

Effects of computer interfaces on computer-based statistical analysis

P. WESLEY SCHULTZ and DALE E. BERGER
The Claremont Graduate University, Claremont, California

Thirty-one first-year psychology graduate students in a computer applications course completed a set of structured problems, unstructured problems, and data-screening problems in each of two statistical computing environments: a menu-based interface (SPSS for Windows) and a traditional command-based interface (SPSSx). Performance on the menu-based interface was generally superior to performance on the command-based interface. More of the structured problems were completed successfully within the menu-based interface. The menu-based interface also facilitated error identification, was rated as easier to use, and was preferred nearly 4 to 1 over the command-based interface. For the unstructured problems, students identified more relationships in the data set and issued more statistical commands when working with the menu-based interface. These findings are consistent with the interpretation that because the menu-based interface requires fewer mental resources to be dedicated to the mechanics of analysis, more resources are available to devote to higher level problem solving.

We thank Will Brown and Jenny Houghton Mattocks for adapting their Computer Applications class to accommodate our study and Lorne Olfman for providing comments on an earlier draft of this paper. Correspondence concerning this article should be addressed to P. W. Schultz, Psychology Program, California State University, San Marcos, CA, 92096 (e-mail: psch@mailhost.csusm.edu).

Over the past 10 years there has been an overwhelming shift from command-based to menu-based computer interfaces. Although menu-based computer interfaces are designed to make computer use easier and more productive than command-based interfaces, little is known about the effects of these interfaces on the user. It is important, especially given the current direction of computing and the ubiquitous nature of computers in our society, to understand how computing interfaces alter the way in which people interact with the computer. A specific issue addressed by the current study is whether the newer menu-based interfaces change the way people use computers to solve problems and perhaps even change the solutions they obtain. Computing environments have been classified as command based, direct manipulation, or menu based (Davis & Bostrom, 1993). *Command-based interfaces* are those in which the user issues specific requests using syntax defined by the program. *Direct manipulation interfaces (DMI)* utilize icons or objects on the computer terminal that represent specific commands. Using a mouse, the user can perform tasks by pointing and clicking. *Menu-based interfaces* allow the user to select a desired command from lists of available commands, perhaps also by pointing and clicking with a mouse. However, many DMI programs also make substantial use of pull-down menus and menu-driven commands. In the current study, we compare command-based and menu-based interfaces—the two most commonly used computer interfaces.

Research investigating the effects of computer interfaces on human-computer interactions has produced mixed results. For example, several authors report that DMI computer interfaces aid in learning (Michard, 1982; Woodgate, 1985), while others report no differences between DMI and command-based interfaces (Carroll & Mazur, 1986; Whiteside, Jones, Levey, & Wixon, 1985).

In a recent study, Davis and Bostrom (1993) examined the relative effects of both training and computer interface (DMI vs. command-based) on 80 novice computer users. Undergraduates from introductory computer courses completed tasks using either a DMI interface (Macintosh) or command based (DOS) interface. Their results showed that the computer interface was a strong determinant of the outcome. Students working in the DMI environment performed better on tasks involving simple recall of information and tasks that required the application of the recalled information. A marginally significant effect showed that the DMI system was perceived to be easier to use. The authors argued that the advantage of a DMI environment was that it allowed users to work with on-screen representations of information in a way analogous to real life. Because the DMI interface is more representative of real life, it reduces the users' "cognitive processing load" (p. 74). This explanation suggests that computer interfaces in which users do not need to devote as much cognitive capacity to recalling and processing a command language will produce quicker and more accurate problem solutions (Hutchins, Hollan, & Norman, 1986; Schneiderman, 1982, 1983). The differences in the required amount of cognitive processing between computer interfaces may be the direct result of recall versus recognition. Both menu-based and DMI interfaces rely on recognition, while command-based interfaces rely on recall. As humans have only a limited amount of cognitive capacity, any activity that diverts mental resources away from a task decreases the probability that the task will be successfully completed (cf Martindale, 1991). Research on reading comprehension demonstrates the costs of devoting cognitive resources to low-level aspects of a task. In studying the development of reading ability, Perfetti and his colleagues (Perfetti & Hogaboam, 1975; Perfetti & Lesgold, 1979) found that the faster a reader can decode individual words, the better his/her comprehension of the material. The more quickly that individual words are understood, the more mental resources are available to process the meaning of the combinations of words (LaBerge & Samuels, 1974).

