USING DATABASE SOFTWARE IN UNDERGRADUATE SURVEY COURSES

John F. McClymer
Assumption College

Computer-assisted instruction (CAI) has already achieved buzzword status in American education. A revolution in teaching and learning is occuring all around us. Or, at the very least, one is about to occur. Such is the conventional wisdom. And wisdom it may well prove to be. Not only does the computer perform arithmetic feats, it also enables the user to model complex processes and to gain access to, and control over, vast bodies of information. What is more, the technical capabilities of both hardware and software continue to expand while the cost continues to decline.

All of this makes it plausible that computers might be the most important educational devices to come along since the advent of printing. You could not prove it, however, by examining current undergraduate history instruction. One measure of just how little use historians have found for computers in the classroom lies in the nature of the historical software textbooks publishers have decided to market which, for the most part, consists of test banks. More convenient to use than printed collections of multiple choice and matching column questions publishers have traditionally provided, such test materials cannot, and are not intended to, have a substantial effect on either teaching or learning.

Some faculty, impatient with the reluctance of publishers to sponsor more innovative software, have forged ahead on their own. One popular idea is to "soup up" the standard test bank by designing short-answer tutorials. In these students answer randomly selected questions (usually in multiple choice format since allowing students to fill in blanks requires expanding the program to recognize correct but misspelled answers). The program typically keeps score, tells students where to find the answers to questions they got wrong, and either congratulates users on their burgeoning historical knowledge or urges them to study harder in the future.[1]

A second approach that requires a greater knowledge of programming techniques is to design games and simulations. These recreate some historical situation or event and then challenge the students to figure out how actual or hypothetical historical figures responded, or might have responded, in those circumstances. Simulations engage students' competitive interests by affording the opportunity to "win" while they learn.[2]

More ambitious still are attempts to use so-called authoring languages to monitor and critique the ways students read texts or other materials. The idea is to use the machine to check on how students are, or are not, making sense of the information they are encountering in the course while they are in the process of reading and/or thinking. Projects of this sort involve considerable investments of time and money since program designers have to routinely expand the program to recognize the endlessly creative ways students have of (mis)interpreting even textbooks.[3]

Most history teachers, to the extent they have thought about these and other attempts to use computers in undergraduate instruction at all, have adopted a posture of watchful waiting. Sooner or later, they assume, one or another of the pioneering attempts to develop software specifically for undergraduate instruction might prove out. In the meanwhile, they will continue to teach their courses as they always have.

Historians do not have to wait. A growing number are finding that they can use programs designed for other purposes. This is the case with statistical packages. Teachers have incorporated the computer into courses on quantitative methodology by such interactive software as Minitab and SCSS.[4] Nor is number-crunching the only viable use. Richard W. Slatta has described how he teaches non-quantitative research techniques by using database management software on microcomputers.[5]

Database managers can be employed in surveys as well as methods courses. Many of the basic questions around which we organize our American and western civilization courses are implicitly or explicitly quantitative. We may not employ elaborate statistical procedures in their study or explication, but we often base analyses upon numerically verifiable differences between revolutionaries and loyalists, say, or between McKinley and Bryan voters. Significant segments of our narrative accounts, to the extent that they are sound, rest upon large data sets.

Further, as recent textbooks demonstrate, the survey now typically incorporates a good deal of social history.[6] This material represents some of the most exciting research of the last several decades, but it is often singularly difficult to teach. In part, this is because students cannot imagine easily how social historians proceed. They can picture how one would read the Lincoln-Douglas debates but not the census of 1860.

Database management software (DBMS) provides undergraduates with the opportunity to explore these staple questions of our survey courses by enabling them to examine the evidence in a manner analogous to the way collections of documents allow them to analyze literary evidence. This has two major advantages. One is that it engages the student actively in the process of figuring out the meaning of the past. The other is that it demystifies, to some degree at least, social historical research.[7]

Students depend upon questions we provide to determine what is worth knowing about the past and, while we may bemoan their passivity and lack of intellectual curiosity, we rarely can provide them an opportunity to formulate questions of their own. This is particularly true of survey courses where class size and the amount of material combine to limit the amount of student initiative we can encourage.

