
- Determine which menu option best addresses the question
- Select the sub-option which best addresses the question
- Successfully interpret the result

For an excellent student, this is a challenging exercise. For a less than excellent student, it is often overwhelming. When the task becomes overwhelming, it appears that students often cease to explore, and active experimentation is replaced by a short session of button pressing and recording of results. All but the very best students seemed to fall back into a passive style of learning at the earliest opportunity. The challenge of redesigning the software was to provide students with enough structure to pursue an active, experimental style of learning without having the program tell them what to do. Direct instruction would allow students to assume a passive style of learning, and they would then be likely to consider the interactive experiments as a rote exercise. Our concern about this applied not only to the software, but to any curricular materials that went with it. Written assignments that dictated what buttons to press, what experiments to perform, and what results to record might also encourage a student to fall into the very passive style of learning that we were trying to eliminate.

3. Redesigning the software

We tried to address each of the four operations that seemed to stand in the way of students performing and successfully interpreting experiments. We saw two flaws in the original design that might make it difficult for students to pose a question. Firstly, they could perform any number of statistical operations without having a clear question before them to begin with. For instance, faced with a data table, students could create histograms, sort the data, consider relative frequency, etc. Narrowing down the number of choices seemed imperative. Secondly, even if the students did begin with a clear question, they might not see how to translate it into a program option.

To remedy these problems, we first decided to divide each program, and the functionality it offered, into a series of screens. Each screen would offer only a few options.

Some of the options would permit the students to perform an experiment about a specific concept, and others would permit the student to move to another screen and perform a new set of experiments. To help students focus on the particular concepts and set of experiments on a screen, we also provided a one or two sentence "scaffolding" at the top of the screen. For instance, a screen in the Displaying Data program might be devoted to examining how the number of intervals used to draw a histogram affect how data are lumped and consequently how they end up being displayed graphically.

In order to permit a smoother translation between an experimental option and a question, we presented all the options on a screen as questions. To follow the previous example, the program would prompt the student, "How many intervals would you like to try?", and then offer a set of options, posed as questions, including the opportunity to enter their own number.

Finally, for each question, we offered both a WHY and a HELP button. When we originally introduced the software, we observed three fairly distinct patterns of response among those using it. Superior students tended to learn effectively from it from the outset, with no need of prompting. Others would use the programs, but without much purpose or sense of direction. Finally, most would simply give up shortly after they began. The WHY option is intended to help those who were having difficulty seeing a purpose. Hitting the WHY button when facing a choice produces a reason why the choice is an appropriate one to be facing. This usually takes the form of a one sentence statement of a typical consideration that someone might focus on when making the choice. For example, hitting WHY when hesitating over the question, "Do you want to change the number of intervals?", produces, "Maybe the histogram will take on a very different appearance with a different number of intervals."– just the sort of thought that a good teacher might whisper in the ear of a student who is hesitating in the middle of an experiment. We also hoped that the ubiquitous presence of the WHY button would induce students to begin asking themselves, rather than the program, why they might want to pursue each option. In the long run this might even yield a reflex action of posing questions and asking what experiments would be appropriate for answering them.

The HELP option is intended to rescue those who feel lost in trying to use the software to execute experiments. HELP responses are often one or two paragraphs aimed at helping the student to understand the goal of the experiment. Sometimes the text includes instructions, detailing how to carry out specific experiments yielding clearly interpretable results. But even in these cases the student is actively eliciting the information, looking for specific things that will help them to take the next step, in much the way that a superior student will take a quick look at one or two pages of a book when working something out in a thought experiment.

4. ConStatS: the redesigned package

The first screen of the redesigned Displaying Data program is shown in Fig. 2.

The scaffolding at the top orients the student. There are three options, posed as questions. For any of the options, the student can ask WHY, or select HELP. The first option, learning about the data set, is a simple experiment. Students selecting this option will see a short description about the data, why it was collected, etc. The other options take students to different screens. These choices lead to different pathways through the program, pathways that often converge into one another. These pathways

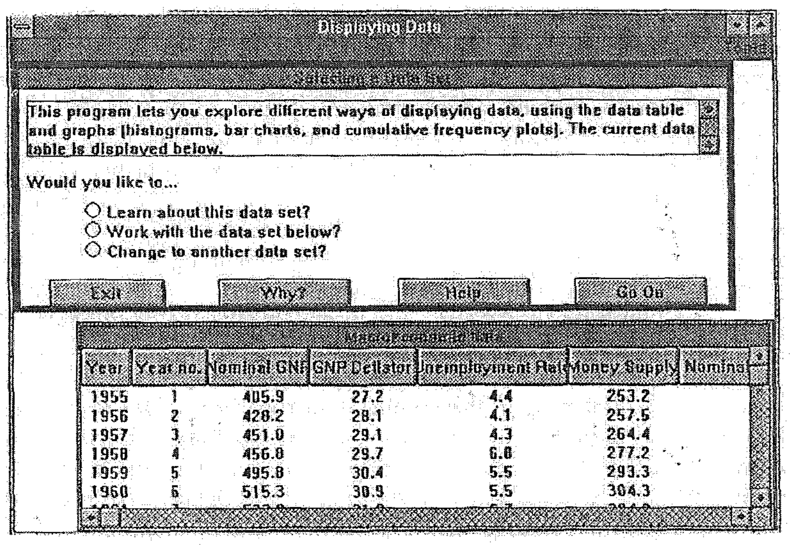

Figure 2

provide an unobtrusive built-in structure that helps to guide the student along in an orderly fashion. More guidance is provided at some places by making the decision one between a default value offered by the program and a value of the student's own choosing. Each screen has a "Go Back" option so that the student can always back up along a pathway to review or reconsider earlier choices. The students are always in control, not just in the sense that they choose what to do next, but in the sense that nothing ever happens on the screen except through choices that they make.

