Interactive Computer Graphics in Multivariate Statistical Research

by

Anna Frances Grundy

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Computer Centre
University of Keele

June 1977

CONTAINS PULLOUTS

ACKNOWLEDGEMENTS

The work for this thesis has been carried out at the Computer Centre, the University of Keele, under the supervision of Dr. H. H. Greenwood. In submitting this thesis the author would like to thank the following people.

Dr. H. H. Greenwood for his advice and constant encouragement in the planning and development of this work, and for suggesting particular lines of analysis.

Dr. A. Hill for data from his investigation of women at work.

Mr. R. Barr for the selection of the census data and for presenting the original cluster analysis problem.

Mrs. C. Goulding for typing this thesis, and the staff of the computer centre for their help through many hours of computing time.

CONTENTS

ABSTRACT

The value of interactive computing and of interactive computer graphics is discussed with particular reference to the study of statistical data. A system designed for interactive graphics for multivariate statistics is described, including a command language to be used as part of this system.

The system is in two parts: administrative programs and statistical programs. Types of statistical problems suitable for interactive graphical analysis are given, with the multivariate techniques used in the solution of these problems: principal components, factor analysis, cluster analysis and discriminant analysis. Each form of analysis is described with an outline of the program which implements the technique, including interactive aspects and forms of graphical output provided.

The latter part of this thesis covers the use of these programs. Principal components analysis is used for two sets of data; a selection of results is shown and discussed. Results obtained from factor analysis of data collected for a psychological study are similarly given. For a fourth data set, initially a cluster analysis solution is found. The graphical presentation of results led to the identification of a set of variables which dominates this initial solution. The interactive procedure for identifying this and further sets of variables is described. Finally, results obtained from the initial cluster analysis solution are input to the program for discriminant analysis.

INTRODUCTION

In many fields the production and use of pictures is an integral part of the research or design process, in others the graphical presentation of data is essential. Even in instances where diagrammatic representation of data and results is not essential it can help to improve understanding of models and techniques.

The introduction of interactive computer graphics firstly facilitates the production of pictures, and secondly, because of the means by which pictures are produced, it widens the range of applications where graphics may be of value. There are many types of display, of varying sophistications, available for interactive graphics. The most comprehensive of these are refreshed display tubes. A picture once displayed rapidly decays and has to be refreshed several times a second to maintain a continuous display; this allows for partial deletion of pictures. The resolution of lines generated on refreshed displays is good, and the more sophisticated of these devices have hardware rotation and windowing and a wide range of intensities for display. The rate at which a picture is refreshed depends on the complexity of the picture and the quantity of data to be drawn; it is difficult to maintain a very complex picture without flicker. Storage tubes are cheaper and also have good resolution, but a picture does not require refreshing; once displayed it can be maintained on the screen. However items cannot be selectively deleted, the entire screen has to be erased. The simplest type of graphical display is a point plot device where vectors are not drawn, lines are made up of a series of points and there is only poor resolution. The range of input devices available for use with these is limited. Their main advantage is their relative cheapness.

The most common input devices available for use with these displays are alphanumeric keyboards for conveying textual information to a program, and sets of program switches or keys for indicating operator choices. The other major form of input devices required are graphics input devices, to identify items on the screen and to specify co-ordinates. For refresh tubes the most usual device for graphical input is a light sensitive pen which can be used to "see" items or identify parts of a picture or it can be used in conjunction with a tracking cross, which moves with the pen, to specify co-ordinates. These are unsuitable for use with a storage tube. A cursor is another device for graphical input usually employed for storage tubes. This can be controlled by various devices such as a joystick, thumbwheels, mouse or tablet. The current X- Y co-ordinates of the cursor are maintained and can be accessed from an applications program. Point plot devices are used in very much the same manner as conventional V.D.U.s and simply display diagrams in addition to alphanumeric text. The only form of input device usually available with these is an alphanumeric keyboard.

Exactly how are these devices of use and in which areas of research and production? At the simplest level they can be used for program testing, for instance in problems where a hard copy of a single picture is eventually required, or where a choice has to be made from a sequence of many diagrams. The intrinsic value of these devices stems from being able to interactively choose, modify and construct diagrams piece by piece.

