Interactive User-friendly Interfaces to Statistical Packages

A. M. R. Smith

Biometrics Department, Beechams Pharmaceuticals, Sub-Group Headquarters, Biscience Research Centre, Great Burgh, Epsom, Surrey KT18 5XQ, UK

L. S. Lee and D. J. Hand*

Biometrics Unit, Institute of Psychiatry, DeCrespigny Park, London SE5 8AF, UK

This paper outlines some of the ways in which the widespread availability of computers may be used to lead to better statistical data analysis. Attention is focused on the necessary properties of the interface between the user and that part of the program which does the actual analysis. An illustration is given using a program developed by the authors.

INTRODUCTION

The development of the digital computer has, over the past two decades, completely changed the face of statistical practice. Statistical analyses which once required considerable investment of resources can now be performed in minutes or even seconds at the touch of a button or insertion of a keyword. Furthermore there is no reason to suppose that this rate of change will slow down. Falling hardware prices and more general availability of machines means that more and more people have access to computers.

This hardware boom is being followed by a software boom, and for us the question is: what form will this boom take as far as statistical analysis goes? A glance through recent *Statistical Computing* sections of *The American Statistician* reveals that speculating on this topic is quite a popular pastime.[1-3] It also reveals a number of common threads. Notable among these are the future development of graphics facilities, user-friendliness, and systems for interactive statistical analysis. These developments are, of course, quite distinct from the impact computers will have by making feasible statistical methods which were previously out of the question (just as multivariate techniques have grown in importance).

In this paper we look at some of the requirements of interactive analysis (though our conclusions could be applied, somewhat clumsily, to batch processing).

TWO PROBLEMS

There are now a large number of statistical packages in existence and this poses problems for potential users. Consider the expert (statistician) first. There is the obvious question of how to choose which package to use. This is usually solved by using the one which has already been learnt. Learning how to use new packages, which often have completely different control structures, can be rather tedious and is often given a low priority. Compounded with this is the fact that command languages are quickly forgotten unless they are regularly used. This thus militates against the use of several packages. Although in this paper our interest will be more on the problems of the naïve user, we shall also cast a glance towards the expert.

The naïve user (non-statistician) also has the difficulties associated with new packages—but doubly so. Frequently, learning how to use a sophisticated package will require learning a branch of statistics—requiring a time investment which cannot be made. Again the tendency will be to stick to one package, whether or not it is the best for the job.

The problem of which package to use is an obvious one. Less obvious is the effect that the command language will have on the statistical analysis performed. Take, as contrasting illustrations, the programs MULTIVARIANCE and SPSS. The former basically uses a system of number codes to describe the analysis to be performed, and to choose the analysis using only the manual[4] requires considerable knowledge and understanding of multivariate statistics (incidentally, those having difficulties coping with MULTIVARIANCE are referred to the excellent book by Finn and Mattsson,[5] which consists of a number of worked examples using MULTIVARIANCE). Needless to say, this discourages non-statisticians from using MULTIVARIANCE, despite the fact that it is a powerful and flexible program. They thus frequently end up using less powerful programs which may not suit the data so well (for example, the methods may require unjustified assumptions). Of course, some statisticians would regard the difficulty of using a program to perform complex analyses as a merit rather than a demerit since this deters the naïve from misunderstanding and misperforming such analyses. This leads us to SPSS.

SPSS,[6,7] in contrast to MULTIVARIANCE, uses a simple system of mnemonic keywords describing the type of analysis to be performed. Little statistical knowledge of the range and limitations of the techniques is needed in order to be able to perform analyses using SPSS. The risk, of course, is that inappropriate methods will be used because using them is made so easy. (To quote Hooke[8] 'Use (of statistics) has been replaced by overuse and misuse. Regression is being used in foolish

* Author to whom correspondence should be addressed.

CCC–0010–4620/83/0026–0199 $03.00

ways in the neighbourhood of almost every computer installation.')

We thus have two contrasting problems: first, how to enable naïve users to use programs which have a control language oriented to statisticians and require a high level of statistical expertise, and second, how to prevent naïve users from misusing statistical techniques which are available so readily and easily from those programs which have an easy control language.