The reduction in demand on mental resources provided by both DMI and menu-driven interfaces makes available more resources for working on the task. However, depending on the task, this may not be advantageous. In a study of undergraduate papers written on computers, Halio (1990) examined differences between menu-based and command-based word processing programs. Halio randomly assigned 25 undergraduate college students to write a paper using a DOS interface and 25 students to write a paper using a Macintosh interface. Analyses revealed that the Macintosh papers were less sophisticated than the DOS papers. Papers written on the Macintosh used fewer complex sentences, were judged to be written at a lower grade level, used fewer words per sentence, and had more spelling errors per paper than did the DOS papers. Halio argued that the differences in papers were due to the computer interface. Because the menu-based interface is easier to use, it facilitates a more conversational style of writing, causing loosely structured sentences. Students "tend to think of the Mac as a sort of a toy" (p. 18). The simpler writing style of students in Halio's (1990) menu-driven condition reflects a more creative, less structured approach to completing the task. Although this style of problem solving is not advantageous for a paper written for a structured writing course, it is important to note that it may be better suited to courses in creative writing or other less structured tasks. The purpose of the current study was to extend the research on computer interfaces by investigating the effects of menu-based versus command-based computing environments on students doing statistical analyses.

We hypothesized that if cognitive resources could be conserved through the use of a menu-based program intended to reduce the memory load in the selection of commands by using recognition instead of recall, the quality and sophistication of statistical analyses would improve. Because an analyst using a menu-based interface is not required to remember specific commands or syntax, more cognitive resources will be available for interpreting and exploring the data than when a command-based interface is used. We expected this advantage might be more evident for unstructured problems than for structured problems because there are more opportunities for expression of creativity and higher level problem solving in unstructured problems. Two sets of hypotheses were tested:

1. Participants using a menu-based interface will accomplish both structured and unstructured problems faster than participants using a traditional command-based interface. This effect will be more pronounced for unstructured problems.
2. Participants using a menu-based interface will perform better (i.e., solve problems more accurately, use more commands, identify more relationships in a data set) than will participants using a command-based computing environment. This effect will be more pronounced for unstructured problems than for structured problems.

METHOD

Subjects

The subjects in the study were first-year graduate students in psychology enrolled in a computer-applications course. At the time of the study, the students had completed 7 weeks of intermediate-level statistics, and most were new to computer-based statistical analysis. The course covered a variety of computer applications, of which both the mainframe and Windows versions of SPSS were included. The course included approximately 6 h of instruction on each program. All students enrolled in the course agreed to take part in the study.

Materials

A mainframe version of SPSSx (Version 3.1) running on a VAX system was used as a command-based computer interface. This mainframe version of SPSSx operated in the

traditional computing environment, required knowledge of a command language, and processed command files in batch. Users were able to examine output files on their computer terminals. A Windows version of SPSS (6.1) was used on IBM 386 desktop computers as a menu-based computer interface. The Windows version of SPSS operated in a menu-based environment featuring pull-down menus, quick responses to requested commands, and easily produced graphical data representations (i.e., charts, plots, and graphs). SPSS was selected for this study for two reasons. First, the two versions of SPSS are identical in the command options, format of the output, and in the types of statistical analyses available. Secondly, SPSS is widely used among psychology students, and both versions were taught as part of an existing course in computer applications.

Two parallel data sets were constructed for the study, one containing information about different cars, the other containing information about different brands of beer. Identical versions of each data set were constructed for use on the mainframe version of SPSSx and the Windows version of SPSS. Each data set contained eight variables of interest and 45 cases. In all conditions, students were given code sheets in which the variables were clearly defined. In the Windows condition, students were given the data files in .SAV format, which included the format and label for each variable. Similarly, in the mainframe condition, students received a partial command file in which the correct data set was identified along with the *data list* command used to define the format and label for each variable.

