Middle Articles

Computer Programmes for the Medical Man: a Solution

MAX HAMILTON,* M.D., M.R.C.P., D.P.M.

Brit. med. J., 1965, 2, 1048-1050

The application of electronic computers to medical problems offers roseate visions, which the experts are busily working towards. Even ordinary medical research workers and clinicians are beginning to understand some of the possibilities, and a few of them are even beginning to think that these machines might be of help to them in their everyday work. Unfortunately, when they try to find out something about the practical use of electronic computers they come up against difficulties. This is true even for those who are acquainted with programmers, or can get in touch with computing installations. It is the purpose of this paper to give an account of these difficulties and to provide a solution.

How a Computer Works

A digital computer is a device for manipulating symbols. These may represent the digits of arithmetic, or they may be symbols of logic. It is a "general purpose" device, and will deal with specific tasks only when it is first "adjusted" to do them. This adjustment is known as programming. The programming of these machines was at first extremely complicated, and only the expert could do it. It became obvious that the process had to be simplified, and this was done by dividing the procedures into two stages. The first is to write the programme in a simple form, in a "language" easy to understand and use; and then to write another programme which will "translate" the first into a form that will actually operate the machine. The latter programme is called a "compiler," and once it has been designed and tested it can be fed into the machine and stay there permanently. The prospective user of the machine has merely to learn the simple language and write his programme with its aid. Since the simplified language requires only a few weeks to learn, this marked a great step forward in making the machines accessible to computer "laymen."¹

In actual practice the advance was not so great. It is easy to write a programme, but it is difficult to write one *correctly*. A computer cannot work if a programme has errors, and the compiler includes procedures for checking that a programme is correctly punched. When an error is found, the machine stops and prints appropriate symbols, which enables the programmer to find the errors. These may be (to the human

mind) extraordinarily trivial: a small letter instead of a capital letter, the omission of a comma or semi-colon, and so on. Once the error has been found, it has to be corrected by re-punching, and the programme tried again on the machine; another error may then be found. Though computers can add two figures together in millionths of a second, it may take weeks, even months, to get a programme into working order. Most of the time is spent queueing for access to punching machines, the rest on waiting for time on the machine. Very little time is actually spent in looking for errors and correcting them: a mere total of hours.

Standard Programmes

Most of the work done by digital computers is routine work. The same programmes are being used every day by hundreds of machines all over the world, and it is waste of time repeatedly to rewrite and test them. Most of the medical computing done consists of two types: first, the sorting out and collation of data, with perhaps the calculation of proportions and percentages; and second, the statistical analysis of experiments. To cope with routine work, most computing laboratories and manufacturers of computers have prepared a "library" of standard programmes. This stage was reached a few years ago.

The physician or research worker who tries to get some computing done for him is soon disillusioned. The standard programmes exist, but as far as he is concerned they are a myth. In the first place, they may be written in a language which cannot be used on the local computer. If the language is correct, the "conventions" may be wrong—that is, even when a machine can use a standard language the appropriate form in which data are punched varies from one machine to another. So do the instructions for reading the data into the machine and for printing it out.

For example, to read 15 figures into a machine using the Auto-code language, the instruction is written and punched on to the tape as

V0 = TAPE 15

which means that the machine will store the first number into the "container" called V0, the next one into the container V1, and so on, the last going into V14. The machine will then proceed to the next instruction. In the Algol language the same instruction is written

for j: = 1 step 1 until 15 do
Data [j]: = read (20);

which means that the machine will read the figures punched in the tape into the set of 15 containers labelled Data 1-15. The letter j is used as a counter and when it reaches 15 the last figure is read and the machine goes on to the next instruction. In some machines a figure is printed out using the instruction "PRINT," in others it is written as "WRITE," and so on.

It is therefore necessary to find a programmer who has time to spare and explain what is required. He will then find a standard programme and rewrite it to suit the machine. The programme will have to be punched and checked for errors

* Nuffield Professor of Psychiatry, University of Leeds.

