STATISTICAL PACKAGES

E. J. Carney*

Biometrics Unit, Cornell University
Ithaca, New York

BU-495-M February, 1974

Abstract

Statistical packages provide a variety of data processing capabilities for use by statisticians, consultants, and investigators. Features of several packages are reviewed as examples of differing approaches to providing data processing help to users with differing problems. Some of the difficulties attendant upon the use, management and design of packages are explored, and a few suggestions made. Particular emphasis is given to the viewpoint of the statistical consultant.

Quite a few years ago I had occasion to consult with a lawyer friend of mine. Since he lived nearby, he stopped off to see me on his way home from the office. When he got to my house, he was laughing and shaking his head about something which had just happened to him. A man had come in to consult him about a problem with a loan. The man had borrowed $200, and was to pay it back, with interest, $50 at a time. Each time he made a payment the lender calculated the new balance by adding the payment to the principle. After paying $250, the man now owed $450. He did not really understand principle and interest, but he felt that something was wrong.

I suppose all lawyers could tell stories like that one. I don't even know if it is true. In fact, on several occasions the same lawyer told me how

---

*On leave from the University of Rhode Island

much he liked my wife's apple pie. Since my wife makes excellent apple pie, I wasn't much surprised, but when I mentioned it to her, she told me that he had never tasted her apple pie. It was just something he told people. I can tell you a story, however, which I know is true (you may judge for yourself). It is something like the one my lawyer friend told me. One day a lady came in to see me. She was rather angry, feeling that the people who ran the computer center were not treating her fairly. She was a new faculty member and had been conducting an investigation involving nutrition, diets, or something like that. Anyway, she had a large amount of data and wanted to use the computer to analyze it. She had gotten herself an account number, had all the data punched onto cards, and taken it to the computer center. There she left it on the counter at the place for incoming jobs. She went back a day or so later, but nothing had happened. She felt she had been treated rather shabbily by the people in charge of the computer. When she protested they sent her to see me. Her story rather amazed me. Not because of her ignorance about the computer. She probably knew as much about that as I did about nutrition. But because of her faith. How did she suppose that the computer could find out what she wanted done with her data? Of course I explained to her that what was needed was a computer program, and we must have found what she needed somewhere. Perhaps we used some package program.

This story illustrates the gap which the statistical package is designed to fill. The investigator had the data and the account number. The computer was ready with its high speed arithmetic registers, megabytes of core, disks, drums, tapes, readers and printers waiting, champing at the bits. There are, of course, several options available in providing computer programs for the statistical processing of data. One option is for a program to be written each time one is needed. This option is not considered very seriously any

more. For one thing writing the program might take as much time as analyzing the data by hand. Furthermore, whoever wrote the program would probably keep it and eventually it might be used again. After awhile a group of programs would accumulate, which would be publicized by word of mouth and rumor and so a sort of informal package would be produced. Other programmers would have their programs too, and so we would have several such informal packages floating around. Some of the programs would occur in several of them. None of the programs, or very few of them, would have been planned as general purpose programs, and so they would be awkward to use on new data sets requiring the same analyses. They would often need to be modified, and this would produce still more redundant programs in the system. Another problem is that almost none of these programs would be well documented. The person who wrote the program might be able to figure out how it should be used but that would be quite hard for others, and probably impossible for non-programmers. This informal package arrangement would therefore waste time for a lot of people, both users and programmers.

Supposing that a statistical package will be used for processing data, there are several choices with respect to the type of package to be implemented. One type of package consists of a set of subroutines. The IBM Scientific Subroutine Package (1) is as example. Another is IMSL (2). IMSL stands for International Mathematical & Statistical Libraries, which is the name of the company which distributes this library. User organizations can subscribe to this library for $840 per year, which seems little enough for association with an organization with such an imposing name. The user of a subroutine package has to prepare his own main programs in some higher level language, usually FORTRAN. These main programs often must take care of the input and output aspects of the work. In many cases this is not a trivial part of the task.

One way such library subroutines can be profitably used is by applications programmers in providing general purpose programs which are tailored to the needs of particular users. Such libraries provide a set of standardized components which can be incorporated into other programs. In addition to the benefit of not having to write major parts of the program, there is the attendant advantage that the components have known properties: They have (supposedly) been extensively tested, they are based upon algorithms whose numerical properties have been studied. In the case of IMSL the algorithms used are claimed to be the latest and best available. The disadvantage of subroutine packages is, of course, that they cannot be used without some programming. If the end user is not himself a programmer then he must avail himself of the services of someone who is.