Interactive construction and alteration of pictures is useful in a wide range of design problems. Components can be added to a diagram one by one on the instruction of the designer, and the consequences of the addition of each component examined. If a particular design configuration

is unsatisfactory steps can be taken to reinstate an earlier configuration. Designers no longer have to continually consider costs and engineering constraints since these can be automatically monitored; designers can also be relieved of routine activities such as producing drawings and machining instructions. This step-by-step design procedure occurs in the use of Multipatch and Multiobject (Armit, 1971a, b) for the design of 3-d objects, for example in the design of glassware (Hart, 1972). A similar procedure is used in DIECAST, an interactive system for the design of 3-d objects (den Hartog and Veenman, 1972). A surface is subdivided into smaller sections or "patches" which can be individually moulded and joined together to produce the required shape. Interactive curve smoothing employing related techniques is used for car body design (Ciaffi and Marello, 1972). In architectual applications the architect can maintain control of overall design and aesthetics while engineering problems associated with the design and design costs are assessed and reported automatically to the user at the terminal (Walter, 1969). In this type of application building components can be moved around the screen to assume different design configurations (Maver, 1972).

Interactive graphics has applications in the field of electronic engineering, for instance for integrated circuit design (Annoni et al, 1972) (McDouall, 1969). In civil engineering interactive graphics is used for highway design. Different aspects of a proposed design can be inspected; these may be used to simulate driving along a proposed highway. Since these problems involve large quantities of interdependent data, any modification to the data can have widespread repercussions for a given model. With a suitably designed interactive system modifications can be monitored and the necessary adjustments made to the

data base. Feeser (Feeser, 1972) describes a system for generating and displaying perspective views of a proposed highway construction.

In cartography interactive graphics is used for automated map drawing. For geological applications, stresses and climatic changes can be simulated, for instance to predict structural changes in the earth's crust, or to determine the original shape of fossils.

Interactive graphics has been introduced for the solution of a problem of region partitioning (Cheung, 1974). Boundaries are specified and modified interactively, as are service centre locations and cost functions. Linear programming techniques are used to find the minimum costs given the constraints. Results are presented graphically and the constraints may be further modified.

An extensive survey of interactive graphics systems for mathematics is presented by L. B. Smith (Smith, 1970a). The systems surveyed include APL/360 based on APL (Iverson, 1962), also SCRATCHPAD (Blair, Griesmer and Jenks, 1970) and MATHLAB (Engleman, 1965) which provide interactive symbolic computational facilities. More recently a system known as ISLAND has been developed (Chau, Davies and Zacharov, 1974). This system is designed primarily for use in a physics research environment; it provides facilities for interactive vector and matrix manipulation with graphical representation of results.

There are currently only a limited number of interactive graphical systems for data analysis and multivariate statistics. These are discussed in more detail in Chapter 1.4; they include STATPAC (Goodenough, 1965), PEG (Smith, 1970b), OLPARS (Sammon, 1968), GOLDA (Harris, 1972) and an interactive version of some BMD programs (Britt, Dixon and Jennrich, 1969).

Interactive data analysis does not involve the construction of a model, or a process of simulation, but systematic investigation into the structure of a data set. J. W. Tukey, in a discussion on the nature of data analysis (Tukey, 1962), proposes that it should have the following three characteristics (page 6):

"(b1) Data analysis must seek for scope and usefulness rather than security.

(b2) Data analysis must be willing to err moderately often in order that inadequate evidence shall more often suggest the right answer.

(b3) Data analysis must use mathematical argument and mathematical results as bases for judgement rather than as bases for proof or stamps of validity."

There are no precise answers to the problems of data analysis. Although statistical techniques provide precise tools, the choice of individuals forming a test population is subject to sampling errors and the choice of variables describing this population is a matter of subjective judgement. Problems are often loosely defined and, therefore, data analysis can be expected to do no more than to suggest a solution.

With an interactive system for data analysis which has the precision of individual multivariate techniques, the user can obtain results which point to the correct answers. An interactive system allows sufficient flexibility for systematic and rigorous investigation into the structure of a data set.

Interactive graphics is of value in this process in that the results can be presented more concisely, and suggestions and pointers to the direction in which the analysis should proceed are made by means of

easily assimilated pictures and diagrams. These can be systematically chosen, and rapidly produced. There are algorithms for presenting n-dimensional data in a 2-dimensional picture to convey different aspects of the structure of a data set in a form which numbers alone rarely achieve. However, there is no single picture which can convey all the information required from a data set; it needs to be examined in different ways by means of sequences of pictures. This process involves being able to choose subsets of variables and observations for analysis, to initiate different multivariate techniques; it also involves being able to examine the results of these in graphical form where possible, and in numerical form if required. The graphical output may involve the production of many pictures, for instance histograms for each of the variables, or in some instances the same results can usefully be presented in two or three different ways.