¹ A programme is written on sheets of paper, as are the results of experiments. The written symbols are then punched on tape or cards. The tape is then put into a machine, which prints out in ordinary characters the material which has been punched. In this way the accuracy of the punching can be verified and errors corrected by re-punching. The programme tape and data tape are then fed into the computer. The information fed into the computer is known as "input," a term which is also used to refer to the actual tapes or cards. The results produced by the computer are the "output," and can appear either in the form of punched tape or cards, which then have to be printed out, or directly as printed material. In the former case, the output is in a form which could be used as input; in the latter, it would have to be hand-punched and checked first.

(more delay) and tested on data the results of which are known (still more delay). After this the data will have to be punched, checked for errors, and finally dealt with. It is possible that the standard programme has already been punched in the appropriate conventions, but it is almost certain that it will have to be rewritten to cope with the particular data. Be that as it may, the user who cannot go personally to a computer installation and discuss his problems with a programmer finds himself in an almost impossible situation.

Problems in Medical Investigation

For special computing jobs it will always be necessary to call upon the services of a skilled programmer, but he should not have to waste any of his time on standard or routine tasks. For such it should be possible to have the data punched, hand them to a machine operator, and receive the results in due course. Before describing a system of programmes that achieves just this, it will be opportune to consider some traps for the unwary.

In any investigation involving human subjects the most difficult part is getting hold of them. It is expensive and takes a long time. The investigator is therefore tempted to obtain as much information from them as possible. If he knows that he will be able to use machine methods for analysing his data, the temptation becomes almost irresistible. When a large number of items (variables) has to be analysed, the usual statistical tests of significance begin to fail. With 200 variables, one would expect to find about 10 results "significant at the 5% level," and if the variables are correlated, the number could be much greater. To overcome this difficulty one has to examine a further series of cases. Admittedly this ought to be done anyway. In the second series, one would reject the variables that have been shown to be irrelevant, and perhaps investigate new ones; but it would be better to resist the temptation to collect every kind of information and spend the time first in thinking more carefully about what would be relevant, and to devise hypotheses to be tested. Without machine methods the sheer labour of analysis of the data acted as a deterrent to the unthinking collection of figures and encouraged preliminary thought. It is not now necessary for the investigator to confine himself to a mere half-dozen variables, but he should use judgment and common sense when deciding the number he will tackle.

When results are analysed by hand they appear slowly, one by one; the investigator has time to think about them and consider not only their meaning but what he should do next. With access to a computer he may have to wait many months before he sees his results, but when they do arrive they come in an enormous pile and he needs months to digest and assimilate them. Ideally the machine should produce the results in little dribbles over a reasonable period of time. This suggestion always shocks the computer man. It wastes the time of the computer and this is his chief concern; that the alternative wastes the time of the investigator is not his problem.

System of Programmes

What is required is a standardization that begins with the form of the data. Standard programmes for routine procedures should then be able to handle the data without any adjustment whatsoever. In order that intermediate results should be capable of being further processed, it is necessary that the output of any one programme should be compatible with others, again without further adjustment. The desired goal is a set of programmes that are interlocking, self-adjusting, and, because they can be indefinitely expanded in number and type, are completely flexible. Such a set now exists in the department of psychiatry at Leeds University. Provided that the data are punched in the standard form, all that is necessary

is to hand to the computing department the data tape and the programme tape. They can then be handled by the machine operators without requiring the assistance of a programmer. The details are given in the appendix below, but it is worth describing a few of the procedures that can be undertaken.

One programme will sort out the cases into two or more groups and will then do a statistical test on the differences between the groups for all the variables recorded. For example, if the sex of the patients is recorded, it is possible for the programme to sort the patients out into the two sexes and test for the differences between them. If the patients have been given several treatments, the cases will be sorted out according to the treatments and statistical tests done for the differences between the groups given the different treatments. This enables the investigator to see whether the groups were reasonably alike to start with. If some of the patients have been followed up to ascertain their subsequent progress, but not all of them, it is then possible to punch into another tape the results of their treatment or their eventual outcome, and this can be fed, together with the data tape, into the machine. It will then pick out the cases which have been followed up and will examine the statistical relationship between the variables recorded on the tape and the outcome of treatment. In this way it is possible to search for factors related to outcome.