A third type of statistical package is that which consists of a set of independent general purpose programs. Such programs may have a few common features, a first input card which is the same from program to program, a somewhat similar input deck set up, etc. But each program is essentially separate and complete, with its own parameters, input, and output. Perhaps the best known example of such a package is the BMD series of programs. The programs, distributed at a nominal charge by the Health Sciences Computing Facility at the University of California, Los Angeles, have been used throughout the country for the past 12 or so years. Programs in the series have been occasionally revised, replaced, deleted, consolidated, and added to but the system has changed only gradually in that time. Part of the stability of the system is due to the reluctance of users to give up programs with which they are familiar, and the sympathetic response to that inertia by the facility. The programs are quite well documented in the publication Biomedical Computer Programs (3). The most recent version of this publication (January 1, 1973)

combines the previous BMD and BMDX series programs to form a single series of 53 programs covering many generally used statistical techniques. The fact that each of these programs is complete and that the manual gives detailed instructions for card preparation allows them to be used by non-programmers. Though each program is written in a quite general way its options controlled by parameter cards as part of the input for a run, the fact that programmed intervention by the user is not necessary carries with it the concomitant that it is not, generally, possible. Thus use of the package for non-standard analyses, running a second analysis on the results of the first, obtaining output which is not part of the usual output from a program, and other such novelties which crop up from time to time may not be feasible, or, at best, may be awkward.

Another type of statistical package is the statistical system. Like other package types such a system provides algorithms for a variety of analysis capabilities, usually embodied in subprograms, but also provides a control program which reads and translates instructions from the user and uses the information thus made available in deployment and control of the various algorithms in the system. The objectives of such a system, besides that of providing programs to perform the various statistical analyses, will be to give the user additional control over the programs available, and to do this in a way which is convenient, easily learned, and easily remembered. Such a control program will require some language processing features, and will have to have some file management capabilities as well. A problem with the design of such a system is that the goals of flexibility and ease of use are antithetical. A given user wants to do his thing but may be very unhappy that he has to learn a language designed to let a lot of other people do other things. Of course if he himself wishes to do many different analyses he may be happy to invest the time required to learn the system. On the other hand, if those with data to be analyzed refuse

to invest the time required, and instead employ others to handle their analyses, the objectives of the system may be partially defeated. One day I happened to mention to a friend of mine that a language I was using had a 'GET' macro. Since my friend was not a computer sophisticate he retorted with a "whatif": "What if I punched 'GIT' instead of 'GET'?" I told him I thought that that would cause an error message. He speculated that the error message might say "You dumb hillbilly, that's 'GET' not 'GIT'!" The language which is natural to one class of users is not necessarily natural to another, and if a statistical system is designed to be used by people who know about statistics--the meaning of its technical terms--but the system is really used by others, for example programmers, such use may be less than optimum.

Two statistical systems which embody some of the features which I have been discussing, each of which is in fairly wide use, are SAS (4) and SPSS (5). SAS, which stands for Statistical Analysis System, is distributed by the Department of Statistics, North Carolina State University, at a cost of $650 per year. SAS is in part sponsored by participating schools of agriculture and so its capabilities are likely to include many of those useful in analysis of data obtained in agricultural research. The system is not limited to these facilities, however, and in any event the same techniques are useful in analyzing data from other sources.

As an example of the "language" which SAS requires for program control the following represent the statements which I used to run a fairly routine analysis of variance using the SAS regression procedure.

```
DATA SQUARE;  COMMENT  4 X 4 COMPLETELY ORTHOGONAL SQUARE;
INPUT R 1 C 2 L 3 G 4 Y 7-9;  CARDS;
11111  44
```

```
12222  48
   :
   :
44123  52

PROC PRINT;

PROC REGR;  CLASSES R C L G;

MODEL  Y=R C L G / PREDICTED DUMMYB;
```