The ideal solution (in retrospect) is to write the programs in the first place bearing these two problems in mind. (But note that nothing short of the existence of The One Program will ever completely remove the question of choice.) Novick *et al.*[9] go in this direction with their Computer Aided Data Analysis program which 'leads a user in a step-by-step manner through a data analysis, thus making it possible for relatively inexperienced users to perform complex analyses'. However, the pressures of statistical consultancy meant that we had to adopt some alternative solution, predicated on the facts that (a) many packages for analysis already exist and (b) writing such packages is a long-term effort. We therefore chose to concentrate on the central aspect of the problem, the *interface* between the user and the package.

The program which is described here concentrates primarily on the first of the two problem types (how to enable naïve users to write programs in a technical control language). The reason for this emphasis was simply one of demand: many researchers at the Institute require the methods available in the target package but few have the necessary expertise to use it directly. However, it is implicit in the work that some attention must also be paid to the second problem type. At its most elementary level this manifests itself by combinations of restrictions placed on the available solutions (so that analyses contravening basic principles cannot be made). Some statistical expertise is thus necessarily incorporated into our program. One consequence of this is that our program reduces the overall flexibility intended by the designers of the target package and focuses on the solution of a set of commonly occurring problems. In general, in a given application the idea is that use of particular features of the problem area is made to specify more precisely the class of eligible solutions. Our approach to the second problem type has thus been essentially pragmatic, moulded by the needs of our statistical consultancy service.

Our program is an interactive interface to the MULTIVARIANCE package, so, in the next section, we outline MULTIVARIANCE and its users. First, however, we look at an important trend in statistical packages. This trend influenced our approach to the interface.

The differences between the command structures of MULTIVARIANCE and a program such as GLIM[10] exemplify this trend in statistical computing. These differences arise as a consequence of the gradual transition from batch mode to interactive submission of jobs. In batch mode one tends to think in terms of card input and, because of this, fixed format program control commands have dominated many of the packages. However, once the user is presented with a terminal and a good file editor with which to create and submit his jobs the desirability for such rigid formats disappears. In its place we see, in newer high level languages and packages, the introduction of a syntax of commands and parameters, although remnants of the fixed format ideas still remain. Some programs, of course, have been left behind by this change—a problem which has been recognized by others. (For example, Bock's MULTIVARIANCE preprocessor converts a string of commands given in a defined syntax into the fixed format of MULTIVARIANCE control cards.)

The idea of a command syntax is a step in the right direction but it still confronts the user with a manual (of the syntactic commands) to be digested. Although this may be easier than the original manual, it will still require some mental effort for an inexperienced user (witness, say the GENSTAT manual.[11] An effort to ease this problem has been made by Alvey *et al.*[12])

Since terminals usually come attached to a processor of some sort with a certain amount of disc space available we can project along the continuum from fixed format through structured command syntax to the next stage. This will interact with the user in a way which requires the user to spend less preparation time merely learning how to use the package. The precise style of this program interfacing the user with the set of statistical procedures will depend on how much emphasis is placed on different aspects of the desired interface—for example, how much emphasis is placed on each of the two problems identified above.

THE PACKAGE AND THE USER

MULTIVARIANCE is a program for generalized univariate and multivariate analysis of variance, covariance, regression, and repeat measures. To quote Finn,[4] it 'will perform univariate and multivariate linear estimation and tests of hypotheses for any crossed and/or nested design, with or without concomitant variables. The number of observations in the sub-classes may be equal, proportional or disproportionate . . . The program performs an exact least squares analysis by the method described by Bock [Ref. 13].'

The program functions in three stages: a stage describing the data (input stage), a stage describing the between groups contrasts (estimation stage), and a stage describing which variables to use and whether to use them as dependent variables or covariates. Table 1 shows the general form of a MULTIVARIANCE program.

The level of complexity can perhaps be hinted at by observing that, for example, the input description card requires 7½ pages of the manual to describe the parameters to be encoded on it. (The values of these parameters are, as we have already noted, given by numeric codes.) Table 2 lists the parameters for this card.

It is important to emphasize here that the complexity resides not merely in the MULTIVARIANCE package but is intrinsic to multivariate statistical methods. When we remark below that an integral part of the usage of our interface is a consultation with a statistician this is because, without such a consultation, for most of our clients (see below) a sophisticated analysis could not be attempted.