This thesis describes and demonstrates an interactive graphical system for multivariate statistics. Graphical output and the control of analysis at the graphics terminal is a prime attribute of the system. This emphasis on graphics has to some extent determined which kinds of multivariate analysis have been included in the first instance. Those chosen provide interesting forms of graphical representation and of user interaction.

That both graphical and numerical output are of importance in statistical data analysis is demonstrated by Feder (Feder, 1974) in using various graphical displays to demonstrate the results of standard statistical techniques. It is shown how these displays influenced the investigation of a particular set of experimental data and initiated the use of additional numerical techniques. In the introduction Feder writes

that "... many of the usual (and not so usual) statistical analyses can and should be preceded and supplemented by graphical analysis." This system is designed to facilitate such a procedure, and, through more extensive use of this and similar systems, to widen the range of possible representation of data and statistical results which may generate new ideas for investigation.

The thesis is introduced with a description of the position of an interactive graphical system for multivariate statistics in relation to other interactive and batch statistical systems. The purpose of individual programs is described, how they are used and the various forms of graphical output provided. The use of these programs is demonstrated with four separate data sets. Some of these are conventional exercises with the results demonstrated with graphical output. The last study is more extensive and demonstrates the way in which the system can be used to best advantage. Graphical output is used to isolate the set of variables which contribute most to an initial cluster analysis solution. The analysis is repeated several times with the remaining variables, on each occasion isolating a new subset. This procedure does not find the 'best' set of variables or the set which can be used to most nearly reproduce a given solution. But, by examining sequences of pictures, the way in which different sets of variables contribute to the structure of the data set is better understood.

1. INTERACTIVE COMPUTING AND MULTIVARIATE STATISTICS

This chapter describes the relevance of interactive computing and interactive computer graphics to multivariate statistical research. In any field, large scale research involving the use of batch programs is slow and unwieldy. The instantaneous responses and the control provided by interactive programs make this heuristic research procedure more powerful and efficient. Interactive graphics provides additional features which become of fundamental use in problem solving. The use of batch programs for statistics implies many of the drawbacks of batch computing. Interactive statistics programs make use of some of the features developed within batch programs, such as a common data base and facilities for filing results for later reference, while at the same time providing the advantages of interactive computing. If computer graphics are introduced, the pictures and diagrams become of paramount importance in multivariate statistical research.

1.1. Batch operations and the significance of interactive computing

The traditional method of running a user's large scale computer program is to run it in batch mode out of direct control of the user. In the absence of widely available time-sharing facilities to run large scale programs interactively, compilers and programs have largely been written to operate in a batch environment. However, batch mode operation has limitations which often make it inappropriate for problem solving.

There can be little control over processing in a batch environment and no intervention. A job cannot be abandoned when an examination of preliminary results can show that it is unnecessary to complete the run, nor can a run be extended if it has not gone far enough. There can be no

instantaneous response to results: jobs are submitted to cover many steps of an analysis, the steps taking a predetermined path, and there is no opportunity to alter the direction of the analysis after the calculation of initial results. This mode of operating requires many runs of a program, with an appreciable time elapsing between each run. These runs frequently produce large volumes of output for analysis, an analysis often done by the time consuming process of drawing graphs from numerical data obtained from different parts of the output. This procedure is employed in, for example, problems requiring the variation of parameters, or the search for an optimum. Slow turnround encourages batch users to follow several lines of investigation when concentrating on one theme might be more advantageous. The development of programs in batch mode is inefficient; not all the errors, typographical, syntactical and logical are necessarily discovered in an initial run, it can take several runs to diagnose and correct them all.

Interactive systems can be designed to overcome some of these problems and to provide, in principle, for more efficient use of programs and program development. Large scale interactive programs provide powerful tools for heuristic research. The nature of this process involves the presentation of results which suggest new directions for investigation. Facilities exist for the ready implementation of these ideas, the results of these in turn suggesting further new ideas. The user has control over the direction of an analysis at all points, and may interrupt unsatisfactory procedures and continue and expand those which appear more fruitful. This control is relevant for determining the broad outline of an analysis, for example, the choice of which statistical procedure to use. It is used for specifying the details, for instance, for specifying and altering the

stepsize in an iterative optimisation procedure, or in interactive graphics, for the construction of a diagram piece by piece. The flexible nature of this mode of operation allows output to be arranged to suit each problem; for instance a complete multiway table of frequencies does not always have to be printed, only those portions which are relevant to the solution of the problem in hand.

Interactive systems have their limitations. Hardware usage is less efficient, users feel under pressure to initiate jobs while sitting at a console, although with increased availability this attitude may change. Sequences of analysis have to be carefully planned, since too much ease of access can result in the investigator spending time pursuing lines of analysis tangential to the central problem.