The SPSS system (Statistical Package for the Social Sciences) was designed particularly for use by social scientists. It contains extensive capabilities for handling frequency counts, and programs for factor analysis and others. The present versions of the two systems somewhat complement each other but I suppose that in the future each will incorporate many of those parts now missing. The SPSS "language" is more restrictive and somewhat less natural than that of SAS; certain things have to go in certain columns, and options for various routines are specified by number rather than by naming them. This is a minor criticism, I suppose, but I can remember fairly easily that if I want predicted values in regression I should write 'PREDICTED', using SAS, but I will probably forget that if I want Kendall's tau B from the SPSS cross-tabulating program I should write 'STATISTICS 6'. An example of the program control statements needed for a simple SPSS job, (copied from the SPSS manual) is shown below.

```
Col. 1                  Col. 16
RUN NAME                DEFINE, CROSSTABULATE AND SAVE
FILE NAME               FACSTUDY, SURVEY OF FACULTY PARTY PREFERENCES
VARIABLE LIST           PROF,PARTYPRF,AGE,SEX,RELIGION
INPUT FORMAT            FIXED (A8,1X,A1,1X,F2.0,1X,A1,1X,F1.0)
PRINT FORMATS           PARTYPRF,SEX(A)/AGE,RELIGION(0)
# OF CASES              20
INPUT MEDIUM            CARD
VAR LABELS              PROF,FACULTY MEMBER'S NAME/
                        PARTYPRF,POLITICAL PARTY PREFERENCE/
                        AGE,AGE IN YEARS
VALUE LABELS            PARTYPRF ('C')CONSERVATIVE('L')LIBERAL('S')SOCIAL CREDIT
                        ('N')NEW DEMOCRAT('R')NOT GIVEN/
                        SEX ('M')MALE('F')FEMALE/
                        RELIGION (1)PROTESTANT(2)CATHOLIC(3)JEWISH(4)OTHER
```

```
MISSING VALUES        PARTYPRF('R')/AGE(0)
CROSSTABS             SEX BY PARTYPRF
OPTIONS               3,5
STATISTICS            1,3
READ INPUT DATA
    ... DATA CARDS ...
SAVE FILE
FINISH
```

In the example shown the SPSS language appears to be more long-winded than does SAS. This is not necessarily true, because many of the cards shown for the SPSS example are optional. Furthermore the SAS example could have been made to appear longer. I might have used the class names: ROWS,COLUMNS, LATIN,GREEK for the orthogonal square, instead of the names: R,C,L,G. Several of the optional cards in the SPSS example, VAR LABELS and VALUE LABELS in particular, illustrate the capability of the language for making the output more easily read and understood. Such features might be called "report generation" capabilities, and while they are not extensive in SPSS, they do represent, perhaps, a step in the right direction.

Both SAS and SPSS systems provide the user with programming statements which may be used to edit and transform incoming data. These statements are similar to those found in a programming language such as FORTRAN. In general they are applied to each observation as it is read and allow the values of variables to be examined, changed, combined to form new variables etc. The editing capabilities allow observations to be included or excluded from the data set to be processed depending upon the values of certain variables or combination of variables. My impression of the comparison of the capabilities of these two languages is that SAS is more flexible but that SPSS would be more convenient for certain types of tasks, for example in recoding questionnaire data.

Both systems also have some file management capabilities. SAS allows data sets to be concattenated, merged, sorted and broken into subsets. The SPSS system has more limited capabilities of this type but does allow a subfile structure to be imposed upon data files, with the possibility of processing subfiles together or separately, adding subfiles, deleting subfiles etc.

All of the packages which have been discussed so far are batch systems. That is, the control cards, program cards, and data are submitted by way of some computer terminal, perhaps reading data off a disk or tape, and later results come back on a printer, or are stored away somewhere. No one intervenes while the job is being run. Interactive packages are also available. One such package is STAT-PACK 2 (6), available from the University of Alberta, and consisting of a set of about 50 routines to be used with the interactive terminal language, APL (13). A second type of interactive package is one which asks the user questions. I have had some experience with one such system, CALL/360 STATPACK (7), which is distributed by IBM. This system allows data sets of up to 100 observations on 12 variables and performs many common statistical procedures. To use this system you sit at a terminal, sign on, and then type "run ****statpack". In a few seconds the terminal responds with the question: "are you a statpack expert?" If you now type "yes", you become an expert. Another system which can be used in an interactive way is the new BMD P-series (8). This series contains a set of subprograms, some main programs, and includes a parameter language translation program. It is possible to use this system interactively in a couple of different ways, one similar to the way it would be used in batch operation, entering a paragraph of control information at a time, and one similar to the way CALL/360 STATPACK is used, entering each item of control information in response to a specific question. Interactive systems which ask questions are very easy to learn to use. Of

course, since data is usually entered at the terminal their use is limited to small data sets. It may be possible to obtain data from some auxiliary storage device, but even so if there is a large amount of output, or if the run time is fairly long, interactive computing will probably be less convenient than batch operation. An exception to this may be analysis of data using graphical display devices (11).