Patients can be sorted out into groups according to their scores on a continuous variable such as age, blood-pressure, or length of illness. For example, the ages can be classified into groups, and the patients will be sorted out into these groups. Again, the programme will examine and test the differences on all the other variables. There are programmes for calculating correlation matrices and for further analysing these. One programme will sort the cases on two variables and determine the bivariate frequency distribution. If a criterion is available, it will determine the mean of the criterion for each cell of the distributions. For example, it is possible to determine the mean outcome of treatment for the different ages and length of history combined. Many of the programmes are designed to be cumulative, so that the data from a new series of cases can be added to the results of previous series. The programme for checking the accuracy of punchings is one of the most useful, since it thereby reduces to a minimum one of the duller, most time-consuming tasks connected with computing.

The system of programmes described is not perfect. At the moment it is available for use only on the English-Electric KDF9 machine. A fair number of British universities have such machines, so the system can be used by them. It is also confined to data punched on 8-hole tape. Steps are being taken to translate the system of programmes for use on other machines using data punched on 5-hole tape or cards. The number of programmes now available is limited, but the advantage of the system is that anybody who writes a programme that is compatible with the system automatically makes it available to all other users of the system. It is the underlying principle of the system that is fundamental: a standard layout for the data; programmes which produce an output in a standard form so that the output tapes can be used for other programmes (this avoids the difficulties and errors of further punching); programmes that produce results in stages, and that give intermediate results so that they can be used cumulatively. For convenience in referring to these programmes they have been named "The P.L.U.S. System," an acronym derived from Psychiatry Leeds University Standard System of Computer Programmes.

Summary

The physician or research worker who wishes to make use of computers for the analysis of his data finds practical difficulties unless he is in close contact with an installation which provides a service. The situation is slowly improving

but it is still unsatisfactory. A system of programmes has been described which will enable the prospective user to get routine statistical work done without calling on the services of a skilled programmer. The system has been designed so that it is as flexible as possible; it will handle a large number of variables and any number of cases. Results are produced in stages for ease in comprehension. Its great asset is that it is available here and now.

The major part of the work was done while I was working on the External Scientific Staff of the Medical Research Council, to which credit is due. I have to thank Mr. R. J. McGuire and Miss M. J. Goodman, of the Department of Psychiatry, University of Leeds, for help in translating the programmes into Algol.

Appendix

The primary data tape starts with a number which acts as identifier. The number is copied on to the output, which serves to identify the data from which the results are obtained. This is important when more than one series of data is being analysed. After the identifier comes the number of variables, and then the number of cases. The data for each case is then punched, preceded by the case number, and followed by the sum of all the figures for that case (this serves as a check). Finally, a marker indicates the end of that case.

Example of Standard Data Tape

```

Identification
number ..... 1290365 ;
Number of
variables ..... 6 ;
Number of
cases ..... 7 ;
First case ..... 21 ; 32 ; 0 ; 2 ; 97 ; 1.53 ; 157.73 ; 190.26 ; 1.0-6 ;
Second case ..... 22 ; 45 ; 0 ; 0 ; 103 ; 1.67 ; 232.09 ; 381.76 ; 1.0-6 ;
Third case ..... 25 ; 23 ; 1 ; 1 ; 107 ; 2.01 ; 183.47 ; 317.48 ; 1.0-6 ;
Fourth case ..... 27 ; 55 ; 0 ; 0 ; 95 ; 1.89 ; 113 ; 264.89 ; 1.0-6 ;
Fifth case ..... 33 ; 39 ; 1 ; 1 ; 110 ; 1.07 ; 154.2 ; 306.27 ; 1.0-6 ;
Sixth case ..... 35 ; 47 ; 1 ; 0 ; 101 ; 1.4 ; 195.673 ; 346.073 ; 1.0-6 ;
Seventh case ..... 40 ; 24 ; 1 ; 2 ; 91 ; 3 ; 211.6 ; 332.6 ; 1.0-6 ;

```

→

Case No.	Variables	Sum	Marker
----------	-----------	-----	--------

In the example above the third case on the tape is case number 25. The six figures for this case are 23, 1, 1, 107, 2.01, and 183.47. Then follows the sum of these six figures, 317.48. Finally comes the number 1.0-6 (a millionth), which is the marker indicating the end of that case.