There are implications for program design and program writing. The possible routes which may be taken through an interactive program are far more numerous than those through a batch program. For instance, allowance must be made for the user typing any sequence of characters at every juncture, or pulling any of the available switches. The program must not terminate without due warning and explanation, and where there is error recovery it must be clear and straightforward. Results should be saved at regular intervals, so that a restart after a system crash or at a later session is easy to implement.

While it is possible to write interactive programs in languages primarily designed for batch operation, interactive languages such as Basic and Pop2 have advantages over batch oriented languages. Firstly, they are more efficient for program development. At execution time a suitably designed program can be easily extended in a direction relevant to the current situation. The nature of these languages is such that they

are well suited to interactive research, and to being able to continuously monitor and determine the direction of an analysis as a run proceeds.

1.2. The need for graphical output and for interactive graphical computer systems

Graphical output is essential for some problems, and for others would enhance the quality of output and make results easier to assimilate. Pictures give concise representation of results, and help to make a quick judgement about the quality of results, and whether or not to obtain further detail with additional pictures. Graphical presentation highlights trends in results which are not so easily gleaned from lists of numbers. For example, maxima and minima are identified very quickly from a pictorial representation of a function, especially small local fluctuations which often otherwise go unnoticed. The existence of any outlying members of a data set is more easily identified in one or a series of pictures.

In addition to the advantages of interactive computing, interactive graphical computer systems offer rapid graphical presentation of results. The way in which these are of use in the interactive process depends on the field of study. The examination of graphical output may suggest further experimentation or more detailed study of certain aspects or sections of a model or data set along lines indicated only by the graphical output. Applications for interactive computer graphics in design problems where models are interactively built and modified have already been mentioned. The results and repercussions of each amendment can be examined and acted upon. A complete model does not have to be built in one single operation without any user intervention; a mode of operation which may necessitate further runs, and involve the time consuming production of several versions of digital plotter output.

In simulation problems and others where graphical presentation of output is available, an examination of the diagrams can help to determine whether or not to proceed with a particular line of analysis.

Graphical output presented within an interactive system may influence the choice of the next step in much the same manner as numerical results. For instance, it may suggest entry to a different form of analysis, or variation in the parameters of a model.

In some instances it is possible to examine the same results in different ways, either by use of different representations or by viewing a model from different angles. One major advantage of interactive graphics is that a chosen sequence of pictures can be displayed at a speed dictated by the current situation. The display of sequences of pictures can be of use in many instances. To quote a few examples; a model can be viewed from a continuous sequence of angles,not only selected ones, and a data set can be shown in terms of each of its variables in succession. In problems involving optimisation not only can the optimum conditions be displayed, but also many other sets of conditions which may be instructive.

Interactive graphics widens the scope of interactive computing by making combined use of the concise representation of results in pictorial form and the instantaneous presentation of these pictures. In many applications the use of these pictures becomes an integral part of the research procedure.

1.3. The nature of batch processing systems in relation to multivariate statistics

The traditional approach to computing in multivariate statistics has been based on batch mode operation. Programs were written for batch processing system to perform standard statistical operations, such as one way and multiway tabulations, multiple correlation, multiple regression, analysis of variance and covariance, principal components, factor analysis, discriminant analysis and cluster analysis. In the first instance these were stand-alone programs designed primarily for a specific form of analysis applied to a given data set. For more complicated analyses involving several stages these programs are more difficult to operate. There are several reasons for these difficulties, firstly, since input formats to separate programs performing related analyses are often different, data and parameters have to be rearranged for each program. Furthermore data has to be submitted as a simple rectangular matrix, only limited provisions are made for transmitting data from one program to another and there are often only limited facilities for transformations and data selection.

Sets of related programs or packages for statistical methods were produced to overcome these and similar operational problems and to provide flexibility for data analysis. Not all the available packages overcome all problems, and some emphasize features related to a specific type of problem. Complex and hierarchical data structures are available in some packages and also different forms of data manipulation and data editing. Within one package input of data and parameters is uniform to all procedures; uniformity of input and output formats makes it possible to use output from one procedure as input to another. Transformations are

frequently required prior to analysis; these may be algebraic transformations or transformations resulting from some experimental problem, such as recoding alphanumeric responses. These transformations may have to be conditional, and in a package facilities for interpreting arithmetic and logical expressions need only be supplied once. The syntax for these is uniform throughout, and if identical transformations are required for several steps of an analysis, these have only to be specified once and will remain effective as long as they are required. Once some standardisation has been achieved it is a simple matter to extend a package with the addition of new procedures.