Three problem sets were assigned for each condition. The first set of problems was highly structured, asking students to identify which beer/car scored the highest on a single variable, to compare two specific groups of cases, and to generate a specific histogram. The second problem set was less structured, asking students to characterize a specific relationship in the data, and to find and describe "a couple of other relationships." The third problem set asked students to identify data-coding errors in a modified version of each data set. The modified data set contained eight specifically inserted errors, two of each of four types: Extremely high outliers (e.g. a car that gets 150 mpg), extremely low outliers (e.g., a beer that has 5 calories per serving), impossible data (e.g., a rating of 5 on a 4-point scale), and missing data codes that were not defined as missing in the program (e.g., 9 or 99 entered into the data set). The possible range for each variable, along with missing values, were clearly listed in the codesheet included with each data set.

After completing both sets of problems, the students completed a questionnaire designed to measure past experience with each computer program, the amount of time required to complete each problem, assessment of the ease of use of the computer program, and overall preference for either the command-based or menu-based program. With the exception of the preference question, which was dichotomous, responses were made on a 7-point Likert scale.

Procedure

Thirty-one students attempted to complete the computer problems at the conclusion of a 7-week computer applications course. The computer applications course was taught by a pair of instructors who were blind to the purposes of the study. The assignment consisted of six computer-based problems, three from each data set (cars and beer). For each participant, one data set was analyzed using SPSS for windows (menu driven) and the other using SPSSx (command based). The order of the interfaces (menu based and command based) was counterbalanced, as was the order of data sets (cars, beer), producing four test versions. Participants were asked to complete the problems independently within a 1-week period.

They were also asked to record the sequence of commands issued for each problem, as well as the amount of time dedicated to each problem set. Student names were replaced by code numbers to ensure anonymity. Upon completing the problems, the students were debriefed.

RESULTS

Prior to testing the main hypotheses, differences between the two data sets and possible order effects were tested. Two 2 X 2 between-subjects analyses of variance were conducted to test for equality of the two data sets (beer vs. cars) and possible order effects (menu vs. command based first). For this analysis, responses only to the first problem set were analyzed. Each participant solved problems in either the command-based or the menu-based program first, and used one of the two data sets. Three dependent measures were the total amount of time required to complete the problems, the total number of issued commands, and the number of errors identified in the modified data sets. These analyses showed no significant main effects or interactions for the total amount of time or for the number of problems solved correctly. However, a significant main effect of data set was found for the number of errors identified [$F(1, 28) = 4.52, p < .05$]. Although the two data sets were intended to be similar and the same types of errors were inserted into both data sets, more errors were identified in the beer-data set ($M = 5.16, SD = 2.06$) than in the car-data set ($M = 4.40, SD = 1.76$).

This study contained 12 dependent variables, listed in Table 1, and one independent variable at two levels (menu based, command based). To control for Type I errors, an omnibus multivariate test of differences was used to compare the menu- and command-based interfaces. Ten of the dependent variables (interface preference and the number of perfectly correct responses were excluded because they were dichotomous and not normally distributed) were compared across the two computer interfaces using a within-subjects MANOVA. Several instances of missing data occurred on the self-report variables because students failed to record the starting and finishing time for each problem, and 2 students did not complete the self-report questions at the end of each set of analyses. A total of 10 cases contained one or more pieces of missing data. To avoid excluding these cases because of list wise deletion on the omnibus MANOVA, missing data were replaced with the variable mean score. This replacement will slightly increase the alpha level by reducing the amount of variance. The results from this multivariate analysis revealed a highly significant overall difference between the two computing environments [Pillai's $F(1, 21) = 13.18, p < .001$]. Table 1 presents the means, standard deviations, degrees of freedom, and 95% confidence intervals around the differences between the two interfaces. Scores reported in Table 1 are not adjusted with mean replacement, and missing cases were excluded on a pairwise basis in each of the univariate analyses reported below.