Some Programmes Now Available

1. *Testing data.* Checks the punched data for errors in number of figures in each case, errors in the sum check—that is, incorrect punching of figures so that the total is incorrect—and prints the case numbers in which each type of error has been found. It then calculates the sums, crude sums of squares, variances, standard deviations, and means for all the correct cases.

Input : Standard data tape.

2. *Sums and Products.* Calculates sums and matrix of crude cross products for either integral or real values of the variables. The output is suitable as input for other programmes (see S.P. derivatives). Two versions allow input data to be processed or new data to be added.

Input : Standard data tape.

3. *S.P. Derivatives.* From crude sums and cross products forms (1) total corrected sums of squares and cross products (deviance-codeviance), (2) dispersion (variance-covariance), (3) correlation

matrices, (4) variances, (5) standard deviations, (6) means, for all variables and their sums.

Input : (1) A steering tape on which is punched six zeros, replaced by a 1 for whichever output is required.

(2) Output tape from any of the sums and products programmes.

4. *Latent Roots and Vectors.* Performs a principal component analysis on a symmetrical correlation matrix (unity in diagonal). Output of factors is in decreasing order of magnitude and gives saturations for each variable and the latent root (variance).

Input : Correlation matrix (part of output from S.P. derivatives).

5. *Derived Scores.* Produces a series of weighted sums of the variables on a primary data tape. Used also to change the means and standard deviations. The output from this programme is in the form of a primary data tape, so it can also be used for producing a tape containing a set of selected variables.

Input : (1) A directive tape in which is punched the number of derived scores required, the number of variables, and the appropriate number of constants and weights.

(2) Primary data tape.

6. *Analysis of Variance (1).* Using one of the variables (including the case number and the sum) a list of cases is divided up according to whether the score on that variable exceeds or not a number of boundary scores. For all other variables, the frequency, sums, and means of each sub-group are printed, and a test significance of the difference between the means is carried out.

Input : (1) Standard data tape.

(2) Number of separate sortings (n), then for each sorting the criterion variable number (0 to p + 1), number of boundary scores (s), and boundary scores in order of increasing magnitude.

7. *Analysis of Variance (2).* From the standard data tape certain cases are selected by case number, and for each variable an analysis of variance tests whether the difference between the means of the selected cases and the remainder is statistically significant.

Input : (1) Number of cases to be selected and their case numbers.

(2) Standard data tape.

8. *Regression to a Criterion : Quantitative data.* Selected cases are chosen from a standard data tape and a criterion score supplied. The linear and curvilinear regression equations for each of the variables on the criterion score is then calculated, together with the statistical significance of each regression.

Input : (1) Number of selected cases and for each case its case number and criterion score.

(2) Standard data tape.

9. *Regression to a Criterion with Distributions.* Similar to No. 8 with the addition of the calculation for each variable of the frequency, sum, and mean of the criterion score for each of the values of the variable (which must be integral).

Input : (1) The highest value of any variable, the number of cases to be selected, and for each case its case number and criterion score.

(2) Standard data tape.

10. *Mean Criterion Sort.* Selects cases from a standard data tape by case number, and calculates bivariate frequencies together with total and mean criterion score for each non-zero frequency.

Input : (1) Number of variables, number of cases to be selected, case numbers and corresponding criterion scores, maximum score of any variable, number of sorts and variables to be sorted on.

(2) Standard data tape.

11. *Discriminant Function.* Selects cases from a standard data tape to produce two groups and then calculates a discriminant function for distinguishing between them.

Input : (1) Number of cases to be selected and the case numbers.

(2) Standard data tape.