Database software and well chosen data sets can help, especially since projects using them readily lend themselves to the large lecture/small discussion group format characteristic of survey courses. What they allow the user to do is quickly determine how many signers of a Shaysite petition in Massachusetts in the 1780s, say, had taxable property, or belonged to the First Congregational Church in Worcester, or were defendants in debtor actions in state courts. DBMS permits the user to explore for patterns and connections and to test hypotheses.

One helpful way of thinking about database managers is in terms of the old-fashioned counter/sorter that statisticians used in the pre-computer era. This spiny device would spear IBM cards that had been punched in a given row and column. Each column represented a single type of information (e.g., age); each row a particular value for that variable. Like the counter/sorter, the database manager goes through a set of cases (no longer physically residing on cards of course) and pulls out those that meet whatever logical conditions the user has specified (e.g., Shaysite petitioners under 40).[8]

For purposes of illustration I have drawn on a DBMS project I use in a survey course on nineteenth-century American history to enable students to explore more fully issues connected with the impact of urban and industrial developments on working-class families. Comparable units could be designed for any number of other standard survey topics.

The data came from Margaret Byington's classic account of life in a mill town in the Pittsburgh region in the early years of this century, HOMESTEAD: THE HOUSEHOLDS OF A MILL TOWN. Byington built her book around the experiences of some ninety working-class families and, as part of her research, got them to keep detailed accounts of their earnings and expenditures. She recorded each family's nationality and race, the father's occupation, whether a family enjoyed running water in its home, the weekly amounts it spent on food, rent, tobacco, and other commodities, as well as other bits of information.[9]

Entering the data proved a simple and straightforward task. The software I used was DATATRIEVE,[10] a database manager developed for use on Digital Equipment Corporation's VAX series of minicomputers. This was a choice based on the fact that, as of the fall 1984 semester, when I first used this material, Assumption students had relatively easy access to the college's VAX/750 and relatively limited access to its then small number of microcomputers. On the other hand, I did nothing with DATATRIEVE on the VAX that I could not, just as easily, have done with dBase[11] or many other DBMS packages on a microcomputer.

Each case in a data set--in the Byington data each family--consists of variables (or "fields" in the terminology of most database managers). A variable is simply a particular type of information you have about a given case. Father's occupation and total weekly income are examples from the Byington data. DATATRIEVE and many other DBMS packages will help you set up the database by asking you to name each variable (or field) and then describe whether the data be stored under it will consist of numbers or dates or amounts of money or letters or some combination of letters and numbers. After each field has been set up, the program will ask you if you want to define another. When you answer "no," it will prompt you to enter the data by displaying for you the name of each variable, in the sequence you specified. If you attempt to enter the wrong sort of data (e.g., a date in a field you defined as containing percentages), the package will alert you to the error and ask you to reenter the correct data.[12]

Because I defined the database and did the data entry myself (a matter of several hours work since there are only 25 fields and 90 cases), I did not have to worry that 40 students would, willy-nilly, produce 39 different data sets. So, the task of teaching them to use the software reduced itself to acquainting them with how to use the program's query and report-writing features. This is a very important point since the pressure to cover large amounts of material is an inescapable component of teaching surveys and few of us are likely to make room in our syllabi for elaborate insturctions on how to use this or that program.[13]

I did have to explain standard Boolean expressions for logical relationships and give examples for the Byington data (see Table 1) since DATATRIEVE and other DBMS packages require the user to employ them. The field or variable names are those I made up when I defined the database. Different database managers have varying rules about the length and other characteristics of these names. As a rule of thumb, one should make up names that are short, descriptive, and simple to spell.

