EDITOR:
ROBERT B. DAVIS

The Journal of Mathematical Behavior

SPECIAL ISSUE:

Boxer!

APRIL 1991
VOLUME 10, NUMBER 1

ABLEX PUBLISHING CORPORATION
NORWOOD, NEW JERSEY

THE JOURNAL OF MATHEMATICAL BEHAVIOR
Volume 10, Number 1, 1991

SPECIAL ISSUE:

BOXER!

CONTENTS

JOURNAL OF MATHEMATICAL BEHAVIOR 10, 91–113 (1991)

Learning About Sampling with Boxer

HENRI PICCIOTTO
The Urban School, San Francisco

DON PLOGER
University of California, Berkeley

This article describes a class taught in the summer of 1988, an introduction to probability and statistics for a heterogeneous group of 12 academically talented secondary students. The main focus was on the concepts of sampling and binomial distributions. The approach was based on simulation, including extensive use of the Boxer computer language. We present the work of a group of 3 students who had minimal prior exposure to computer programming. During the course, these students used, modified, and created computer tools to produce a sophisticated simulation. This project demonstrates the value of integrating programming with teaching subject matter.

INTRODUCTION

Probability and statistics are relevant to many areas of mathematics, science, and the humanities. Until recently, these subjects had rarely been taught at the precollege level. Where they were taught, courses in these subjects placed heavy emphasis upon formulas and calculation. In the last decade, a number of educators have recognized the limitations of the traditional approach, and the need to reach more students with the basics of statistical reasoning. They have called for reform of the curriculum in this area. Along these lines, the National Council of Teachers of Mathematics (NCTM, 1981) has been exploring a variety of approaches. Subsequently, a joint venture of NCTM and the American Statistical Association produced the Quantitative Literacy Series (QLS). One book in the series (Landwehr, Swift, & Watkins, 1987) provided the backbone of the curriculum for the course described in this article.

Two features of the QLS have made it attractive to mathematics educators.

We thank the staff and administration of the Academic Talent Development Program, who made this course possible, and were very supportive throughout. We thank Andrea diSessa for many helpful suggestions.

This research is supported primarily by a grant from the National Science Foundation, NSF–MDR–88–50363, to Andrea A. diSessa. We also gratefully acknowledge a contribution to support students form Apple Corporation, and a contribution of equipment from Sun Microsystems.

Correspondence and requests for reprints should be sent to Don Ploger, The Boxer Research Group, Education in Mathematics, Science, and Technology, 4533 Tolman Hall, University of California, Berkeley, CA 94720.

The first is the emphasis on exploratory data analysis, particularly the use of graphic techniques applied to data that is of interest to secondary students. The second is extensive practical experience with simulation. Both these features incorporate recent trends in the field of statistics, as well as distinct pedagogical choices that acknowledge how students construct their understanding. The QLS is very flexible:

- It can be used in Grades 7–12.
- It is appropriate for a broad spectrum of students.
- It can be incorporated into existing courses or form the core of a new course.
- It can be taught with or without the use of computers. (Simple graphics and simulation programs for the Apple 2 are available for use in conjunction with the QLS books.)

The combination of sophisticated content and teacher-friendly format has allowed it to have a significant impact on secondary statistics education.

More recently, the topic of statistics education has been addressed in the *Journal of Mathematical Behavior*. Rosebery and Rubin (1989) noted that the traditional approach to teaching statistics makes the mathematics obscure and does not help students understand real-world applications of statistical reasoning. They favored an approach to the teaching of statistics that emphasizes learning by doing. Rosebery and Rubin (1989) developed a computer-based approach to statistics education, ELASTIC, that is in accord with the general principles of the QLS.

Rosebery and Rubin's use of computers goes much further than QLS. It includes a number of sophisticated tools which students can use to explore the underlying meaning of important statistical concepts. For example, one such tool, called Stretchy Histograms, allows students to manipulate the shape of a hypothetical distribution represented as a histogram and observe the changes in the measures of central tendency. Such manipulation leads to deeper understanding of the median and the mean, as well as to a better grasp of the meaning of histograms.