The first hypothesis predicted that problems solved using the menu-based interface would require less time than those solved using the command-based interface. This effect was predicted to be stronger for less structured problems. An examination of the mean scores reported in Table 1 shows substantial differences between the two interfaces—a difference of -7.9 for the structured problems and 4.7 for the unstructured problems. However, the confidence intervals indicate that these differences are not statistically significant.

The second hypothesis predicted that more problems would be solved correctly in the menu-based interface than in the command-based interface. Because the unstructured problems allowed the students to examine the data in their own way, it was difficult to code answers as either correct or incorrect. For this reason, responses to only the structured problems were scored as right or wrong. The 31 students answered two structured problems in

each environment: (1) identify a specific case with the highest score on a variable (e.g., most calories, best miles per gallon), and (2) conduct a *t* test (e.g., Do lite beers have fewer calories than regular beers?). A paired-samples *t* test comparing the number of correct answers (out of a possible 2) found that the problems were less likely to be solved correctly in the command-based interface ($M = 1.64$) than in the menu-based interface [$M = 1.97$; $t(30) = 3.00$, $p < .01$]. A nonparametric McNemar test (1975) for dependent proportions confirmed that the rate of perfect performance (i.e., solving both problems correctly) was greater for the menu-based program (97%) than for the command-based program [67% ; $\chi^2(1) = 5.82$, $p < .05$].

For this analysis, participants were categorized as either answering both correctly or making at least one error.

The unstructured problems asked students to identify several possible relationships among the variables in the data set. Three variables were analyzed to test for differences between the two interfaces: the number of commands issued for the problems, the number of relationships identified, and the number of words used to describe the results. Paired-samples *t* tests were conducted to compare the menu-based versus command-based unstructured problems on each of these three dependent variables. As shown in Table 1, menu-based users identified more relationships [2.32 vs. 1.68; $t(30) = 3.23$, $p = .003$] and used more commands to solve the problems [3.19 vs. 2.52; $t(30) = 2.57$, $p = .015$]. No significant differences were found for the number of words used to describe the results of the analysis [$t(30) = 1.43$, $p = .26$].

The students were expected to identify more erroneous data in the modified data sets when working in the menu-based interface than when working in the command-based interface. They also were expected to complete the problem in less time. These hypotheses were tested with paired-sample *t* tests on the number of errors identified and the amount of time required to solve the two problems. Of the 31 students, 6 failed to record the amount of time used for both interfaces. An average of 18 min was used to search the data for errors for the menu-based interface; the command-based problems used an average of 22 min [$t(24) = -1.79$, $p = .09$]. Students correctly identified more errors when working in the menu-based interface [5.4 errors vs. 4.3 errors; $t(30) = 2.23$, $p = .034$].

A postsession questionnaire asked the students to rate both the command-based and menu-based versions of the SPSS program on several dimensions. The menu-based version was rated as easier to use on a 7-point scale [5.4 vs. 3.6; $t(28) = 3.74$, $p < .01$]. No differences were detected in the rated ability of the program to conduct the desired analysis [$f(28) = 1.54$, $p = .14$] or in the ability of the program to generate a desired perspective on the data [$t(28) = 1.60$, $p = .12$]. As shown in Table 1, the menu-based program was rated as more enjoyable to work with [4.0 vs. 3.1 on a 7-point scale; $t(28) = 2.14$, $p = .04$]. Similarly, when asked which program they preferred, 22 students rated the menu-based version as preferable, 6 rated the command-based version as preferable, and 3 did not answer (binomial test, two-tailed, $p = .003$).