A survey of statistical packages is presented by Schucany, Minton and Stanley (Schucany, Minton and Stanley, 1972). A selection of those most widely available are briefly discussed.

BMD (Dixon, 1973), a package of interrelated programs intended for Biomedical statistical research, is widely available in batch mode. Data must be submitted as a rectangular matrix with no hierarchical structure. Parameters are supplied in numeric form without identifying text. Since each program is a self contained unit for a single procedure, a procedure is called by loading the relevant program. Facilities for transformations have to be provided with each program, which can be wasteful, since identical transformations may have to be respecified. The BMDP series of programs is a more recent version of BMD (Dixon, 1975), (Frane, 1976). The major difference between the two series is the introduction of parameter language control. Parameters no longer have to be provided in fixed columns; they can be clearly and easily specified. Further enhancements include additional printer graphical output, more easily specified transformations, facilities to save files, to select subsets of data and

improved numerical techniques. Some preliminary work on interactive BMD programs is described by Britt, Dixon and Jennrich (Britt, Dixon and Jennrich, 1969). Both the interactive and graphical aspects are demonstrated by means of a non-linear least squares program.

SPSS (Statistical Package for the Social Sciences) (Nie et al, 1975) is written for batch mode operation. In most implementations SPSS appears to the user as a single program and control words are used to request particular statistical procedures, allowing several different analyses to be executed in one run of the program. The data must be supplied in one form only, as a conventional rectangular matrix, and files can be saved for later analysis. A syntax for transformations and data selection is applied uniformly for all subsystems of the package, and if required, specifications for these remain effective throughout the use of different subsystems.

A more flexible method of accessing statistical procedures lies in the use of subroutines such as those provided in the IBM Scientific Subroutine Package (SSP) (SSP, 1970). A main program written in Fortran calls these subroutines and provides annotated output where required. In practice, it is usually possible to provide sufficient flexibility with program packages, these have the additional advantage that their use does not require a knowledge of a programming language.

Systems designed to be flexible in that they cater for a wide range of problems and eventualities within the field of multivariate statistical research are those which allow the execution of procedures and editing and manipulation of data sets by means of instructions or commands. ASCOP (Cooper, 1967, 1969) was originally available in batch mode, but was written with a view to interactive work. A program is submitted

consisting of a series of instructions which include arithmetic assignment statements specifying arithmetic operations to be performed on the data, instructions to initiate statistical procedures and editing instructions to merge and append files. Instructions can be labelled and referenced by conditional and unconditional branching statements in order to omit the execution of instructions as necessary. User defined subroutines may be supplied. The syntax of commands is simple and the meaning of instructions is easily understood.

GENSTAT (Nelder, 1975 a, b) is a batch statistical system with its own command language and extensive facilities for multivariate analysis. Complex data structures can be defined and arithmetical operations include matrix and table operations which take account of this complex structure. Matrices specifically required for statistical procedures can be easily and explicitly obtained, and used for subsequent analysis.

These batch systems have done much to cater for non-standard sequences of analysis without representation of the data. It is arguable that, to realise their full potential these and similar systems should be adapted for interactive working, although this would not necessarily be the ideal mode of operation under all circumstances. A carefully planned batch operation following a set of clearly prescribed steps can, in the right circumstances, be both more fruitful and more economic.

1.4. Interactive statistical systems

With the development and increased use of multistage statistical analysis it becomes necessary to be able to modify the analysis as it proceeds. Interactive systems make this possible by presenting intermediate results and allowing the user to determine the path of the analysis.

Interactive statistical systems are of varying complexity. The simplest are small interactive programs which perform a single straightforward statistical procedure. Input data and parameters are supplied at the console and results presented on the console. Apart from the initial input there is no user interaction with the program.

There are small scale interactive programs for single statistical procedures which allow the user to interact with the program to determine details for the procedure; for example MULREG, an interactive multiple regression program available on the ICL 4130 (KOS User Manual, 1973). Transformations may be introduced interactively at the console, variables entered into the regression equation and residuals and associated statistics examined. If necessary, further equations may be defined. This system is designed for a specific purpose and apart from the absence of built in data manipulation and file-editing facilities, it operates adequately in solving regression problems.

