SESSION IX INSTRUCTIONAL COMPUTING

A Symposium
N. John Castellan, Jr., *President*

Integrating statistical software into laboratories and laboratory courses

DARRELL BUTLER
Ball State University, Muncie, Indiana

Reasonable quality statistical software that is easy to use and easy to learn to use is available, and more will become available. This paper describes nine features of statistical software programs that make such programs easy to use. Some methods for evaluating the capability, accuracy, and speed of statistical software are discussed. In addition, some issues about university and software company support are considered. Finally, a technique for integrating software into laboratories is presented.

In some courses, an instructor may want students to calculate statistics as part of a project but may not want to spend days instructing the students on the use of computers or complex statistical packages, such as SPSS. In laboratory settings, a researcher may desire to involve students in data analysis but have similar attendant concerns. The turnover rate of student researchers can be quite high, especially with undergraduates and master's degree candidates, and training students to use complex statistical packages can become very time-consuming. Statistical packages that are easy to use and easy to learn can be very advantageous to instructors in either of these situations (see Brecht, Woodward, & Gistenson, 1985; Butler & Eamon, 1985).

The purposes of the present paper are to describe some characteristics of statistical software that make the software easy for students to use, to discuss some methods for evaluating the accuracy and capacity of easy-to-use programs, to consider issues of university and software company support, and to propose a procedure for integrating statistical software into research and classroom laboratories. Two assumptions are made in this discussion. The first is that the intent is to have students use statistical software as a tool to accomplish a task, in the same way a word processor is used to accomplish a task. This is not the same as computer-aided or computer-assisted instruction. The second is that an important goal is to reduce work rather than create work.

CHARACTERISTICS OF STATISTICAL PROGRAMS

The two basic characteristics of statistical programs that need to be considered are (1) ease of use, and (2) capability and accuracy, as described below.

Ease of Use

There is growing evidence (e.g., Brecht et al., 1985; Butler & Eamon, 1985) that microcomputer software is generally easier to use and easier to learn than mainframe software. Brecht et al. found that students like microcomputer software much more than mainframe software. Butler and Eamon found that mainframe statistical packages were generally harder to learn than microcomputer statistical packages. But any computer can be used, especially if the software has features that make statistics programs easy to learn and use. Nine features that I have found to be important are described as follows.

Menu-driven programs. There are three basic techniques for making selections in programs: menus, questions, and special command languages. Menus are lists from which a user can make a choice. Questions are usually simple queries requesting yes-no answers, numbers, or other simple responses. Command languages require users to type commands such as "RUN MULTIPLE REGRESSION."

For novices, menus and questions are much easier to use than are command languages because they require no memory or familiarity with options. Many statistics packages use menus sometimes and questions other times. Informal tests in my courses and lab suggest that menus are

---

The author's mailing address is: Psychological Services, Ball State University, Muncie, IN 47306.

often more efficient than questions for selecting statistical procedures. However, if a single menu becomes too large, there can be problems. Students can become confused and unable to make appropriate discriminations. Packages that offer many options should have hierarchically related menus.

On-screen editing. Data entry may contain some errors, and it is desirable that a statistical package allow students to change data values. Many techniques have evolved for doing this. The two most common are on-screen editing and numbered data. On-screen editors display the data (or part of the data) on the CRT and permit users to change what they see. Numbered-data editors sequentially number data but do not (necessarily) display the data. Users change data by entering a new value for a particular number. On-screen editing is more desirable because it does not require that users remember the number associated with each datum. However, a numbered-data editor is better than no editor at all.

Error handling. A program to be used by students should detect all errors (entry or computational) and suggest appropriate behavior to the user. If it does not, students will constantly be looking for the instructor's assistance.

Escape from wrong "path." Students occasionally misunderstand a question and make an incorrect response, or they accidentally press the wrong key. Programs can provide obvious solutions to these problems. Menus can include an option that permits the user to return to the previous menu or screen. Questions can include an answer that defaults to the previous question. For example, "How many subjects? (press 0 to return to previous question)." These solutions are particularly helpful when students are selecting procedures and entering data.