Table 1: Boolean Expressions of Logical Conditions

Expression	Condition	Examples
eq or =	equal to	group = 1
gt	greater than	boarders gt 2
ge	greater than or equal to	yrshere ge 3
lt	less than	rooms lt 3
le	less than or equal to	income le 10.00
ne	not equal to	ownhome ne 0

With DATATRIEVE one uses the FIND command to ask questions about the data. Thus FIND MILLS (the name I gave to the Byington data) WITH GROUP LE 6 AND BOARDER GT 0 causes the program to cull out of the full number of cases those families of foreign birth, that is, those whose group code was less than or equal to six, who also had boarders living with them. DATATRIEVE will respond with a statement stating how many cases meet the specified conditions. The users may stipulate as many logical conditions as they wish; but, if there are only 90 cases to begin with, it is not practical to subdivide the data too many times. DATATRIEVE calls the subset of records it locates a collection and, by default, gives it the name Current. FOR CURRENT PRINT ID, OCC, FAMSIZE, ROOMS causes DATATRIEVE to produce a table with four columns, one each for identification number, occupation, family size, and number of rooms occupied.

It is possible, in other words, for students to start generating useful output within a few moments. They simply need to learn the basic syntax of two DATATRIEVE commands, FIND and PRINT, and learn the Boolean terms for the relationships they wish to explore.[14] With these they can see if families headed by native-born whites were more likely to own their own homes than those headed by native-born blacks, for example. They can see if immigrants who had been in the United States for longer than ten years, say, were more likely to hold skilled jobs in the mills than more recent immigrants were. They can take the model diet for a family of five prepared for the Pittsburgh Chamber of Commerce, and reprinted in HOMESTEAD, and see how many families could afford this minimal level of nutrition. They can compare the families that could to those that could not in terms of ethnicity, father's occupation, presence or absence of boarders, and whatever other factors the data permits and their historical imaginations suggest.

The first time I used the Byington data I assigned a short paper that asked students to "sketch out two separate questions about working-class families you would like to explore. Explain why each is worth answering . . . (and) describe what you could learn about each" from the data. I did not tell them to answer the questions, but to ask them. Those who wished could write a final paper on one of their questions, but this assignment asked only that they think through the significance of their questions and formulate tentative strategies for answering them. This worked well in that it occasioned several excellent papers in addition to some workmanlike ones. Some students, however, needed more guidance in the formulation of sensible historical questions. So I have moved to a more structured assignment that gives students the choice of responding to a question of mine about the data or of formulating and answering one of their own. (See Appendix.)

All of this is possible without going beyond these two commands, FIND and PRINT, and without asking DATATRIEVE to generate any statistics beyond simple counts. Students do, however, soon want the program to calculate some simple descriptive statistics and DATATRIEVE--like most good database managers on minis and micros--does possess a limited statistical capability. Specifically, it will count the number of cases it prints, compute their average value, add the values, identify the highest and lowest value, and calculate their standard deviation. For most survey courses, this is more than adequate since the pedagogical goal is to familiarize students with the sorts of data historians use and the sort of questions they ask rather than to introduce them to cliometrics in any formal way. However, for those students who may have taken a course in statistical analysis and who wish to pursue a line of inquiry in depth, it is possible to get DATATRIEVE and many other database managers to write an ASCII standard file that can be read into a full-fledged statistical package.[15] So, while it will not do multiple regression, such a software package will do useful counting and sorting in response to a few easy-to-learn commands.

The commands are easy to learn, but students with little or no prior experience with computers are going to need a fair amount of handholding. And the teacher--not the staff at the campus computer center--is going to have to do it. Naive software users often experience problems that the computer staff cannot help them with since they cannot know how you defined or structured the data set your students are using.

Students, like the rest of us, are endlessly ingenious when it comes to making errors, but there are several types of problems you can confidently expect. One comes from the beginning user's failure to realize how precise software syntax is. He thinks he has entered exactly what you said would work, and all he gets is a frustrating series of error messages. There is no way to prevent this from happening, but it is important that there be someone (usually you) handy who can explain what went wrong. Otherwise the student may spend hours in fruitless repetition of the incorrect command. After such an experience students are not likely ever to want to see another computer or computer project again.