We have looked briefly at representatives of several types of packages which may be used for statistical data processing. There are many more which might be mentioned and those I have chosen to discuss are those with which I am familiar, or which seem to be most widely available. I do not mean to imply that these are the best that can be obtained. The availability of such a large number of statistical packages presents quite a problem to the managers of computer service facilities. At a university, with new faculty frequently arriving from other universities where they have become familiar with the local version of some package, there is a constant pressure to make more and more such systems available. In addition to the cost of acquisition of the programs, this also increases the need for auxiliary storage facilities; systems programming support for installation and maintenance of the systems; and applications programming support so that the user can be taught how to use the package, manuals acquired, updated and made available, etc. The wide choice of packages available also complicates the life of the statistical consultant. In fact, any packages at all do this. Here we have the statistical consultant, and out there somewhere is an experimenter who is generating data. When he has his data in hand what will he do? Will he go to the statistical consultant with it, or to the computer services facility? There is some chance that he will do the latter. Now he may be introduced to the statistical package. If he knows enough statistics, if the manual for the package which is in use is

intelligible to him, if the consultant he talks to at the computer facility is knowledgeable about statistics, or at least knows enough to know what he doesn't know, everything may turn out all right. But it is possible that one or more of these requirements may not be met. I think that it may sometime have happened that the method of analysis applied to a set of data was determined by the type of processing program available rather than upon the basis of statistical considerations (12).

In view of these problems, the statistician may feel justified in his previous vague feeling that the computer and the computer types associated with it are a subversive influence, coming between the experimenter and his data. After all, how can he <u>really</u> know what is going on if he hasn't added up the raw sum of squares five times on his good old desk calculator? And gotten five different answers. Suppose our experimenter, either on his own, or upon the suggestion of someone at the computer facility, does turn up at the office of the statistical consultant. It is to be hoped that he will get good advice with respect to the statistics he should use. But will he also get advice on how to prepare his data for processing, what program or package to use, where to find the necessary manual, or at least, who to see to find these things out?

While those statistical packages which consist of sets of subroutines do require some programming for their use, they also require knowledge of statistics if the use is to be appropriate. The other types of packages, those consisting of sets of independent main programs, like the BMD series, and those which are statistical systems, like SAS and SPSS, are primarily designed for use by people with a knowledge of statistics and do not require programming knowledge as a prerequisite for their use. Both SAS and SPSS do have program

statement capabilities, and these may be employed to use the systems in more powerful and flexible ways, but are not essential for routine work. A statistical consultant who becomes familiar, perhaps even proficient, with such a system will be in a much better position to advise his clients who have data to be processed. In addition to providing better consulting help to his clients, he will aid in the establishment of better channels for users with data to be analyzed, lessening the probability that the statistician will be left out of the process. His familiarity with the packages will also put him in a better position to advise those charged with the management of computer service facilities on the acquisition, retention and deployment of such packages. He may also have the opportunity to influence the design of such systems. Of course, there is the danger that he will become another computer nut, design his own system, and thus help to complicate the problem.

In view of the fact that there are many statistical packages on the market, an obvious question is, "Which one should we use?" The answer must depend upon who you are. If you are someone with data to analyze the answer will depend upon whether you need to use the computer frequently, with many different kinds of problems, or always need the same analysis, have small data sets or large data sets, have access to a typewriter terminal, and so forth. Are you a programmer? How much do you know about statistics? If you are the systems programmer you may have different criteria from those of the user. The statistical consultant will have still other views. Since most of my contact with statistical packages is in this latter capacity, although I have also sometimes been involved in design and implementation of packages, and sometimes in the management of their use, and even once in awhile have been a user with data of my own, it is from the point of view of a statistical consultant, attempting to help people with data to be analyzed (or who will eventually have

data to be analyzed, leaving room for the remote possibility that someone will come to se me before collecting his data, that I would like to answer that question.

In my opinion there is one outstanding criterion for a statistical system or other package. Now, of course, the system ought to contain algorithms to do the kind of computing that is needed, that goes without saying, but given that the application matches the system fairly well, I think there is one aspect of the system which should be given consideration before every other. I wonder if anyone would like to guess what that criterion is?