Table 1
Means and Standard Deviations for Menu-Based and Command-Based Solutions

Problem Type	Menu-based		Command-based		<i>df</i>	Difference 95% CI
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>		
Minutes spent	36.46	20.19	44.32	23.98	27	-7.86 ± 11.67
Solved both problems correctly *	97%		67%			*
Unstructured						
Minutes spent	27.54	14.68	22.88	12.78	25	4.66 ± 6.95
Number of relationships	2.32	.94	1.68	.65	30	.64 ± .41 _L
Number of commands	3.19	1.19	2.52	.77	30	.67 ± .54 _L
Words in write-up	18.48	10.65	14.28	11.72	30	4.20 ± 6.03
Error Identification						
Minutes spent	18.46	9.14	22.17	12.36	24	-3.71 ± 4.27
Errors identified (out of 8)	5.42	2.13	4.31	1.71	30	1.11 ± .98*
Self-Report						
Ease of use ^b	5.38	1.45	3.55	2.06	28	1.83 ± 1.00 _L
Can do desired analysis ^b	5.69	1.34	5.24	1.15	28	.45 ± .60
Enjoyable	4.03	1.72	3.14	1.88	28	.89 ± .86*
Preferred program ^c	71%		19%			_L

*McNernar test two-tailed $p < .05$. ^bRated on a 7-point Likert scale (1 = *strongly disagree*; 4 = *neutral*; 7 *strongly agree*). ^c10% expressed no preference; binomial two-tailed, $p < .01$.

* $p < .05$. _L $p < .01$.

DISCUSSION

This study was designed to assess the effects of computer interfaces on statistical analysis. We hypothesized that because the menu-based computing environment makes fewer cognitive demands on the user than does the command-based environment, students would not only solve problems faster and more accurately when working in the menu-based environment, but would also perform better on higher level unstructured problems. Overall, the data supported these expectations. These novice students clearly felt that the menu-based program was easier to use. Analyses of several self-report questions indicated that they had found the menu-based program easier to use and that they preferred it over the command-based environment nearly 4 to 1.

The first hypothesis predicted that problems solved using the menu-based computing environment would require less time to complete than the problems solved using a command-based computing environment. Although this hypothesis was not supported, the confidence intervals were large relative to the mean differences. Given the small sample size and large variance, there was not enough statistical power to make any firm conclusions regarding the time scores. The second hypothesis predicted that students would perform better when working in a menu-based environment. Only 1 student failed to solve a structured problem when working in the menu-based environment, while 10 failed to solve a comparable problem when working in the command-based environment. Subsequent analyses revealed that students used more commands and identified more relationships in the data when working in the menu-based environment.

The advantages of the menu-based program are most likely due to the ease of use. The primary difference between the menu-based and command-based programs is the way in which commands are issued. The menu-based program allows students to point and click with a mouse to select a desired command from a list of those available, whereas the command-based program requires typing the commands and perhaps searching through a manual to find the command. The menu lists all the possibilities, which may stimulate the analyst to conduct other statistical tests and examine the data in greater depth. Students were more willing to try out different commands and explore the data in greater detail when working in the menu-based environment, and the solutions for the analyses conducted in the menu-based interface were more accurate. Additionally, there was a nonsignificant trend for the students' descriptions of their results to be longer (a mean difference of 4.2 words), indicating a better understanding of the results. However, more power (either through a larger sample or more homogeneous scores) is needed before any solid conclusions can be drawn regarding this variable. An especially important advantage of the menu-based program was that more relationships were identified in an unstructured problem. Research from cognitive psychology indicates that as a lower level task becomes easier, more mental resources are available for more sophisticated processing of information. For example, as children learning to read develop the ability to decode written words more quickly, their comprehension of meaning improves (cf. LaBerge & Samuels, 1974). Similarly, as computing environments become easier to use, fewer mental resources are required for the lower level mechanics of analysis and more resources are available for elaboration and consideration of new directions for analysis. This could account for the greater success in identifying relationships in the data found when students used the menu-based statistical program.

One limitation of this study is that the students were fairly new to computer-based statistical analysis. Although the students had similar exposure to the two programs, they may have found the menu-based program easier to learn than the command-based program,

and thus had greater expertise with the menu-based program at the time of the study. Long-term users of either type of program can become very facile with producing desired output. The results from this study may not generalize to experts at computerized statistical analysis. In fact, it seems likely that experts at a specific command-based interface may not suffer the impairments demonstrated by novice users of the same program. Despite these limitations, the results of this study are clear: Novice students who conducted statistical analyses in a menu-based environment answered more questions correctly, identified more errors in a data set, used more commands, and identified more relationships among variables than when they worked in a command-based environment.

