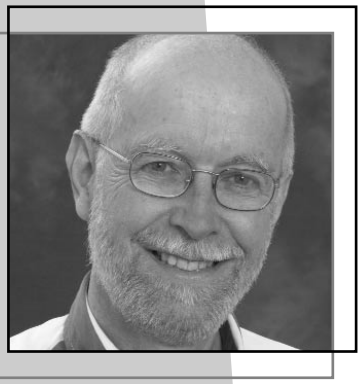
Megabytes and Colour, but Learning is Still the Issue



ABSTRACT

In the first issue of Australian Educational Computing, in 1986, Cumming and Abbott reported a controlled comparison of top-down and bottom-up teaching strategies for Grade 5 and 6 students' use of a simple logic programming language. They found that both strategies were rated highly by students and teachers, and gave useful learning; and that the top-down approach was especially effective. In the modern classroom children now have wonderfully powerful hardware, and highly flexible and impressive software. However the fundamental questions about teaching and learning strategies have changed less with the years, and careful experimental research on these strategies is still needed.

GEOFF CUMMING

School of Psychological Science

La Trobe University

BIOGRAPHY

Geoff Cumming's research interests have combined learning and computing. He has studied young children learning to read, the use of logic programming in middle school (The Prolog project), artificial intelligence and intelligent tutoring systems, and graphical simulation software to help understanding of statistical concepts. He has been at La Trobe for many years, following postgraduate study in experimental psychology at Oxford, which followed a first degree in psychology and statistics at Monash. He has three children, now scattered around the world, and enjoys mountain-biking, wood-working and word games.

THEN

It's 1986. Elizabeth Abbott and I are working with Grade 5 and 6 students and teachers at St Clare's in Thomastown West, near La Trobe. The students are crowded together, two to a computer. The computers are Microbees (machines assembled in Australia!), with clunky orange characters glowing on a dark screen, 64K of memory and disk drives that take 5_ inch disks that really are floppy! The first Apple Macs were a year or two old, expensive, and yet to make an appearance in schools. We had only line editors, no mouse was to be seen. Hard disks, networking, email and the Internet were all in the distant future.

There was excitement but also apprehension about computers in schools. The public and political imperative was that students should learn to use computers so they could find jobs in the highly automated world of the future. Remember the Japanese 'Fifth generation' project, intended to leap-frog Japan directly into that scary world? Many teachers did not want the extra headaches that computers seemed to bring, but many also saw glimpses of great possibilities to expand what students could do. Seymour Papert from MIT, one of the creators of the Logo language, visited Melbourne and sparked headlines like 'Expert predicts death of schools'. 'Computer as pencil!' was one of his slogans, and it's hard to quibble with that these days: Simply do what you want to do—this probably involves a computer, but you might hardly notice that it does!

Papert argued that giving children computers with a flexible tool like Logo would let them discover all sorts of great things, and develop all sorts of higher abilities, including planning, finding out, collaboration—oh, and perhaps maths as well along the way. He seemed to give the impression that all these good things would simply happen, although his famous book *Mindstorms* (Papert, 1980) did say that teacher guidance is needed. Educational computing leaders in Australia were very much up with the international game, and contributing to worldwide discussions about Logo, computers for learning, and the support teachers needed. Anne McDougall and other locals wrote books (for example McDougall, Adams, & Adams, 1982) that were found useful and practical in classrooms around the world, and helped numerous students use Logo to write procedures, draw stunning spirals-within-spirals, compose strange poetry and even tinker with recursion.

On sabbatical in 1982-83 I had worked with the group in Edinburgh who were studying the cognitive effects of using Logo in the classroom, and I also visited a London group using Prolog with children. In the arcane world of AI (artificial intelligence) programming, Lisp has long been the mainstay, but some scientists were advocating 'logic programming' as an interesting alternative, and had developed the language Prolog for this purpose. Logo was a humanised version of Lisp, which retained much of Lisp's power while providing a friendly syntax. Logo also offered Turtle Geometry—which allowed you to draw a square on the screen by giving a sequence of commands as simple as

REPEAT 4 (FD 10 RT 90)

(where FD = forward, RT = right turn), just as you might instruct a robot to dance a square on the floor.

The London group wanted to do the same for logic programming. They developed what they intended to be a humanised version of Prolog, called Simple. I brought this back with me, and by 1984 had secured Australian Research Council support for our study of how students in middle school might benefit from working with Prolog.