Results of research using ELASTIC demonstrate that a wide range of students gained greater mastery of essential concepts such as distributions and linear regression. Students with poor math backgrounds were able to practice statistical reasoning more readily than in traditional approaches because there was less emphasis on formulas. A very important benefit was that many of these students demonstrated an improvement in their mathematical skill, notably in proportional reasoning. The materials were also useful for mathematically sophisticated students, who were able to delve deeper into the statistical concepts and appreciate better that mathematical concepts have applications outside the classroom.

The ELASTIC software runs on the MacIntosh Plus, a machine that is relatively inexpensive and widespread. Teachers can easily learn to use the system, and the time investment for students in learning how to use the software is minimal.

There are, of course, other ways to apply computers to the teaching of statistics. In this article, we present one such possibility using Boxer: a computational environment that has the capability of a complete programming language. With Boxer, a new view of tools is possible, which will be detailed below.

Prior work in Boxer included a case study of a 12-year-old learning about probability by modifying teacher-written simulations (Ploger & diSessa, 1987). That study demonstrated the feasibility of learning statistical subject matter and Boxer programming in a mutually reinforcing manner. The case study was, however, far less ambitious than a complete course. A first attempt at such a course resulted from the collaboration of a math teacher and curriculum developer (HP), and a Boxer researcher (DP).

DESIGNING THE COURSE

As a teacher of mathematics in elementary and secondary schools, HP has been writing software for his classes since 1977. At first, he used BASIC to write programs that helped his students learn arithmetic and develop their number sense (Picciotto, 1987). Later, he used Logo to create games and tools to help teach high school mathematics (Picciotto, 1989, 1990). Logo is the prominent example of a programming language designed for education (Papert, 1980). Logo was effective in allowing the teacher to develop lessons using computer tools, but our experience is that it stopped short of allowing students to inspect or modify easily the teacher's tools. This is because Logo procedures can neither be easily executed step-by-step, nor is there an easy way to access relevant portions of a procedure. Moreover, a necessary aspect of handling data in Logo is the use of recursion, a very difficult concept for beginners to master.

In the past, HP had offered a traditional probability and statistics class at the high school level, and had gradually moved towards the approach proposed in the QLS materials. In particular, he had taught the sampling unit (Landwehr et al., 1987). In that unit, students develop a feel for the concept of sampling by taking random samples from known populations with a binomially distributed random variable ("yes–no" populations) and analyzing their distribution. Students experience firsthand the impact of different sample sizes and population percentages, laying the groundwork for a solid understanding of the concepts of the law of large numbers, binomial distributions, and confidence intervals, based upon their own experience. Part of the course involves the building of box plots to represent the middle 90% of a distribution. Tables of such box plots are used to evaluate confidence intervals for a given sample proportion and sample size visually.

In the context of this work, he heard about Boxer for the first time. Boxer was strongly influenced by Logo (diSessa, 1986), and its designers' pedagogical pronouncements on tool building (diSessa, 1988) were very much in line with the HP's last work with Logo tools. HP was offered the opportunity to create a summer school class in the U.C. Academic Talent Development Program where he could use Boxer to teach probability and statistics.

The course was to last 6 weeks, two sessions per week. Each session lasted 3 hours in the morning; students had access to an open Boxer computer lab in the early part of each afternoon. HP organized much of the course around the QLS sampling unit. The book's emphasis on an intuitive understanding based on simulation seemed to lend itself to the approach he wanted to take: using computer experiments to develop statistical insight. He decided to enrich the hands-on simulations beyond the limited coin flipping and random number table experiences presented in the book by designing a number of additional hands-on experiments. These simulations were to provide a transition to student-designed computer simulations in Boxer, the key innovation in the course. In addition to work with hands-on and computer labs, students were to apply what they had learned about sampling to run their own surveys on questions of their own choosing.

DP's role was to contribute a general perspective on how to integrate Boxer programming into the course. He wrote some essential Boxer tools and showed students how to use and modify those tools. He also showed students how to assemble those tools to create simulations of probability experiments. In addition, DP helped HP and his students to debug their programs.