Fig. 3 shows an experiment screen from the Probability Distributions program. It enables students to examine how probability distributions convey probability, and to compare the PDF and the CDF. Clicking on the abscissa of either the PDF or CDF will update the highlighted interval on both distributions. The corresponding probability is illustrated in the pie chart to the left. Students do not arrive at this screen until they have selected a distribution for study, and have examined how varying parameter values affects the form that the distribution takes on.

In all, ConStatS consists of nine independent programs that offer about 15 to 20 hours of curricular time for the first 7 to 10 weeks of an introductory course. As redesigned, ConStatS embodies the view that microcomputer-based curricular software is truly effective when it functions as a medium in its own right, entirely different from classroom

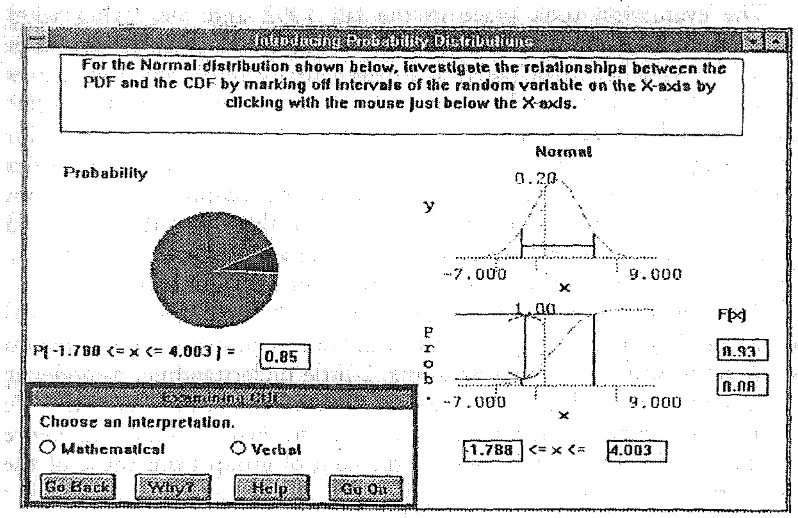

Figure 3

lecture, drill, textbook reading, and educational television. None of these others can be devoted exclusively and uncompromisingly to active experimentation. Carefully designed curricular software can be.

5. Using and evaluating ConStatS

ConStatS is used in almost all the discipline-specific introductory statistics courses taught at Tufts. Even though most students seem able to use the software for an extended period of time without a great deal of direction, most of the instructors provide assignments that provoke open-ended use of the software. These assignments usually contain short essay questions that address the kinds of conceptual issues captured in ConStatS experiments.

During the past year, ConStatS has been at the center of a large scale, FIPSE funded project for evaluating the effectiveness of curricular software. The goal of the evaluation was to assess if ConStatS in particular, and curricular software in general, could help students to develop a deeper conceptual understanding of statistics (or of any other subject matter built around a comparably abstract theory).

The evaluation took place in the fall 1992 and the spring 1993 semesters. Classes at three universities participated. In total, seven classes with 303 students participated as experimental groups, and two classes (from Tufts) with 49 students participated as control groups. Over 200 experimental subjects came from the introductory statistics course for psychology students at Tufts. Other subjects were enrolled in statistics courses for biology, education, child study and economics. The main control group was taught by a member of the team that designed ConStatS. Both control groups used tool-based statistics software for mainframe and personal computers, but did not use ConStatS.

The principal assessment instrument was a 103 item test of conceptual understanding that was derived from ConStatS. Questions were scored on a five point scale: 1-no understanding, 2-little understanding, 3-moderate understanding, 4-good understanding, 5-excellent understanding. We computed average results for each question, and then computed difference scores by subtracting the results of the control group from those of the experimental groups:

- The overall experimental group did better on 82 out of 103 questions
- The Tufts experimental group did better on 89 out of 103 questions

The 10 questions on which the experimental group showed the greatest differential involved transformations (z scores and linear transformations), probability, the concept of deviation, and the sensitivity of summary measures. In many respects these results need to be conditioned on exactly what parts of ConStatS were used in each class, and on the kinds of experiments students performed. The version of ConStatS being used in the evaluation program has a facility that records just how each student interacts with the software, and the specific experiments they do. An analysis of all this trace data from the evaluation is in its infancy and beyond the scope of this paper. However, the comprehension test results, and our ability to connect them with specific screens, have already helped us to add or to redesign 20 screens, and to make a number of small but significant changes. We will repeat the evaluation next year, and learn then if we have improved the software in the way we have intended.

Two other preliminary results are worth noting. On many questions, the patterns of the errors were very revealing. For instance, when students were asked "What is it that makes a variable a random variable?" most students gave one of three wrong answers, indicating specific misunderstandings in the absence of correct comprehension. These results will be of value independently of the software. Secondly, when we scored the experimental group results using three categories (2 or less, 3, 4 or greater), 74 out of 103 questions had bimodal distributions. Students appeared to get either a fairly strong or a fairly weak understanding of most concepts. The trace data, we expect, will indicate which students fall where, and why.