I feel that the most important feature of a statistical package, especially to me, in the capacity of a consultant, is its documentation. The documentation is important in several ways. To put first things first, if the system is well documented from the standpoint of how it is to be used, what cards are necessary, etc. I can often send my client off with a manual and he can find out for himself how to use it. This not only saves time for me, but is probably more accurate and informative for him. And it is written down, so he can read it over again if he forgets. He may not be able to understand the manual completely, and even if he does, he may wish to verify some things with me, but if the program is well documented it will save me a lot of time.

In addition to instructions of how to do it the documentation for each procedure should contain some explanation of the statistical basis for the method. I have taken several statistics courses, but they did not cover every method of analysis. If someone has programmed some method, tomorrow or the next day someone will come to see me wanting to use it. If I haven't seen the statistic before, I do not hesitate to admit that, but I would like to have some justification for my advice to use or not to use it, and I don't want to take the time to derive it from first principles. I want a reference.

A third thing I want the documentation to provide me is description of the algorithm. This does not have to be a detailed flowchart, but it should identify the algorithm sufficiently so that some indication of its numerical properties, suitability for large numbers of observations, and so on, can be obtained. I would like to know in which parts of the algorithm double precision calculations are used. References will again be welcome. You can tell me all day that your algorithm is the best one ever, but I want to know what it is, and why you think so, and what other people think. And I do not want to perform a Monte Carlo experiment to find out if it is any good (but if you have performed one I would like to know the results). You may think that I am not entirely pleased with documentation of statistical packages I have used. That is true, although the systems and packages I have mentioned today are fairly well documented, with some shortcomings. Of the systems I have mentioned perhaps SPSS has the best documentation. This opinion may be partly due to the fact that the printing quality is much better in the SPSS manual than, for example, in the SAS manual. I would guess that the documentation is one of the reasons why SPSS has had such wide acceptance.

Preparing the documentation of programs is not an easy task. One reason for this is that those working on the design and preparation of a statistical package may be very much involved in the problems of selecting and coding algorithms and of debugging and testing programs. They may have little interest in writing the necessary manuals. Secondly, preparing a manual which gives all the specific details needed, and remains readable, is a formidable task. The various facets of a program are interrelated. Wherever one starts to describe it, it seems necessary to use terms and concepts which have not yet been defined. The result is that the manual either seems to keep saying the same thing over and over again, or it is necessary to keep looking ahead

to find the meaning of something which is needed here but is discussed there. The task of writing a manual for a statistical system is further complicated by the fact that it is necessary to meet the needs of a variety of different kinds of users. Some will want to use only one of the many capabilities of the system, and do not want to wade through a lot of material which has nothing to do with their job. Others need to learn the system thoroughly, because they plan to use it on a variety of tasks. They would like a systematic development of the necessary information. Despite the problems in preparing adequate documentation, it can be done. It is my opinion, some would dispute it, that the IBM language reference manuals have improved greatly over the past few years. I think that the main necessity is that the documentation of statistical packages be given a very high priority. Not only should the manuals be planned and written with the same conscientious effort that is devoted to building the system, they should also be subjected to a critical review and editing process. It seems likely that the abundance of packages will, because of its competitive pressure, result in improved documentation. If you want your system to be used look to its documentation.

What other criteria should guide us in selection of a computer package or system for statistical processing? An important one is reliability. Computer hardware manufacturers sometimes say, modestly enough, "We have never detected an undetected error." It is in this same spirit that statistical packages need to be reliable. It is too much to expect, for many algorithms, correct results for every set of data. Near singular coefficient matrices, for example, are certain to cause trouble in algorithms requiring solution of linear systems. When algorithms break down, because of peculiarities of the data, or for whatever reason, this breakdown should be detected, and appropriate messages made part of the output. Algorithms which depend upon the data sequence for correct

results should check this sequence. There are various techniques available for checking data quality, but few packages include these.

The type of package which will be most useful, as has been suggested earlier, will depend greatly upon the type of computing which is to be done. If only one type of analysis, or a very few types, are contemplated, it is probably easier to use programs from a main program set, such as the BMD series. There are two reasons for this. One is that it will only be necessary to read the instructions for that one application, and not necessary to learn a "language" which includes capabilities for a lot of other things. A second reason is that the single main program will probably be more efficient for its single application than will a system which makes provision for many applications. There will be less system overhead. On the other hand, for many and varied applications a statistical system like SAS will be appropriate. When a great deal of file manipulation is necessary, combining files, taking them apart, and so on, the SAS file manipulation capabilities seem to be indicated.