Special purpose but more comprehensive interactive systems are available. IDA (Interactive Data Analysis) (Roberts, 1974) is also primarily for regression, but with extensive file-editing facilities. Graphical output is available, for example, plots of the residuals against the dependent variable. In IDA these are character plots which give only poor resolution on a VDU with its limited number of lines. GLIM

(Generalised Linear Interactive Modelling) (Nelder, 1975c) is an interactive system with an interpretive language for fitting generalised linear models. OLPARS (On-Line Pattern Analysis and Recognition System) (Sammon, 1968) is an interactive system with graphical output designed to enable the user to determine "structure" in a data set and to prescribe regions defining subsets of data for analysis. Each of these named interactive systems works in a restricted and fairly narrow field of multivariate analysis.

STATPAC (Goodenough, 1965) is a light-pen controlled system for data analysis which was originally implemented without graphical output. PEG (Smith, 1970) is an on-line data fitting system which provides least squares fitting by various methods and different options for graphical output. Two systems which are more akin to the system presented here are firstly, GOLDA: a graphical on-line system for data analysis with displays of histograms and two or three dimensional graphs, various multivariate statistical procedures being provided (Harris, 1972). Secondly, a system described by Beaujon (Beaujon, 1970) for illustrating elementary properties of statistical distributions for use as a teaching tool. A demonstration of interactive BMD programs has already been mentioned (Britt, Dixon and Jennrich, 1969).

An interactive version of SPSS (Interactive SPSS, 1974) (Muxworthy, 1976) which should ultimately provide facilities for a wide range of research activities is now being designed. At present it contains a subset of the facilities available in batch SPSS. Currently design specifications only cover the definition of data sets, including complex file structures, data manipulation, simple statistics and regression, although there are plans to include more of the advanced facilities of batch SPSS.

Other interactive statistical systems are described in the Proceedings of Computer Science and Statistics: 8th Annual Conference on the Interface (Frane, 1975), for example SIPS (Avery and Avery, 1975) and TSAM (Chamberlain, 1975). The use of interactive computing for the design and analysis of factorial experiments is discussed by Margolin (Margolin, 1976).

It is evident that large scale interactive systems for multivariate analysis, of which interactive SPSS is perhaps the most plausible prototype, are now in demand and becoming available. The project described in this thesis developed from a recognition of the value of interactive processing in the search for structure in complex data sets. It began before SPSS became available even as a batch system on the UMRCC Regional machine, and contains features which, necessarily are now being incorporated in the interactive version of SPSS. However, the areas of multivariate analysis investigated include cluster analysis, non-linear mapping and other procedures not found in interactive SPSS, from which informative graphical output can be obtained. Indeed unlike interactive SPSS, the graphical aspects of the present system are of paramount importance.

1.5. The value of interaction and interactive graphics in multivariate statistical research

Before outlining the position of the work described in this thesis in relation to the statistical programs already mentioned, there follows a discussion of the implications of interactive computing and interactive graphics for multivariate statistical research. The types of pictures and diagrams which are of value are described, also the possible forms of interaction with these displays which may give insight into the structure of the data set under analysis and a better understanding of the statistical methods involved.

An interactive system for multivariate statistics should in the first instance allow the user to define, create and amend a data file. It should then allow a choice of statistical techniques. Once this decision is made there must be facilities to isolate subsets of variables to examine their influence on the structure of a data set. An interactive system should also allow for the definition of subsets of observations, either because of some common property that such a subset may have, or else so that outliers or cases with missing or suspect data can be located, and if necessary excluded. With a suitably designed interactive system, the user can make on the spot decisions about transformations and set parameters for a particular analysis.

The solution of some problems may require several runs of related analyses. Some multivariate techniques are complementary and the output from one analysis may be used as input to another. A comprehensive interactive statistical system allows the user to make a choice of technique, and to proceed smoothly from one analysis to the next.

In addition, interactive graphics offers features of particular value for data analysis and multivariate statistics, not only in the production of a single picture to summarise results, but also in the presentation of a succession of pictures displaying intermediate results for guidance.

The basic forms of diagrams which are of value are limited in number. The value of graphics in this field lies in the many variations of these, and in the different aspects of results which the same basic diagrams can represent. The two most obvious forms of display for data analysis are conventional histograms to show the distribution of a single variable, and scattergrams to show the relationship between two or three and sometimes four variables. With interactive graphics user defined portions of the histogram can be expanded, the number of intervals altered, a distribution curve superimposed on the histogram, and the histogram replaced by a cumulative distribution curve. Simultaneously displaying the distribution of different subsets of the total population for the same variable, as histograms one above the other has proved useful in the examination of the results of cluster analysis. For 2-d scattergrams axes can be rescaled so that, for instance the data can be standardised for both axes. Additional information can be obtained interactively from scattergrams by specifying labels for the data points, with different labels representing different classes, and also by "windowing" or expanding a portion of the diagram for more detailed examination. The fact that variables used for scattergrams and histograms need not only be variables defining the original data set, but also new variables derived by means of statistical techniques, vastly increases the usefulness of this form of output. A rapid response to requests for

this type of picture is valuable in itself; if they are requested systematically one can usually bear important features in mind and make useful comparisons. If the original data set is being examined in terms of one or two variables at a time, it is important to be able to see the data in terms of all the variables in turn in order to avoid being misled by properties which only occur for a few variables.