A second common type of problem comes from the neophyte's altogether understandable desire for software commands to have the same meaning as these words convey in ordinary speech. PRINT is a good example. In DATATRIEVE, as in many other software packages, PRINT causes output to appear on the screen. The student expects it to produce that output on paper. When it does not, he is both baffled and frustrated. A related sort of difficulty arises out of the desire, also entirely understandable, that commands in one software program have the same meaning as they have in others. Many of my students have taken, or are taking, introductory BASIC. There they learn the LIST command that causes the program they are using to scroll down the screen. DATATRIEVE also has a LIST command. Unfortunately, it does something quite different than its namesake in BASIC. It produces a list of the data, case by case. In another of my courses students were using a sample of 1370 cases from the 1880 federal manuscript census. LIST generated a surprisingly large amount of useless screen output.

Students new to computing face worse fates than error messages. They will sometimes enter a command that, while not an error in the technical sense that the program rejects it as invalid, does cause the computer to do something they had not intended. Often they do not even know how to make it stop, to say nothing of knowing how to get the output they originally wanted, so there has to be somewhere they can turn for assistance. A weekly discussion section can be that place. There you can sit them down in front of a terminal or micro and let them show themselves that they have mastered the basic syntax of the key commands. A little success goes a long way toward convincing students that they too, like millions of others, can use computers.

The discussion meeting can serve many purposes. Students can bring up questions, commiserate with one another over the difficulty of getting the software to do what they want, complain about the wait before finding a free terminal or micro, and even attempt to relate their own findings to the material they are encountering in lectures and textbooks. The meetings will not, however, eliminate all difficulties. Many students are going to make some of their most frustrating mistakes late at night when no one is around to help them. The best thing a neophyte can do when faced with some problem beyond his experience is to stop and ask for help even if that means waiting a day or two. So one is well advised to make sure that each student knows how to abort the program. With micros this is usually a matter, all else failing, of turning the machine off. With minicomputers and mainframes it is learning a specific command such as Ctrl C or Ctrl Z. Whatever it is on the software your students are using, they need to know it.

Clearly then using database managers in survey courses is not an entirely painless process for either students or instructors. Both have to learn something about computers and how they operate. Faculty must make discussion section time available to get some of the inevitable initial errors out of the way, and redesign the way they handle in lectures the topics the students will be exploring. These adjustments are inescapable no matter what, but specially designed software for surveys could, were it available, facilitate them. Perhaps it will be some time soon. In the meanwhile, it is worthwhile for teachers to produce their own databases for classroom use. Database managers allow students to become more actively engaged in their own educations. That is no small thing.

The more general lesson to learn is that teachers should look into the range of tasks they can undertake with commercially available wordprocessing, database and file managers, and other software rather than wait for specially designed history programs to appear on the market. In particular, we should ask if some packages might not permit us to undertake teaching strategies that would not otherwise be feasible or to utilize more systematically strategies we already employ. My experience indicates that the answers will be that there are lots of pedagogical uses for computers besides generating short answer tests.

NOTES

[1] A tutorial designed to aid undergraduates in mastering factual information can be found in Fred R. van Hartesveldt, "Using Computers in Lower Division History Courses," Teaching History: A Journal of Methods X (1985), 31.

[2] History Microcomputer Review, in addition to reviewing games and simulations, runs a regular Forum in which readers discuss the strengths and possible drawbacks of this approach to teaching history.

[3] See Willis D. Copeland, "Teaching Students to 'Do' History: The Teacher and the Computer in Partnership," The History Teacher 18 (1985),

189-197, for an account of an ambitious project at the University of California at Santa Barbara.

[4] Minitab is a product of Minitab Project, Statistics Department, 215 Pond Laboratory, The Pennsylvania State University, University Park, PA 16802. SCSS Conversational System is a product of SPSS Inc., Suite 3300, 444 North Michigan Avenue, Chicago, IL 60611.

[5] Richard W. Slatta, "Teaching Historical Research with the Microcomputer," The History Teacher 18 (1984), 45-55.