There are times when special purpose programs are needed. For example, it may be desired that the output from a program should be in a form which can be photocopied directly as a published report. In this case a new program can be written, perhaps making use of subroutines from a subroutine set like IMSL. Problems with a heavy input or output requirement, or with extremely large amounts of computation should have the benefit of examination by professional programmers who are familiar with the idiosyncrasies of the system upon which they will be run. At the University of Rhode Island we have routinely watched for long running jobs, and have sent systems programmers to consult with those users. Sometimes fairly simple reprogramming may save a great deal of computer time on subsequent runs. This of course benefits the user, if he is paying for

the job, and also benefits other users, by reducing the overall load on the system. Sometimes it has had even greater benefits than for the individual job by bringing about changes in general use software systems, thus helping all users of those systems.

To conclude this discussion I would like to suggest some developments which may take place in the design of statistical software and will perhaps improve its usefulness.

All of us deplore redundancy in software systems, but despair of doing anything about it. In one limited area perhaps something can be done. Suppose I am a user or consultant with several packages available to me. I may feel that one suits me better for some purposes and others for other purposes. It would be nice if I could use different systems in successive job steps, using the same data files coming and going. This can be done at some installations, for some systems, because software to do it has been produced locally. I think that some coordinated attack on the problem of file management for statistical systems could be made which would result in very free interchange of files between systems, and better file management for all.

I have said above that one of the objectives of a statistical system is to give the user additional control over operations, including the capability of performing a sequence of analyses, using output from one analysis in the input to the next. This can be done rather easily with subroutine sets since the user writes the main program, but except for certain special cases is not easy with other types of packages, because the user does not have access to the output of an analysis. I think the problem of making computed results available to the user, for further processing, decision making, preparing special output, and so on, deserves attention. The OMNITAB system (9) gives the

user broad control over the location of results in memory, by asking the user to specify the "column" of his "work sheet" where they are to be stored. This is at least a partial solution to the problem. Another approach would be to embed the statistical system in a general purpose language, giving the user some information or control over the names of resultant values, and access to them. But one does not want the statistical system to degenerate into a general purpose language. Perhaps there are more elegant solutions of the problem to be found.

A related problem is that of "report generation". Why should I be satisfied with the standard peculiarly unintelligible output of a statistical system? Can the user be given easily understood and easily operated editing facilities?

I have mentioned a few of the developments which users of statistical systems might hope for. I have tried to suggest things which I think are achievable right now. If one were to look further into the future one might well expect more revolutionary changes. I would conclude that there is a future for statistical packages, that their usefulness and ease of use will increase, and I think statisticians should view these developments with informed interest, rather than alarm.

References

1. System/360 Scientific Subroutine Package (360A-CM-03X) Version III Programmers Manual, International Business Machines Corp. 1968.

2. Computer Subroutine Libraries in Mathematics and Statistics (Abilities), International Mathematical & Statistical Libraries, Inc., April 1, 1963.

3. W. J. Dixon, Ed., BMD Biomedical Computer Programs, University of California Press, Berkeley, 1973.

4. Jolayne Service, A User's Guide to the Statistical Analysis System, Student Supply Stores, North Carolina State University, Raleigh, North Carolina, August 1972.

5. Norman Nie, Dale H. Bent, C. Hadlai Hall, SPSS, Statistical Package for the Social Sciences, McGraw-Hill, 1970.

6. K. W. Smillie, STAT-PACK 2: An APL Statistical Package, University of Alberta, Dept. of Computer Science Publication No. 17.

7. CALL/360: Statistical Package, Version II (G320-1009-1) 2d ed. International Business Machines Corp. Feb. 1970.

8. BMDP Series information is available from Health Sciences Computing Facility, University of California, Los Angeles, California 90024.

9. OMNITAB II, a Computer Program for Statistical and Numerical Analysis, National Bureau of Standards.

10. W. R. Schucany, Paul D. Minton, B. Stanley Shannon, Jr., A Survey of Statistical Packages, Computing Surveys, Vol. 4, No. 2, June 1972.

11. Patricia M. Britt, Wilfrid J. Dixon, Robert I. Jennrich, Time Sharing and Interactive Statistics, Statistical Computation, R. C. Milton and J. A. Nelder eds., Academic Press, 1969.

12. Ivor Francis, Comparison of Several Analysis Programs, Journal of the American Statistical Assoc., Vol. 68, No. 344, December 1973.

13. K. E. Iverson, A Programming Language, John Wiley & Sons, Inc., 1962.