There are various ways of displaying n-dimensional data on a 2-dimensional screen, and a more detailed discussion of these is deferred till a later chapter. However, these may result in histograms, scattergrams, points in a cylinder, waveforms, glyphs or metroglyphs (a single symbol for each observation used to represent the value of more than one variable), or points on the surface of a sphere. These representations are usually displayed to enable the viewer to look for some structure in the data they represent.

Rotation of axes of scattergrams, with the points remaining stationary is useful for demonstrating the rotation of orthogonal factors in factor analysis. For other displays, such as the spheres and cylinders mentioned above, the rotation of a 3-d figure enables the viewer to see such a figure from any angle. The speed, the increment and the axis for rotation can all be supplied interactively.

There are problems which require transformations involving one or more parameters whose values are unknown and cannot be computed from the data. To discover optimum conditions, facilities must be available for providing a sequence of values for one or more of these parameters, and for the inspection of results. It must be possible both to increase and to reduce these values, and by varied stepsizes in order to be able to go back and refine the stepsize when an optimum is reached to avoid

a restart. Graphs can be displayed simultaneously to show the optimised function.

The 2-dimensional diagrams from multiple regression analysis showing observed and predicted values and residuals are most relevant for time series analysis (Chamberlain, 1975). There are other possible graphical representations for stepwise multiple regression involving diagrammatic representation of numerical values for guidance on the choice of variables to be included in the equation. Interactive graphics is of obvious value in problems with only one independent variable. Such problems require facilities for transforming the axes and fitting lines to chosen sections of the data. These sections can be chosen by means of a light pen, or any available graphical input device.

A light pen may be used for case identification; for picking out a data point in, for instance, a scattergram representing n dimensions. A histogram for a specified variable can then be displayed highlighting the chosen data point.

1.6. The present system

The system described in this thesis is a set of interrelated interactive Fortran programs written for a refreshed graphical display. It has been designed to demonstrate the value of interactive computer graphics in the field of multivariate statistical analysis, the production of graphical output is therefore a primary function of these programs.

This system has, of necessity, been written as a series of individual programs because of software limitations. There are programs for the initial exploration and verification of a data set, facilities for the selection of subsets of observations and variables for analysis, for the addition of new variables to the data file and for specifying transformations with Fortran-like statements. The statistical procedures which have been programmed for interactive use with graphical output are principal components and factor analysis, with orthogonal rotations, hierarchical and non-hierarchical cluster analysis and discriminant analysis. Where relevant, output from one form of analysis can be filed and used as input to another, and results can be saved for reinput at a later session.

A command language is defined for the use of this system. This involves keyboard input for complex commands; for specifying lists of identifiers, values of parameters, and transformations, and it involves the use of keys for commands of a simpler nature. Keyboard input is fairly cryptic to minimise typing and where possible parameters take default values, which may be changed if necessary. Prompts on the format and meaning of the commands which are relevant at each juncture are continuously displayed. Pictures are available for display at every juncture where it is possible to present results diagrammatically, and where they

may be of use, either to summarise results or as a guide for the next step. These pictures have been designed to show the results of statistical techniques to better effect than tables of numbers alone.

The emphasis is not on file handling techniques, nor on the use of complex files, nor on providing interactive facilities with fast responses for a large number of simultaneous users. The emphasis is on demonstrating how traditional and established statistical methods can be used in the context of interactive graphics to study the nature and structure of a data set.

2. THE DESIGN AND IMPLEMENTATION OF AN INTERACTIVE GRAPHICS SYSTEM FOR DATA ANALYSIS

The available hardware and associated software affect the design of any system. This chapter describes the influence which the available facilities had on the system to be described. The need for a command language to operate an interactive graphical system for data analysis is discussed. The structure of the programs which constitute the present system is described, also the command language designed for use with these programs, and how this language is used to initiate statistical procedures and to obtain and modify graphical output.