One component of the course, which went beyond the textbook, was the theoretical-mathematical understanding of binomial distributions. HP designed a discovery-based curriculum for the building of those concepts from the foundation laid by the empirical work developed in the text. This approach, which cannot be detailed here, led to much more conceptual clarity on binomial distributions, Pascal's triangle, and so on, than the conventional coverage usually does, mostly because the students had plenty of time in the experimental and simulation part of the course to develop the necessary intuitions. Having discovered the formula for binomial distributions, the students were able to understand how the 90% box plots of a given distribution could be found by computation (as opposed to simulation.) With the help of Boxer, HP got the students to expand the 90% box plot tables (presented in the book for four possible sample sizes, and population percentages in 5% increments) to any sample size up to 160, and any population percentages. An indispensable tool for this was a Boxer PROBABILITY CALCULATOR, written by HP. Students used this tool to create an expanded set of tables to find confidence intervals for the "real-world" surveys they conducted at the end of the course.

TEACHING THE COURSE

The reality of teaching a class with a prototype of a computer language did not quite match the utopian scenario HP had mapped out in his mind. Boxer requires state-of-the-art Sun workstations, and at the time, not enough were available. Furthermore, the language in its mid-1988 implementation had many irritating bugs and there were inexplicable system crashes. This is, of course, normal for any software under development. (Note: The current implementation is much more complete and robust.) A more significant problem was that the class was very heterogeneous. The students were aged 12 to 16. Some had no programming experience and no algebra in their background, others had had 2 years of algebra and many years of programming experience in several programming languages. Nevertheless, the class was surprisingly successful, and teaching under these challenging conditions allowed HP and DP to put Boxer to the test.

To simulate the random sampling of a yes–no population, students picked marbles from an urn (yellow = yes), rolled 10-sided dice (e.g., 0, 1 or 2 = yes), used spinners, and so on. The teacher then asked the students to simulate each process in Boxer and integrate the simulations into a SAMPLER program. The transition from a real-world experiment to a Boxer version was remarkably easy compared with doing the same thing in Logo or BASIC, perhaps because variables and graphics were visible at all times. However, this SAMPLER program still proved to be much too difficult for beginners to write, and two out of four teams failed to complete it. For them, this was rather frustrating.

It became clear that although Boxer was easier to learn than other languages, programming is a difficult discipline that cannot be picked up on the fly in a couple of weeks. In a 6-week course, there was little time for explicit Boxer instruction, and no prepared instructional materials. (Note: There are now five on-line tutorial units.)

A SPINNER program, written by HP, worked well, but many students did not really understand its most crucial procedures. HP felt that the situation was no different from the one he had experienced at his own school with Logo on the Commodore 64s: Everyone used the tools profitably, but only the programmers among the students were able to take them apart or build their own (Picciotto, 1990).

As a result, HP rearranged the student Boxer teams to be more homogeneous and asked the students to create Boxer projects at their own level of programming expertise, using the hypertext capabilities of the language to come up with a suitably rich result and feeling free to borrow procedures from teacher-written programs and from one another. The PROBABILITY CALCULATOR served as a sample of how they could present their finished project by judicious organization of text, procedures, and boxes. He suggested four topics, and to his surprise, all his suggestions were implemented, each one by a different group. The four projects were:

- A probability tutorial, where one could compare theoretical and empirical results of various experiments. The metaphor used was of marksmen of different skills shooting at a target.

- A simulation of a game of tennis to illustrate a real-world example of a binomial distribution.
- An extension of the original SAMPLER program to include the construction of box and whiskers plots for sampling distributions, in order to help calculate confidence intervals (following closely the approach given in the textbook).
- A random walk, simulating the canonical sampling experiment (detailed in the following).

THE SPRITE'S RANDOM WALK

The projects were quite successful, reflecting a solid understanding of the subject matter, and a good grasp of Boxer. It was not surprising that the experienced programmers did a good job. They were the sort of students who would have done well in any language, and who had already proved their programming skill in the early phase of the course.