Figure 1 gives a sample program for a repeat measures analysis using a univariate mixed model (from p. 86 of Ref. 5).

Table 1. Outline of MULTIVARIANCE job deck

Phase 1: Input	Comments
1. Title Cards	
2. Input description card	
3. Factor identification card(s)	
4. Factor level recode cards	Optional
5. Comments cards	Optional
6. End-of-comments card	
7. Variable format card(s)	
8. Transformation cards	Optional
9. End-of-transformations card	Only if transformations are used
10. Minimum/maximum values cards	Optional
11. Missing values key	Optional
12. Variable label card(s)	
13. Data	
14. Transformation matrix cards	Optional
Phase II: Estimation	
15. Estimation specification card	
16. Means key	Optional
17. Arbitrary contrast matrices	Optional
18. Orthogonal polynomial key(s)	Optional
19. Symbolic contrast vectors	If there is an analysis-of-variance design having between-group effects
20. Contrast reordering key(s)	Only if non-orthogonal analysis-of-variance
Phase III. Analysis	
21. Analysis selection card	May be repeated after 25
22. Variable selection key	Optional
23. Variable test card	Optional
24. Covariate (predictor) grouping key	Optional
25. Hypothesis test card(s)	Optional
26. End-of-job card	

```
REPEATED MEASURES ANALYSIS OF LONGITUDINAL
DATA—UNIVARIATE MIXED MODEL
   1  3  1                            1
SEX    2SUBJCT 36GRADE  4
COMMENT CARDS GO HERE
FINISH
(I1,I2,I1,1X,F5.2)
VOCABLRY
DATA CARDS GO HERE

   70  8  1            1            1
1,3,1*3.
GRADE
D0,D0,D0,
D1,D0,D0,
D0,D0,P1,
D0,D0,P2,
D0,D0,P3,
D1,D0,P1,
D1,D0,P2,
D1,D0,P3,
I1,35D1,D0,
I2,27D1,D0,
  1
1,1,3,3,62.
  1
1,1,1,1,1,1,1,1.
                                   STOP
```

Figure 1. A sample MULTIVARIANCE job deck (See Ref. 5, p. 86).

From the package we turn to the user. Researchers in a wide range of disciplines work at the Institute of Psychiatry, including psychiatrists, psychologists, sociologists, epidemiologists, neurologists, pharmacologists, and biochemists (and, of course, statisticians). The level of statistical background amongst these is varied, ranging from the totally ignorant to the relatively sophisticated. It is interesting to note that psychologists tend to have a better grasp of statistics than psychiatrists, reflecting, presumably, the greater emphasis on this subject during their early training and carrying obvious implications for medical research training in the light of the current concern over the level of statistical expertise exhibited in medical journals.

Table 2. Parameters on the input description card

1. Total number of measured variables in the input set
2. Number of factors
3. Data form code
4. Number of format cards to describe data
5. Code indicating whether variables will be transformed
6. Number of variables after transformation
7. Code indicating way in which transformation matrix will be given
8. Number of variables after transformation matrix
9. Logical unit number of input tape
10. Code for punched output
11. Code for input tape rewind
12. Standard deviation code
13. Code indicating whether data should be listed
14. Output spacing code
15. Data screening code
16. Optional output code
17. Number of records to be skipped before data are read
18. Minimum/maximum value code
19. Missing value code
20. Number of factors for which levels are to be recoded

For naïve users writing programs using packages is just one end of the problem. The other is, of course, interpreting the output when it arrives. If a simple *t*-test is being performed then this is straightforward (though, once again, its very ease causes problems—for example, the tendency for researchers to perform multiple *t*-tests and misunderstand the implications for the significance level). On the other hand, if a complex multivariate analysis is being performed then expert advice on interpreting the output (and on understanding it—and even finding within it the particular number you want) is essential. We have made no attempt to use the computer as an aid to interpretation of the output—one of the factors which distinguishes our interface from an expert system (see below). However, we have tried to incorporate automatically supplied meaningful labels to annotate some aspects of the output. These should give users a better chance of locating the relevant parts of the analysis, if not interpreting it. Note that had we been working interactively with the target package (rather than in batch mode) it would have been possible to give the user more guidance through the output.