Prolog in the classroom

We quickly identified our two biggest challenges. First was the need to offer guidance to teachers and students. We could give nice demos of using Prolog to build little databases on any topic, then use a database to answer queries. The databases contained rules as well as facts, and so Prolog could use simple reasoning to answer your question, then explain its answer! Nice demos, but what would the students actually do? How would they start? What guidance would they need? Second, we wanted students to be thinking about the topic area and the reasoning—not the computer tool—but even the Simple syntax was in practice too complicated.

Our response to the first challenge was predictable and hardly novel: We wrote worksheets, but we did go one step further by trying out various different teaching strategies in our trial worksheets. Being psychologists we set up a controlled experiment to assess two teaching strategies that struck us as particularly interesting: top-down and bottom-up approaches. Our report of that experiment was published in the first issue of *Australian Educational Computing*, in July 1986 (Cumming & Abbott, 1986). For more than six months one Grade 5 and one Grade 6 class worked for two periods every week using top-down worksheets to query little Prolog databases that we had built, in a range of topic areas, then to look inside them and modify them, until eventually the students were building their own. So the top-down students started with whole databases, then later learned how to use the logic programming language. Another two classes worked with bottom-up worksheets, which guided the students as they assembled information and built their own databases, and queried these, developed them further and eventually went on to use larger databases that we provided. So they started with the small details of using the language, then later used whole databases that were built by themselves or provided by us. In both cases we tried to focus on content and learning activities rather than details of the computer language. Our worksheets first provided detailed hand-holding, then gradually encouraged students to make choices for themselves and work on projects of their own design.

We used a range of measures to assess what the students learned, and of course we asked the students and their teachers lots of questions about using Prolog and the particular teaching strategy they had seen. There was general enthusiasm, and the teachers claimed they could see in the students' work in other classes lots of benefits of the Prolog activities. Our test results suggested that both groups learned a lot, and that the top-down students did distinctly better on a range of measures than did the bottom-up students. We found evidence that, at least in our situation, a top-down teaching strategy seemed to be more effective than a bottom-up strategy.

Our response to the second challenge was to shift from Simple to Mitsi, an improved front-end to Prolog, and to extend Mitsi ourselves to improve its usability even further. In the Mitsi syntax, a database contained facts, such as:

Jose has recipe.

Jose likes eating.

It also contained rules, expressed by using variables, which were any word starting with **some** or **any**, such as

someone makes cakes if someone has recipe and someone likes eating

You can then query the database by asking, for example

someperson makes cakes?

to which Prolog would reply

YES Jose makes cakes

and then, if we followed up by asking **why?**, we would get

Jose makes cakes because

Jose has recipe and

Jose likes eating

A Mitsi database could contain dozens of facts and a dozen or more rules, and could give answers and explanations that seemed quite informative and helpful. There was also an **is told** feature that led Mitsi to ask the user for a fact that was missing but seemed to be needed to answer a query. This feature allowed us to build small rule-based expert systems that could, for example, ask you a series of questions as it sought to diagnose what was wrong with your bicycle, and then advise you what was needed to fix the problem.

The importance of having an extremely easy-to-use front end, and the enhancements we made to Mitsi, were described in our second article in *Australian Educational Computing* (Abbott & Cumming, 1987). In a separate experiment we were able to assign students to form two matched classes, then we compared half a year of Prolog activities with the same time of other computer activities not involving Prolog. We found that the benefit of the Prolog experience generalised, in that the Prolog class did distinctly better on tests of reasoning and reading comprehension than did the class having other computer activities. We used all we had learned in the classroom to guide development of our published software and workbook *Knowledge Tools: Using the Computer for Learning* (Cumming, Abbott, & Briggs, 1993).

NOW

Prolog in retrospect

Looking back it seems astonishing how far the students, the teachers, and we, could go with such limited and awkward computing facilities. Working with Prolog for six months meant that any benefit of mere novelty had certainly worn off. Text only, line editors, crude mono screens: What an archaic world

that seems! Our extended, 'friendly' interface, as illustrated above, seems so awkward and limited. Even so, using Mitsi to carry out projects to organise information, working with simple databases, and having the computer answer questions and give simple explanations was sufficiently interesting and educationally valuable to give strongly positive results, and to yield improvements in general abilities. Some of our Mitsi programs were justifiably described as simple expert systems, and gave advice that could even be useful!

It is interesting to revisit the old discussions about how best to use computers for learning. So much of it seems to centre on fads and fashions! We were right to focus on choice of teaching strategy and to compare and assess the educational value of different strategies. Teaching strategy remains an important issue, and our results are still worth considering, even though our interface and syntax were quickly and convincingly superseded by vastly improved interfaces and more powerful, flexible software applications—including relational databases, expert system shells, and hypercard tools.