A stunning achievement was the simulation of a random walk experiment by one group of 3 students, all of whom were definitely beginners. They had very

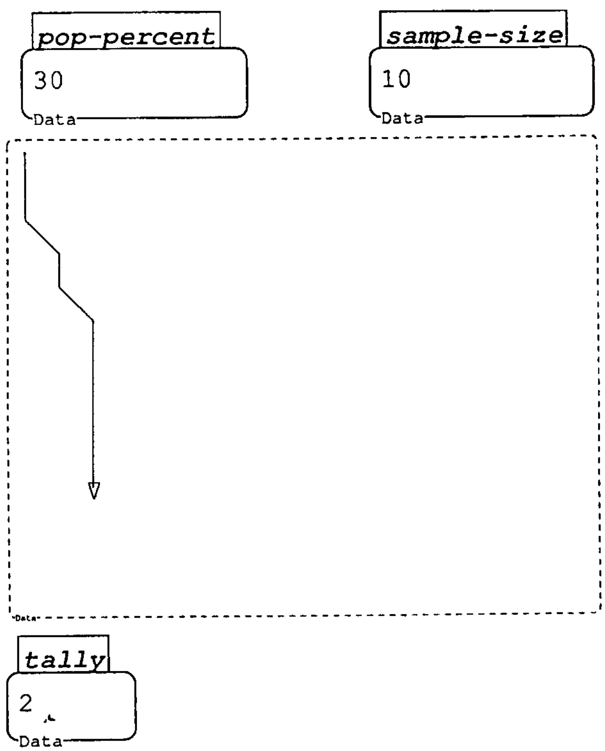

Figure 1. A graphical representation of a sample of size 10.

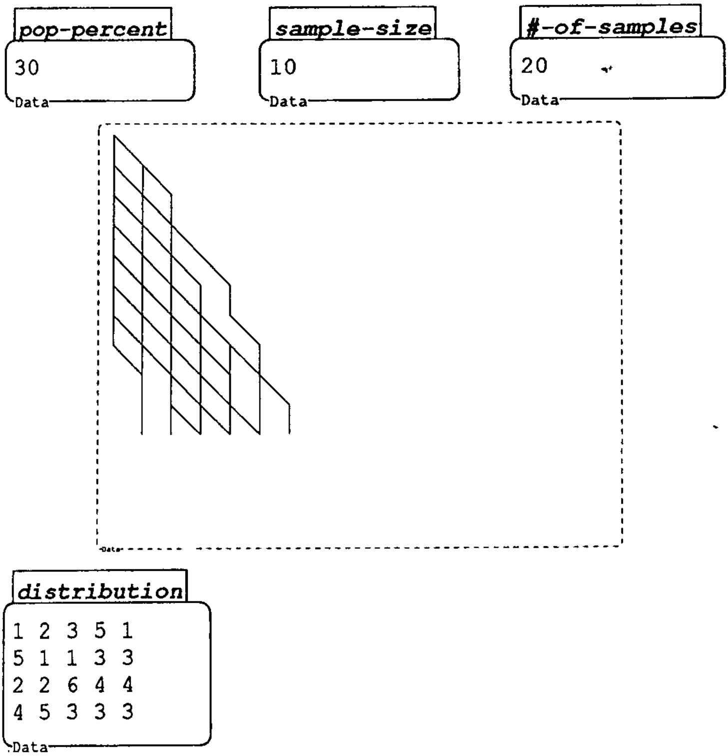

Figure 2. A graphical representation of 20 samples of size 10.

limited prior exposure to programming: 1 had done a little work in Logo, 1 had a little BASIC, and the third had never used a computer. Together, they created the program by using some of the previous work that had been done on sampling, which included a researcher-written sorting routine. Following HP's suggestion, they designed a random walk for the turtle in a graphics box. The turtle would go straight down or diagonally, depending on the outcome of asking a random member of a simulated population for a yes–no answer. Figure 1 shows the graphical representation of a single sample. The data box in the upper left indicates that the population percent is 30. The box to its right indicates that the sample size is 10. The data box, **tally,** indicates that there were a total of two *yeses* in the sample. The graphical display indicates that there are two diagonal lines in the sprite's path, one for each *yes* (a sprite is Boxer's generalization of the Logo turtle).

Figure 2 shows the next stage of the program. A total of 20 samples of size 10 were drawn from the same population as before (30% *yes*). This repeated sampling results in 20 paths that follow the design of Pascal's triangle and provide a dynamic visual model of sampling. A box, **distribution,** stores the results of the

20 samples. When this much was worked out, DP suggested that at the end of its walk, the turtle could drop a "marble." This was implemented by the students, and the end result was a histogram which dramatically displayed the distribution of samples (Figure 3). The databox, *beads*, is a frequency table that was computed by the students' program. There were two samples with zero *yeses*, two with one *yes*, three with two *yeses*, and so on. These results are consistent with the sorted distribution.