Use of our interface program (which we have called BUMP—Biometrics Unit Multivariance Program) occurs in the following way. First, the researcher discusses his/her work with a professional statistician and they decide on the appropriate experimental design and form of analysis. This is in part to make sure that the correct design and analysis is being used and that the researchers grasp its aims, and is in part necessary to familiarize the users with the type of question that the BUMP program will ask to gain its description of the data, design, and the hypotheses the user wishes to investigate. However, it is important to note that the terminology used by BUMP is much less statistically oriented than that of the original MULTIVARIANCE program.

From the problem description BUMP makes decisions about aspects of the analysis. These decisions are, of course, consequences of—deductions from—the nature of the data available, the questions the researcher wishes to investigate, and technical aspects of the design of the experiment. For example, if unequal numbers of subjects per cell are specified in a design, BUMP will automatically generate the re-ordering of contrasts required for independent hypothesis tests. In other areas it will complete the description details required by MULTIVARIANCE from the information it has to hand. It maintains for the user a consistency between the various sections of the MULTIVARIANCE input and also between the labelling used in the output.

When the data have been collected and when information about the design and the data have been constructed by BUMP into a MULTIVARIANCE program, this is automatically dispatched to the central computer for processing. (Note that no cards need to be punched. We have serendipitously discovered that one of the pleasurable features of using BUMP is that it avoids the attendant frustration of punching numerical codes in wrong columns.) The final\ stage occurs when the researcher and the statistician discuss the results of the MULTIVARIANCE analysis. Note that it is the fairly mechanical task of writing the MULTIVARIANCE program once the form for the analysis has been chosen which is automated.

BUMP

Programs which interact directly with users who are neither computer experts nor experts on the material contained within the program must possess several attributes:

(i) They must not bore the user. For example, they must not ask the same question twice, even if the answer has implications for two different parts of the output. This needed attribute can lead to programming complications.

(ii) A limited facility for deduction is desirable. If the program knows the number of covariates and the number of dependent variables it should be able to add them to get the total number of measured variables.

(iii) The order in which the questions are asked should appear logical to the user, so that the user sees some kind of pattern and will have an idea of what sort of questions to expect next. If not, if the questions appear to hop from topic to topic, then the user can become irritated. This desirable factor can also cause programming complication.

(iv) The program must have considerable ability to explain the meaning of any terms it uses and to expand any questions it asks. Note that, unlike expert systems, our program did not need to be able to explain *why* it was asking a question.

The fact that we were designing our program as a front end to an existing program meant that things were made easier for us in several ways. There is the obvious point that we did not have to be concerned with the algebraic and arithmetic details of actually how to solve the problems. But there also is the much more interesting point that we had a fairly well-defined structure that our program should build, namely a MULTIVARIANCE job deck as illustrated in Table 1. Sometimes we felt it was appropriate to ask our questions in an order which did not correspond to the order of parameters in the MULTIVARIANCE job deck: cards belonging to separate phases of the program are not totally independent. For instance, the second parameter of the input data description card was collected from information obtained at a later stage of the design description. Usually, however, we were able to keep fairly close to the basic pattern indicated in Table 1. The point is that we did not have to be too deeply concerned with the internal representation, that is the structure of BUMP's output. The design of this structure, the most difficult phase in writing such programs, was in a large part already done for us.

Our program (which was written in FORTRAN) was modular in form, as follows:

1. **Initialization**
Most of the communication between modules is done via common blocks which are initialized here.

2. **Introduction**
Giving instructions on how to use BUMP.

3. **Job identification**
Requesting details of the user account number and any description or annotation of the job.

4. **Data**
Obtaining details of the data source (see below).

5. **Transformation**
Obtaining details of any transformations which the raw data have to undergo.

6. **Between subject design**
Details of the analysis of variance classification factors.

7. **Keyboard data entry**
The user may analyse data stored in a previously prepared data file or input directly from the keyboard. In the latter case it is entered at this stage.

8. **Within subject design**
Details of the relationships between dependent variables.

9. **Means key**
Details of which groups' means are to be output.

10. **Analysis**
Choosing the variables to be analysed.

11. **Generate file**
Integrating the information obtained by the above modules into a file to be despatched for processing by MULTIVARIANCE.