The results from this study have implications for both researchers selecting a statistical package and teachers attempting to integrate computer-based statistical analyses into their curriculum. In selecting statistical packages for data analysis, researchers should choose a program that minimizes the cognitive resources required to operate it. For the novice user, this means choosing a program that uses a menu-based interface. Our results clearly show that, for novice analysts, a menu-based program can have advantages over a command-based program. For the expert user, our cognitive-load explanation suggests that switching statistical packages may be associated with a period of less sophisticated analyses while the analyst becomes proficient with the new program. However, our results do not suggest that expert users will necessarily perform better with a menu-based program than with a command-based program.

Our results also have implications for teachers choosing a computer package to be used by novice students: Select a menu-based program over a command-based program. Some instructors may be hesitant to use a menu-based program because they feel more comfortable teaching command-based programs with which they are more familiar. However, our study shows that the greater amount of cognitive resources required for command-based programs decreases the quality of the analyses and interpretations among novice users.

In conclusion, as computer technology continues to evolve, it is important to understand how changes in technology affect the ways in which humans solve problems. This study demonstrated that, for a range of tasks in statistical analyses done by novice users, the computer interface does affect problem solving. Understanding these effects is important to researchers who utilize computerized statistical programs, instructors who teach these programs, and computer scientists who design human-computer interfaces.

REFERENCES

- CARROLL, J., & MAZUR, S. A. (1986). LisaLeaming. *IEEE Computer*, **19**, 35-49.
- DAVIS, S., & BOSTROM, R. (1993). Training end users: An experimental investigation of the roles of the computer interface and training methods. *MIS Quarterly*, **17**, 61-85.
- HALIO, M. P. (1990). Student writing: Can the machine maim the message? *Academic Computing*, **4**, 16-19, 45.
- HUTCHINS, E., HOLLAN, J., & NORMAN, D. (1986). Direct manipulation interfaces. In D. A. Norman & S. W Draper (Eds.), *User centered system design-New Perspectives on human-computer interaction* (pp. 87- 124). Hillsdale, NJ: Erlbaum.
- LABERGE, D., & SAMUELS, S. J. (1974). Toward a theory of automatic information processing in reading. *Cognitive Psychology*, **6**, 293-323. MARTINDALE, C. (1991). *Cognitive psychology: A neural-network approach*. Pacific Grove, CA: Brooks/Cole.
- MCNEMAR, Q. (1975). *Psychological statistics* (5th ed.). New York: Wiley.
- MICHARD, A. (1982). Graphical presentation of Boolean expressions in a database query language: Design notes and ergonomic evaluation. *Behaviour & Information Technology*, **1**, 279-288.
- PERFETTI, C. A., & HOGABOAM, T. (1975). The relationship between single word decoding and reading comprehension skill. *Journal of Educational Psychology*, **67**, 461-469.
- PERFETTI, C. A., & LESGOLD, A. M. (1979). Coding and comprehension in skilled reading and implications for reading instruction. In L. B. Resnick & P. A. Weaver (Eds.), *Theory and practice of early reading instruction* (pp. 57-84). Hillsdale, NJ: Erlbaum.
- SHNEIDERMAN, B. (1982). The future of interactive systems and emergence of direct manipulation. *Behaviour & Information Technology*, **1**, 237-256.
- SHNEIDERMAN, B. (1983). Direct manipulation: A step beyond programming languages. *IEEE Computer*, **16**, 57-69.
- WHITESIDE, J., JONES, S., LEVEY, P., & WIXON, D. (1985). User performance with command, menu, and iconic interfaces. *Human factors in computing systems* (CHI '85 Conference Proceedings, pp. 185-191). New York: ACM.
- WOODGATE, H. S. (1985). The use of graphical symbolic commands (Icons) in application programs. In C. E. Vandoni (Ed.), *Proceedings of the European Graphics Conference and Exhibition* (pp. 25-36). Amsterdam: Elsevier, North-Holland.

(Manuscript received February 16, 1996; revision accepted for publication September 9, 1996.)