To summarize, although the key ideas behind the program were suggested by educators, and the most difficult pieces of code were borrowed from the work of more experienced programmers, the scope of the project and every step of its implementation were determined entirely by the students themselves. Clearly, these beginners had picked up a lot about Boxer and programming in the first half of the course, when they had seemed to be floundering. A program of this sophistication could not have been achieved in this short a time in any other

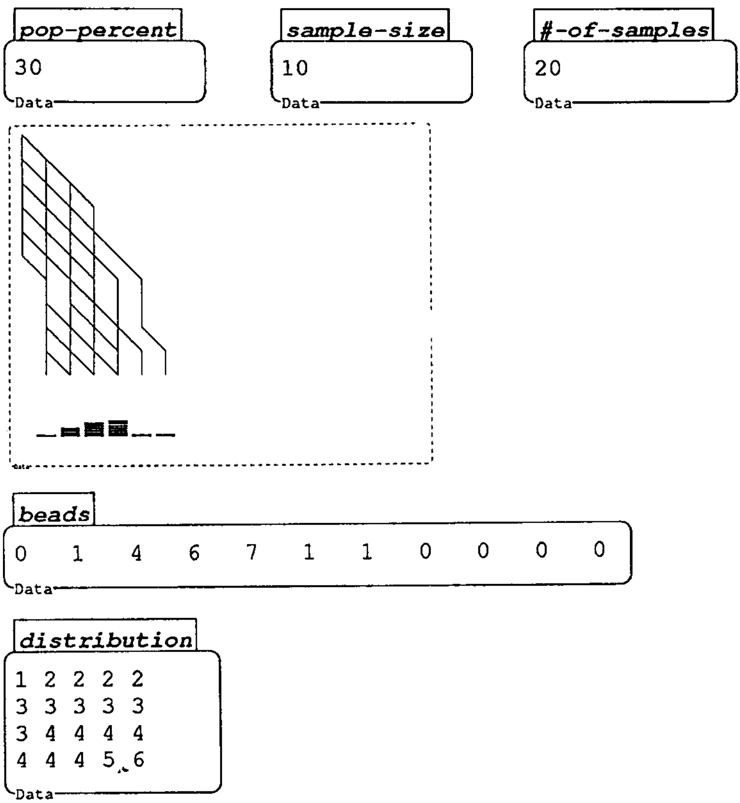

Figure 3. A histogram produced for 20 samples of size 10.

language, by people with as little experience as they had. What made it possible seemed to be five main factors in the design of Boxer:

- The incorporation of Logo's turtle graphics as a subset of the language;
- The fact that Boxer behaves according to the students' intuition of what to expect;
- The ease with which one can borrow entire chunks of code from other programs—a consequence of the natural modularity of the language and the fact that the computers were networked;
- The ease of inspecting any procedure (all you have to do to get at it is open the box it is in);
- The ease of debugging: to see the effect of a line of code, you simply execute the line.

Other factors which contributed to the success of the project were:

- The group's good dynamics;
- Their strong understanding of the concepts in probability that they illustrated—a consequence of the abundance of concrete experiences early on, and of the guided discovery of the theoretical concepts later;
- Availability of advice from various Boxer experts who were eager to help.

The students' felt a sense of achievement when finished with the project. They had created a programming model for the process of repeated sampling, and they extended this model to represent a frequency distribution. They knew they could not have completed it without a solid grasp of the material. Conversely, their understanding was reinforced by the construction of a computer model.

DISCUSSION

Our approach is consistent with Rosebery and Rubin (1989), particularly their emphasis on the primacy of statistical reasoning over formulas and computations and their reliance on student-directed learning. In particular, we are in accord with their emphasis on greater use of "tools" by students. Our view of tools, however, is different. In ELASTIC, tools are defined within MacIntosh applications, and the software creator has the final word in the design of the tool. Boxer, although requiring a more expensive delivery machine, is a complete programming environment. Once students learn to program in Boxer, they can modify existing programs and they can write their own programs (which is not possible in ELASTIC). For example, in ELASTIC, students have a great deal of flexibility in experimenting with histograms, but they cannot build the representation itself. In Boxer, on the other hand, students can build their own histogram, as well as many other tools.