Recent investigations, of course, use much richer software tools, as well as networked computers with mouse, megabytes and colour, but our experiments were larger and better controlled, and ran for longer than many modern studies. They also used a fairly wide range of quantitative and qualitative measures.

Recent research on teaching strategies

Teaching strategies are still, quite rightly, being researched. Consider two recent high quality examples, both of which of course used modern personal computers with high resolution colour displays, graphical presentations, and mouse and keyboard for student input. Swaak, de Jong, and van Joolingen (2004) compared two approaches to studying elastic and inelastic collisions. One group used a graphical simulation, with a discovery learning approach. The other group worked with a hypertext learning environment, diagrams and a more expository approach. Both groups completed many assignments. As expected, the expository group did better on a test of definitional knowledge. It was predicted that the discovery group would do better on a test of intuitive knowledge, but if anything the expository group did better on this test also. This was a careful and detailed study, and gives guidance as to the particular and somewhat limited situations in which discovery learning is likely to be the best approach. Its contrast of discovery and expository learning is reminiscent of the free vs. structured dimension of teaching approaches that was discussed by Cumming and Abbott (1986): A similar question in a modern, more sophisticated form.

Zhang, Chen, Sun, and Reid (2004) adopted a discovery learning approach with simulation-based learning activities, and studied in detail the types of guidance and explanations that were most effective in supporting learning. They identified three distinct types of learning support that are effective in enhancing various aspects of learning, within a discovery framework.

Overcoming statistical misconceptions

My own research continues to be on how computers can help learning, but now I study how interactive graphical simulations can help students and researchers overcome deeply ingrained misconceptions about some fundamental statistical concepts. My software is ESCI ("ess-key", *Exploratory Software for Confidence Intervals,* www.latrobe.edu.au/psy/esci), which runs under Microsoft Excel. I give my students worksheets to guide at least their initial learning activities with the simulations. Cumming and Finch (2005) give advice about using confidence intervals that is based on our research.

The cognitive rationale for this work was described by Thomason, Cumming, and Zangari (1994) and is based on the notion of conceptual change. When misconceptions are deeply ingrained, it is not sufficient merely to be told about the problem and given a description of the correct concept. It seems necessary to confront the misconception with vivid, dramatic demonstrations of the error, and give the student ways to interact with simulations or concrete objects that display the correct concept. It seems to be important to have multiple ways to experience the correct concept and its consequences. My interactive ESCI simulations are designed to give multiple representations of statistical concepts, and to allow learners to carry out a number of demonstrations or experiments to explore the target concepts and their consequences. I try especially hard to provide memorable pictures of the concepts that students can take away as vivid mental images, to serve as mental 'hooks' or mnemonics for their thinking about the concepts.

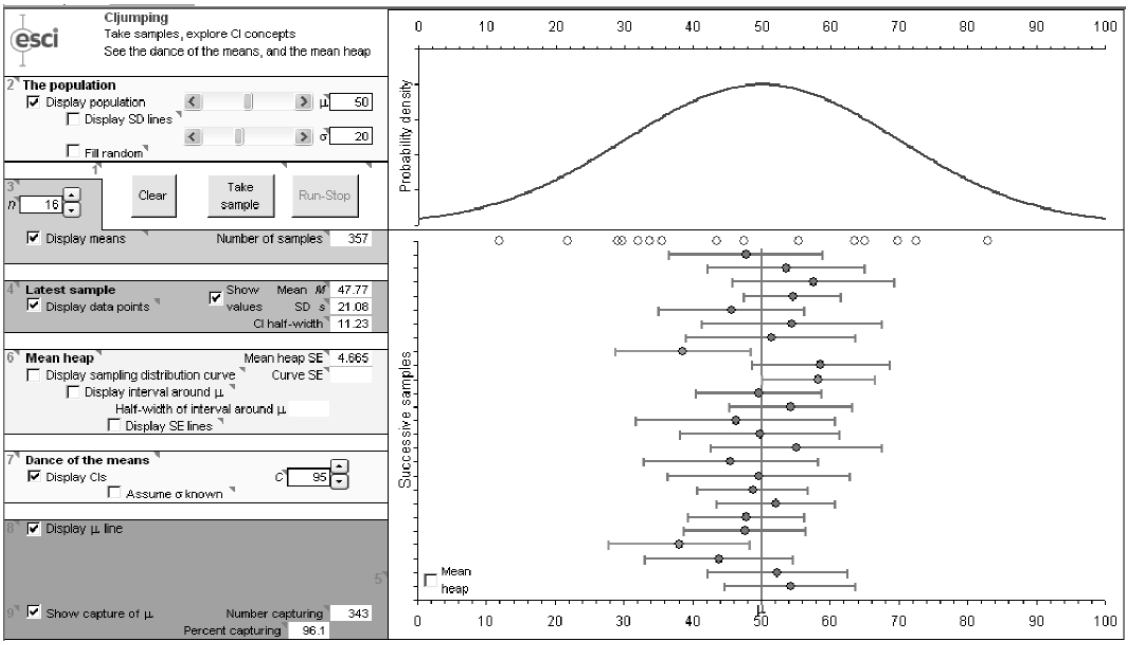
Figure 1 shows the ESCI simulation that displays a sequence of random samples that have been taken from a normally distributed population. People have a very strong tendency to severely underestimate the extent of sampling variability: We seem to think that future samples will be quite like the first sample we took. Seeing the simulation running, and the sample means 'dancing' down the screen, and noting the considerable variation from one to the next confidence interval, may provide a basis for activities that can overcome the misconception and allow students to build more accurate intuitions about the extent of sampling variability.