Printer output. Students typically require only simple statistical output. A printer is not really needed, and in fact can be undesirable. It creates extra, unnecessary obstacles to learning to use computers and can be expensive. Not only can ribbons and paper be wasted, but students can damage a printer. However, in laboratories and in laboratory courses requiring complex analyses (two-way, or more, ANOVA, multiple regression, etc.), printer output is desirable.

Simple, consistent use of keypresses. Butler and Eamon (1985) found that simple keypress sequences decrease errors and increase student satisfaction. One of the most common sources of typing error is pressing the return key when it is not required. This occurs especially when a program uses the return key inconsistently: Some inputs require pressing the return key and others do not. Keypress use must be consistent.

Jargon consistent with student training. I have sometimes attempted to use statistical programs written by professionals in fields other than psychology. This can lead to terminology problems. There are multiple names for procedures even within psychology, especially for ANOVA, but the problems of terminology are magnified

when the package is designed for a field other than psychology. Some packages (such as *Keystat* [Eckblad, 1984] and *Statistics with Interpretations* [Butler, 1985b]) minimize jargon problems by using menus that request user intent (e.g., "analyze means or variances?" or "groups dependent or independent?") instead of procedure names. Students like this approach.

Help messages. Help messages save time, reduce errors, and simplify human-computer interaction (see Cohill & Williges, 1985). Although research suggests that help messages can be on screen or in printed manuals, students strongly prefer screen messages. Without them, since few students will ever use a manual, they turn to the instructor.

Screens not too cluttered and not too many screens. If screens are crowded with letters and numbers, students make mistakes. When called upon to answer questions, they will not be able to make correct decisions in response to computer requests. When reading outputs, they will have difficulty in deciding what information is important. Spreading complex information over many screens does not always help, especially for output. It is more important for the output to be well organized and concise.

Capacity of Programs

Students can become very frustrated if they enter a large data set only to find that the program cannot handle the amount of data they have entered. Other problems concern accuracy, speed, variability of routines, and compatibility of data across routines. Each of these problems is discussed below.

Maximum number of scores and cells. Years ago, there seemed to be a positive correlation between price of package and capacity, in terms of number of cells and number of data. If that correlation still exists, it is not as large. There are several packages (e.g., *ABC* [Consortium Software, 1983] and *Statistics with Interpretations* [Butler, 1985b]) that can handle very large data sets and cost less than \$100.

Students typically do not process gigantic data sets, so a program limited to one or two thousand data is adequate. However, labs differ, and it is worthwhile to think about what capabilities are really needed.

There are several reviews of software that can be used to identify program capabilities (e.g., Butler & Eamon, 1985; Carpenter, Deloria, & Morganstein, 1984; Lehman, 1986). Such reviews tend to provide more complete information than is supplied by manufacturers and marketers, and can be useful in comparing and contrasting packages.

Reasonable digits of accuracy. Accuracy of programs can be a problem no matter what kind of a computer is used. There are many algorithms that can be used to calculate statistics on a computer, and the algorithms can be extremely important in determining the accuracy of the program. Butler and Eamon (1985) reported that *MINITAB*, running on a VAX mainframe computer, had only two digits of accuracy in the ANOVA table. The

ONEWAY procedure is SPSS is even less accurate. At the other extreme, *SYSTAT* (Wilkinson, 1984), running on an Apple II with a CP/M card, has greater than six digits of accuracy in all procedures.

There are several ways to check accuracy. The user can develop his or her own example problems. This technique has the advantage of allowing the user to know how the program works on problems similar to those he or she typically needs to solve. Unfortunately, it is time-consuming to develop good tests and then run all of them. Longley (1967) published a data set (and technique) for testing programs. This data set has been used to benchmark many programs (e.g., Carpenter et al., 1984). A similar approach has been taken by Butler and Eamon (1985) and by reviews published in *The American Statistician*.