In fact, there is no reason why the tools available in ELASTIC could not be programmed in Boxer. Curriculum creators can use Boxer for the quick development of tools, games, and microworlds. Such software would allow much of the same work to take place in the classroom as is possible with a package such as ELASTIC. However, teachers would be able to inspect, modify, and/or enrich the package, and the same possibilities would also be open to students. In Boxer, teachers and students have far more independence than with "canned" software: They can do things we cannot even predict.

The price of the added flexibility available in Boxer is a need for more teacher and student training. Boxer is easier to learn than other computer languages. But programming is a nontrivial skill, and extended work with Boxer would probably yield far greater benefits than were conceivable in this 6-week course. Such a commitment of school resources would follow from the realization that the ability to program representations or simulations in probability and statistics, or in mathematics and science more generally, is a powerful way to extend and deepen one's understanding. In this perspective, the use of "black box" software, although still appropriate for some purposes, is deemphasized in favor of inspectable teacher- and student-created tools.

CONCLUSION

One crucial pedagogical lesson of this course is a better understanding of how much is gained when students have the flexibility to integrate prewritten tools with the results of programming their own. We reject both the belief that programming is best left to professionals, and the opposite one that students should program everything from scratch. The appropriate tool may be too hard for the student to build at a given level of programming experience, and in the given time frame. But tool building in the form of computer programming is a powerful learning experience. The students come to understand better their own ability to manipulate computer representations, and appreciate the role of computer programming in a variety of uses: modeling, simulation, graphics, and computation.

The proper balance of curriculum-developer, teacher, and student programming depends on many factors, one of which is the priority of teaching subject matter versus teaching programming in a given class or lesson. This is not just a philosophical question, but often a practical one, whose answer is often determined by school reality. Open-ended student projects allow an opportunity for major advances in both areas.

The beauty of Boxer is that teacher-written tools, as well as tools written by other students, invite inspection. The fact that Boxer is a wide-open language makes it possible for students to appreciate the tools at a level appropriate to them. By the end of this course, most of the groups in our class had taken a look at the workings of the sorting routine we had supplied them. This kind of analysis of tools is just not possible with software written in a compiled language, or even in an interactive language like BASIC or Logo, where programs are difficult to inspect. In Boxer, one can always backtrack and have students analyze the workings of an educator-written tool, or have them reconstruct it from scratch when they are ready for that and time allows.

It appears that Boxer dramatically extends the usefulness of computers in teaching mathematical sciences. It builds on what was beginning to be possible in Logo, and can potentially incorporate the progress made in the design of educational software of all types. In Boxer, educator-written tools and student-programming efforts both pay off. The inviting nature of the language maximizes the synergy between the two approaches.

REFERENCES

diSessa, Andrea A. (1986, July). From Logo to Boxer. *Australian Journal of Computer Education, 8–15.*

diSessa, Andrea A. (1988). *Social niches for future software* (Boxer Tech. Rep. G-3). Berkeley: University of California.

Landwehr, James M., Swift, James, & Watkins, A.E. (1987). *Exploring surveys and information from samples.* Palo Alto, CA: Dale Seymour.

National Council of Teachers of Mathematics (1981). Teaching statistics and probability. In A.P. Shulte & J.R. Smart (Eds.), *1981 annual yearbook.* Reston, VA: Author.

Papert, Seymour (1980). *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books.

Picciotto, Henri (1987). *Guesstimation* [Computer Program]. Danbury, CT: E.M.E. Corporation.

Picciotto, Henri (1989). *Logo math: Tools and games* [Computer Program]. Portland, ME: Terrapin, Inc.

Picciotto, Henri (1990). Teacher-created educational software: Logo tools. *Logo Exchange, 8,* 18–20.

Ploger, Don, & diSessa, Andrea A. (1987). Rolling dice: Exploring probability in the Boxer computer environment (Boxer Tech. Rep. E-1). Berkeley: University of California.

Rosebery, Ann S., & Rubin, Andee (1989). Reasoning under uncertainty: Developing statistical reasoning. *Journal of Mathematical Behavior, 8,* 205–219.