Some aspects of the program are worth mentioning. First, we have adopted a conventional programming methodology. We did not, for example, attempt to use a production system[18] approach because we felt that the flexibility of such a method was unnecessary for our application. Again this is a consequence of the predefined basic shape of the structure our program had to build.

Secondly, on the whole our program is not menu driven. Menus are a common choice for this type of program and if we had been designing an integrated user-interface and analysis program we might well have used such an approach. However, once again because this was merely a front-end things were easier for us if we avoided that approach. Note also that aid is easier to give—explanations can be more specific—when individual questions are asked. Occasionally, for example in selecting a transformation from a table of possibilities, menus were clearly the simplest approach and we did adopt them.

Necessary attribute (iv) above was implemented by permitting the user to ask for clarification whenever a question was asked. We found a highly effective way to provide extra information was to split the screen into two halves, using the lower half for the system questions and the user's responses and the upper half for explanations and expansions. An arbitrary number of help frames were permitted for each question, the user being instructed that the next one could be called up by typing H on the terminal.

Data for the output MULTIVARIANCE program to work on can be supplied in one of three ways. It can be a previously created disc file either locally in our computer centre or remotely at the University of London Computer Centre (where MULTIVARIANCE resides) or it can be fed in directly from the keyboard while our interface program is running. In each case the data file is automatically attached to the created MULTIVARIANCE program and dispatched for processing. The reader may be puzzled regarding the positioning of the keyboard data entry module between the 'between subjects design' and the 'within subjects design' modules rather than with the data sources information in the 'data' module. This is due to the fact that the way in which the data is to be prepared (the data 'form' in MULTIVARIANCE terminology) is not asked until the between subjects design stage because at this stage the program has an idea of the location, size, and design structure for the between subjects analysis and can use this as a template against which the user enters his data. (The program, for example, can go through the design cell by cell.) Although this seems to cause little difficulty it would be more in keeping with atrribute (iii) and would be aesthetically more pleasing if it occurred with the 'data' module.

Figure 2 illustrates some of the questions and help frames.

Q. Do you know how to use this program (N or Y):
A. N

Help frame:
R2:
This program will ask you questions about your study and your aims and will prepare a MULTIVARIANCE deck according to the answers you give.
Possible answers are:
A number, followed by the RETURN key.
Y, meaning YES, followed by the RETURN key.
N, meaning NO, followed by the RETURN key.
H, meaning HELP, followed by the RETURN key. This will lead to some explanations being given in the top half of the screen.

Q. Do you know the form your data uses? (Y or H):
A. H

Help frames:
R10:
Nine forms of data are accepted by MULTIVARIANCE. They are explained below.
Press H to obtain further explanations:
FORM 1. Raw unsorted data, each observation with its own cell identification information. This information to come first on the card(s) for each subject.
FORM 2. Raw data sorted by cells, each cell with its own header card.
FORM 3. Raw data sorted by cells, not header cards.
FORM 4. Within group variance-covariance matrix and mean-frequency summary data.
R11:
FORM 5. Raw unsorted data to be read from an independently prepared binary tape.
FORM 6. Raw data grouped by subclasses to be read from an independently prepared binary tape.
FORM 7. Within-group correlation matrix and mean-frequency summary data.
FORM 11. Same as 1, but data sorted into groups by program.
FORM 12. Same as 5, but data sorted into groups by program.
If you are not familiar with MULTIVARIANCE we recommend you use data form 1.

Q. Is there only one subject per cell? (Y or N or H).
A. H

Help frame:
R26:
Oh, come on. Surely you can tell if there's only one subject per cell. I suppose you're having some problem over the word 'cell'. This is simply one of the groups in the cross-classification of the factors. For example, if there are two factors, SOCIAL CLASS with five levels and SEX with two levels, then there are 10 (=5 times 2) groups in the cross-classification. That is, there are 10 cells. So if you have only ten subjects then obviously you have only one subject per cell.*

* At this stage the program already knows that there is an equal number of subjects per cell.

Figure 2. Some questions and associated help frames.