Reasonable speed. Programs use different computational strategies so direct comparison of speed is often difficult. For example, *ABC* has a complicated set of procedures for defining data. Many descriptive statistics appear to be calculated as part of the data definition stage. As a result, the package can print out descriptive statistics of several thousand data in a few seconds on an IBM PC. On the same machine, *Statistics Software for Microcomputers* (DiFazio, 1984) takes 10 min to calculate descriptive statistics on 200 numbers. Timing benchmarks have been reported in some sources (e.g., Pease & Lepage, 1984), but they are not crucial for most student problems. For students, only really slow programs are a problem. From their point of view, almost anything that does calculations will produce an answer faster than they could without computer assistance.

Variety of statistical procedures. Many newer packages have substantial variety. This is really valuable because users are more likely to use an appropriate test, and it forces users to think about the noncomputational aspects of statistical analysis. Is the data really interval and approximately normally distributed in each cell? Does it really matter?

With packages that have many routines, complications can arise if the various routines are not compatible. In some packages (e.g., *Statistics Programs for the Apple II*, Steinmetz, 1981) students must rank order the data for some routines but not for others. Data entered for a Pearson product moment correlation (r) (and saved on a disk) is probably not in the correct form to be used in the calculation of a Spearman rho correlation. Data must be reentered.

However, some incompatibility is desirable. The programs should check to assure that data saved while one routine is being run is in the correct form for another routine. Otherwise, students can end up running a Pearson r on two data sets of different sizes (perhaps saved after running an independent-groups t test). Few, if any, programs are intelligent enough to be restrictive only when they should be, but it is desirable that they be as intelligent as possible.

SUPPORT

University Support

In general, although minicomputers and mainframe computers are faster and more powerful, microcomputers are easier to use. However, there is great variation in campus computing hardware, software, and support, so it is difficult to make statements that have no exceptions. If you rely on a campus computing system, then maintenance costs and software support may be provided. This economic savings typically removes faculty from the decision-making process, so that it is sometimes difficult for the faculty to obtain the exact programs and equipment they need. Alternatively, individuals, departments, or schools can develop computer laboratories that are relatively independent of campus computing systems. This approach leaves decisions in the hands of those who will be using the machines, but it requires economic support and a faculty member who is willing to administer the facility.

Software Company Policies

There is a wide variety of policies among software companies. Some companies charge a fee for previewing their software; some have excellent telephone support, whereas others provide only sales personnel. It is unlikely that you will find every company's policy adequate for your purposes. Three issues to consider are their policies regarding single machines, backup copies, and multiple-copy discounts.

Many software companies include a contract stating that the software can be used only on one computer. This policy is unreasonable. Many instructors have computers at home and one or more computers in their laboratories or offices, as well as computers available for students. The computers are often compatible so that work can be done at the most convenient location. The single-machine policy is inconsistent with the flexible working arrangements that so many instructors have developed. Some small software distributors (such as Oakleaf and East West Software) do not limit use to a single machine. We must convince larger distributors to adopt this policy.

Backup copies are needed in an environment with novice computer users. Some companies provide a backup copy with any purchase. Some companies sell the disk in a form that allows the buyer to make his/her own backup copy. Other companies copy-protect their products and do not provide backup copies. Search for companies with reasonable policies.

Most computer software companies have multiple-copy discounts; that is, when more than one copy of the software is purchased, the buyer receives each copy for less than the cost of a single copy. This is a common policy, but it is important to consider, before purchase, whether or not multiple copies will be needed so that more than one student at a time can work in a laboratory of computers.

INTEGRATING SOFTWARE INTO LAB ENVIRONMENT

It is important to move slowly toward implementation. If you write software and can debug and change programs quickly to meet your needs, then this admonition is not so important. But if you prefer to obtain software from other sources, then implement slowly. A series of steps that has worked well for me follows.

As a first step, find out what is available. There are a number of published lists and reviews of statistical software (see Academic Computer Center of Gettysburg College, 1984; Butler & Eamon, 1985; Datapro/McGraw-Hill, 1985; Neffendorf, 1983; Pease & Lepage, 1984). Although it is unusual for journals to carry comprehensive lists (for reasons I do not yet completely understand), journals likely to have some lists and reviews are: *The American Statistician*; *Behavior Research Methods, Instruments, & Computers*; *Collegiate Microcomputer*; and *Contemporary Psychology*. At virtually every psychology conference in the last few years, there have been talks, posters, and handouts on available software (e.g., Butler, 1985a; Butler & Castellan, 1985; Porter & Beins, 1985).