[6] See Mary Beth Norton et al., A People and a Nation: A History of the United States, 2nd ed. (Boston: Houghton Mifflin, 1986), and James A. Henretta et al., America's History (Chicago: Dorsey, 1987).

[7] Elaine Keuhn, "Demystifying the Modern European Family with Quantitative Data," The History Teacher 15 (1982), 168-9.

[8] A good description of how historians can use database management software is Blaine A. Brownell, "Data Crunching: The Power & Possibilities of Database & Information Management," OAH Newsletter, Computer Supplement, November 1984, s9-11. This entire supplement is worth reading. For a charming account of using the counter/sorter, see Charles Tilly, As Sociology Meets History (New York: Academic Press, 1981), 53-55.

[9] Margaret F. Byington, Homestead: The Households of a Mill Town (Pittsburgh: University of Pittsburgh Press, 1974 reprint of the 1910 edition). For Byington's account of how she compiled the data, see her Appendix I, pp. 187-199. The data are printed in Appendix II, pp. 206-213. Anyone wishing a copy of the codebook I created for this data should write me at: Department of History, Assumption College, 500 Salisbury Street, Worcester, MA 01609.

[10] DATATRIEVE is a product of Digital Equipment Corporation, Maynard, MA 01754.

[11] An excellent introduction to dBase II is Barbara S. Chirlian, Simply dBase II (Beaverton, Oregon: Dilithium Press, 1984).

[12] Needless to say, some DBMS packages are easier to use, are "friendlier," than others. But, user-friendliness is often overrated. Once you do learn how to use a package, the same features that make it friendly, the extensive help messages, for example, will lose much of their value.

[13] Teachers, in addition to having to choose data sets, create codebooks, and define and enter data, need to write brief instruction manuals for student use since the documentation accompanying the software explains far more, and in far greater detail, about the program and its capabilities than you or your students want to learn. My DATATRIEVE manual is available by mail at the address given in note #9.

[14] They have earlier encountered these terms in math courses.

[15] SPSSx, a non-interactive statistical package from SPSS Inc., will read DATATRIEVE databases directly. Minitab and SCSS will not, but both will read edited versions of DATATRIEVE's data files. All three statistical packages will read ASCII standard data files that dBase and other database managers available on microcomputers will write.

APPENDIX

Hi. 160: Nineteenth-Century U.S., 1815-1914 Dr. McClymer

Homestead's Families

Materials: Margaret F. Byington, HOMESTEAD: THE HOUSEHOLDS OF A MILL TOWN; DATATRIEVE domain, MILL; User's Guide to DATATRIEVE

Rationale: Byington's HOMESTEAD was part of The Pittsburgh Survey, a pioneering attempt to use the techniques of social science to measure the pace and direction of social change. It is thus an important document for understanding not only working-class family life in the first years of this century but also the history of reform.

Topic: Were Homestead families largely hapless victims and/or beneficiaries of impersonal economic forces, as many historians have described working-class and immigrant families, or did they have some measure of real control over their own living conditions?

In Appendix IX, Byington reprinted a report of the Committee on Trade and Commerce to the Pittsburgh Chamber of Commerce on the "Comparative Cost of Food and House Rent in Pittsburgh and Other Cities." One way to appraoch this topic is to use the family budget data she collected (in Appendix II and available in MILL) to determine how many Homestead families could maintain the committee's subsistence standard. What factors (e.g., nativity, skill level, family size, presence or absence of boarders) account for a family's success or failure in reaching the standard? The operating assumption behind this approach is that families unable to reach or get much beyond the minimum standard set by the Chamber of Commerce would be unable to exercise much control over other aspects of their lives. YOU DO NOT HAVE TO ADOPT THIS APPROACH. You can set up the general question of working-class autonomy in any way that makes sense to you. BUT, you must define reasonable ways of determining how much control families had over their lives. You should explain your approach at the outset of your essay. Be sure to include a rationale for choosing the measures of control that you did. THESE MEASURES NEED NOT BE QUANTITATIVE. They need only be reasonable. Byington was herself very much interested in this question. And she had other measures than just budget totals. Remember, however, not to use her argument uncritically.