Alongside my work developing ESCI and using it with students in their introductory statistics course, I and colleagues carry out cognitive experiments to describe the statistical misconceptions that students and researchers have. Eventually we will repeat these experiments after students have worked on a range of learning activities—many of them using ESCI—to assess the extent to which the misconceptions have been replaced by correct understanding.

THEN AND NOW: CONCLUSIONS

Over 20 years there have of course been dramatic developments in the hardware and software facilities that are commonly available for students in the school classroom. Numerous megabytes of memory, and wonderfully detailed colour displays are just part of the story. By contrast, our research progress on

Figure 1



understanding learning processes, and the teaching strategies that are most effective for helping students make the most of their computer tools, has been more limited. We still need to think about, and study, top-down and bottom-up teaching approaches, and discovery and expository approaches, if our students are to get the best from the wonders that modern computers offer. Now, as back in 1986, it should be careful controlled research that gives important guidance as we choose the teaching and learning approaches to use with our students. That beats mere fashion any day!

Figure 1.
A picture of the ESCI simulation that allows you to take an indefinitely long sequence of random samples from a population. It displays the 95% confidence interval calculated for the mean, for each sample. The population is shown at the top. Controls (sliders, buttons, and check-boxes) are at left, as are reports of various values. The row of small circles mark the values of the 16 data points making up the most recent sample. The most recent 25 sample means, with their confidence intervals, appear below. The three intervals shown in a lighter shade do not capture the population mean m, indicated by the vertical line. The key insight is that we expect, in the long run, that 95% of these intervals will capture m. In the picture, 343 samples have been taken, and 96.1% of the intervals have captured the population mean. One important message is that there is a surprisingly large amount of variability from sample to sample.

REFERENCES

Abbott, E., & Cumming, G. (1987). Making front-ends friendly: The case of Prolog. *Australian Educational Computing,* 2(1), 26-32.

Cumming, G., & Abbott, E. (1986). It's the educational strategy that matters, even if the language is Prolog. *Australian Educational Computing,* 1(1), 34-39.

Cumming, G., Abbott, E., & Briggs, J. (1993). *Knowledge tools: Using the computer for learning.* Hawthorn, Vic: Australian Council for Educational Research.

Cumming, G., & Finch, S. (2005). Inference by eye: Confidence intervals and how to read pictures of data. *American Psychologist,* 60, 170-180.

McDougall, A., Adams, T., & Adams, P. (1982). *Learning Logo on the Apple II.* Sydney: Prentice-Hall.

Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas.* Brighton, Sussex: Harvester.

Swaak, J., de Jong, T., & van Joolingen, W. (2004). The effects of discovery learning and expository instruction on the acquisition of definitional and intuitive knowledge. *Journal of Computer Assisted Learning,* 20, 225-234.

Thomason, N., Cumming, G., & Zangari, M. (1994). Understanding central concepts of statistics and experimental design in the social sciences. In K. Beattie, C. McNaught & S. Wills (Eds.), *Interactive multimedia in university education: Designing for change in teaching & learning* (pp. 59-81, and subject of commentary pp. 99-102). Amsterdam: Elsevier.

Zhang, J., Chen, Q., Sun, Y., & Reid, D. J. (2004). Triple scheme of learning support design for scientific discovery learning based on computer simulation: Experimental research. *Journal of Computer Assisted Learning,* 20, 269-282.

AUTHOR NOTE

This research was supported by the Australian Research Council.

ESCI ("ess-key"; Exploratory Software for Confidence Intervals), which runs under Microsoft Excel, is available from www.latrobe.edu.au/psy/esci

Correspondence about this article may be addressed to
Geoff Cumming,
School of Psychological Science,
La Trobe University, Australia 3086. Email:
G.Cumming@latrobe.edu.au