Faculty should preview programs, trying out a few routines that may be useful. It is also highly desirable to have a few students use any package that might be used in a laboratory. The students can test capabilities and limitations. Then these student experts are available to help others. If the package is to be used in a research laboratory, it will then be fairly well integrated and the students can help one another.

If the plan is to use the programs in laboratory classes, then after student and faculty previewing, use the package on one short class assignment before proceeding to many assignments: This provides valuable field test information and can suggest useful supplementary materials. Most programs are not perfect, but a brief handout can clarify use of a package or give students ideas to try if they have difficulty.

After supplements are developed, implementation should go smoothly, although further evaluation after implementation may be desirable (see Castellan, 1986).

REFERENCES

- ACADEMIC COMPUTER CENTER OF GETTYSBURG COLLEGE (1984). *Computing in undergraduate psychology: A practicum software guide*. Gettysburg, PA: Author.
- BRECHT, M. L., WOODWARD, J. A., & GISTENSON, J. (1985). Comparison of three statistical computer programs in teaching analysis of variance. *Journal of Educational Technology Systems*, **13**, 237-247.
- BUTLER, D. L. (1985a, August). *Microcomputer statistics packages in undergraduate statistics and methods courses*. Paper presented at the meeting of the American Psychological Association, Los Angeles.
- BUTLER, D. L. (1985b). *Statistics with interpretations* [Computer program]. Muncie, IN: East West Software.
- BUTLER, D. L., & CASTELLAN, N. J., JR. (1985, October). *Microcomputer software for the undergraduate curriculum*. Activity run at the Second Mid-America Conference for Teachers of Psychology, Evansville, IN.
- BUTLER, D. L., & EAMON, D. B. (1985). An evaluation of statistical software for research and instruction. *Behavior Research Methods, Instruments, & Computers*, **17**, 352-358.
- CARPENTER, J., DELORIA, D., & MORGANSTEIN, D. (1984). Statistical software for microcomputers. *BYTE*, **9**(4), 234-264.
- CASTELLAN, N. J., JR. (1986). Issues in effective use of computers in introductory and advanced courses in psychology. *Behavior Research Methods, Instruments, & Computers*, **18**, 251-256.
- COHILL, A. M., & WILLIGES, R. C. (1985). Retrieval of HELP information for novice users of interactive computer systems. *Human Factors*, **27**, 335-343.
- CONSORTIUM SOFTWARE (1983). *ABC* [Computer program]. Ann Arbor, MI: Author.
- DATAPRO/MCGRAW-HILL (1985). *Guide to Apple software* (2nd ed.). New York: Author.
- DIFAZIO, M. (1984). *Statistics software for microcomputers* [Computer program]. Duxbury, MA: Kern International.
- ECKBLAD, J. (1984). *Keystat* [Computer Software]. Decorah, IA: Oakleaf Systems.
- LEHMAN, R. S. (1986). Macintosh statistical packages. *Behavior Research Methods, Instruments, & Computers*, **18**, 177-187.
- LONGLEY, J. W. (1967). An appraisal of least squares programs for the electronic computer from the point of view of the user. *Journal of the American Statistical Association*, **62**, 819-841.
- NEFFENDORF, H. (1983). Statistical packages for microcomputers: A listing. *American Statistician*, **37**, 83-86.
- PEASE, J. W., & LEPAGE, R. (1984). *An evaluation of selected microcomputer statistical programs* (Working Paper No. 15). East Lansing, MI: Department of Agricultural Economics, Michigan State University.
- PORTER, B., & BEINS, B. (1985, October). *Demonstrations in perception, statistics, and experimental psychology*. Paper presented at the Second Mid-America Conference for Teachers of Psychology, Evansville, IN.
- STEINMETZ, J. (1981). *Statistics programs for the Apple II* [Computer program]. Athens, OH: Author.
- WILKINSON, L. (1984). *SYSTAT* [Computer program]. Evanston, IL: SYSTAT.