CONCLUSION

The scope of the project was initially confined to strictly factorial designs for both between and within subjects designs (as well as the presence of covariates). Tests are carried out assuming fixed effects factors. It has taken 12 man-months to arrive at the stage where most options that are frequently used have been implemented. Work is currently in hand to expand the design set and hypothesis testing possibilities using the Hasse diagrams and related algorithms discussed by Taylor and Hilton.[14]

Although the existence of the target structure meant that we had few design problems one or two arose, mainly regarding the ordering of questions in such a way as to appeal to the user.

The fact that the job was relatively small and well-defined meant that we were tempted away from writing a general and flexible program. To what extent this was a mistake remains to be seen—it clearly means that it will be more difficult to modify the program in the future.

One improvement in the program would be to make it easier for the experienced user to use. For example by permitting the answers to several questions to be given at once, without actually asking the questions.

Norusis and Wang[15] have tackled this problem in their SCSS Conversational System, which 'operates in three prompting styles: verbose, normal, or terse. The user is free to choose among them and can vary the prompting style at any point in the session. Normal prompts ask a complete question. Verbose prompts ask the same question and give suggested responses or helpful information. Terse prompts ask abbreviated questions (a maximum of eight characters).'

Another limitation of our program is that it works only on information provided by the user. It does not look at the data itself. (For an example of a situation where this might be valuable, consider a pre-processing program which automatically checked any normality assumptions.)

Our program should be distinguished from expert systems.[16,17] BUMP does not tender advice, and nor is it able to explain why it has made a decision (this would be inappropriate). It simply automates a task which is fairly mechanical—once the thinking has been done.

To conclude, we would like to remind the reader of the desirability for statistical analysis programs to be designed bearing in mind the two problem types outlined above. For those packages for which it is easy to write programs it should be made harder, in the sense that some check on violated necessary conditions should be made. And for those packages for which it is hard to write programs it should be made easier so that naïve users can successfully carry out appropriate complex analyses.

Acknowledgement

The research described in this paper was supported by grant number 7040 from the Bethlem and Maudsley Research Fund.

REFERENCES

1. M. E. Muller, Aspects of statistical computing: what packages for the 1980's ought to do. *The American Statistician* **34**, 159–168 (1980).
2. J. M. Chambers, Statistical computing. History and trends. *The American Statistician* **34**, 238–243 (1980).
3. T. J. Boardman, The future of statistical computing on desktop computers. *The American Statistician* **36**, 49–58 (1982).
4. J. D. Finn, *MULTIVARIANCE version VI*, National Educational Resources, Chicago (1977).
5. J. D. Finn and I. Mattsson, *Multivariate Analysis in Educational Research*, National Educational Resources, Chicago (1978).
6. N. H. Nie, C. H. Hull, J. G. Jenkins, K. Steinbrenner and D. H. Bent, *Statistical Package for the Social Sciences*, McGraw-Hill, New York (1975).
7. C. H. Hull and N. H. Nie, *SPSS Update 7–9*, McGraw-Hill, New York (1981).
8. R. Hooke, Getting people to use statistics properly. *The American Statistician* **34**, 39–42 (1980).
9. M. R. Novick, R. M. Hamer and J. J. Chen, The computer-assisted data analysis (CADA) monitor. *The American Statistician* **33**, 219–220 (1979).
10. R. J. Baker and J. A. Nelder, *The GLIM System*, Numerical Algorithms Group, Oxford (1978).
11. N. G. Alvey *et al.*, *GENSTAT*, The Statistics Department, Rothamsted Experimental Station, Hertfordshire (1977).
12. N. G. Alvey, N. Galwey and P. Lane, *An Introduction to Genstat*, Academic Press, London (1982).
13. R. D. Bock, Programming univariate and multivariate analysis of variance. *Technometrics* **5**, 95–117 (1963).
14. W. H. Taylor and H. G. Hilton, A structure diagram symbolisation for analysis of variance. *The American Statistician* **35**, 85–93 (1981).
15. M. J. Norusis and C-M. Wang, The SCSS conversational system. *The American Statistician* **34**, 247–248 (1980).
16. D. Michie, Expert systems. *The Computer Journal* **23**, 369–376 (1980).
17. D. J. Hand, Artificial intelligence. *Psychological Medicine* **11**, 449–453 (1981).
18. D. A. Waterman and F. Hayes-Roth, *Pattern-directed Inference Systems*, Academic Press, New York (1978).

Received July 1982