2.1. The available hardware and software

Ideally, large scale computer systems should be written in languages which are widely available and they should be machine independent. In practice, even with high level languages like Fortran, local idiosyncracies make complete portability impossible. With systems which additionally involve the use of unusual devices such as graphical displays, this problem becomes more acute. This is due to the different software interfaces between the languages and the various graphical devices available. (Although with the increased availability of systems of graphics subroutines such as GHOST (Prior, 1972) and GINO-F (GINO-F, 1975), these problems are being minimised.) Different implementations of high level languages under different operating systems have implications for the methods of accessing programs and for file handling. In view of these differences, compromises have to be made in the design of large scale graphics systems in order to make efficient use of the available hardware and operating system. Therefore because of the influence which the

availability of hardware has on the design and implementation of a system, this section desribes the available hardware and associated software.

The graphical device used for display purposes is an Elliott 4280 Graphical Display Unit attached to an ICL 4130. This is a refreshed CRT with one screen only, eight switches or keys, a light sensitive pen and a console typewriter. The light pen may be used to define points on the screen by using it in conjunction with the tracking cross. Unlike some graphical display units, the console typewriter is not incorporated in the display unit and characters typed in do not necessarily appear on the display screen. There are three (hardware) character sizes for displaying text. Hard copy of display output can be taken on a digital plotter, or if there is only alphanumeric data this can be output to a line printer.

Two interactive languages were available, Basic and Pop-2. The particular implementation of Basic with limited input and output facilities made it unsuitable for such a large and complex system. Currently Pop-2 is not widely used and consequently there is not yet a range of numerical algorithms available. Moreover, there were no software routines written to interface with the graphical display written for use with either of these languages.

This system has been written in Fortran, which in view of the widespread use of this language, makes the programs more portable. The software interface for the graphics was written for use with Fortran, and numerical algorithms are available. When the system was first designed there were no packages of graphics subroutines implemented for the 4130. The basic subroutines used for drawing pictures are those defined in the ICL 4100 Technical Manual (ICL 4100 Technical Manual, 1969).

The use of Fortran influenced the structure of this system since the particular implementation and the associated operating system (DES BATCH) imposed certain restrictions on the design. There are limitations on the size of the program, in spite of overlaying or segmentation facilities. Only program code can be segmented, constants and workspace are not overlayed. The space required for data and display purposes is relatively large. In Fortran, array sizes have to be defined at compile time, this restricts dynamic use of storage space for arrays. Two separate runs of the same program may utilise the total array space in different ways, and a language which allocates arrays dynamically makes more efficient use of the available space. It is possible to simulate this in Fortran by programming for it, but that in itself uses space and would have been too complex for this project as implemented on the available machine to have been worthwhile.

The system consists of a series of programs which are self-contained units, although files are passed from one program to the next. There can be no dynamic interaction with the operating system, it is not possible to convey variable information to the operating system from within a program at run time. This has implications for the interactive choice of programs. Although the user can choose which program to load next, this choice cannot be transmitted to the operating system by program control, for instance from within the previous program. The choice has to be made from prior knowledge of the programs available.

The hardware and software facilities for creating and accessing files were limited. There was a shortage of disc storage space and magnetic tapes had to be extensively used for data. Although the facilities available have been used as efficiently as possible, the

emphasis is not on file handling techniques. The available file handling facilities are sufficient for the task undertaken, and if carefully used with this interactive system they can provide an acceptable response to requests from the user.

2.2. The requirements of an interactive command language for use with graphics for data analysis

This section discusses the need for a command language for interactive graphics, and the particular requirements of such a language for data analysis and for multivariate statistical research. Lastly it describes the properties which any interactive command language should have.

A comprehensive system for interactive graphics for a given area of research requires the execution of complex sets of instructions. Specifically designed command languages remove the need for having to issue each of these instructions individually. While being rigid enough to avoid the time consuming process of issuing detailed repetitious instructions, command languages must be sufficiently flexible within certain limits to allow the solution of the problem under investigation.

For any interactive graphics system commands must be provided for the administration and overall management of the system, to restart, exit, to save specified results or to take hard copy for example. For design projects where interactive graphics are used to build up a model, a command language must be designed for the detailed modification of a picture. For instance for the use of Multipatch and Multiobject, instructions are provided to define and modify the shape of individual patches. In systems written for electronic engineering or architectural applications commands are designed for the introduction or repositioning or deletion of individual components. For simulation problems commands must be available to set parameters and to initiate the simulated process. To ensure that interactive systems are flexible and convenient to use the associated command languages have to be designed to suit each particular type of problem.

An interactive command language for data analysis must be designed to meet the requirements of multivariate statistical techniques. It must provide control over processing and output at certain points, these points being determined by the nature of the technique.

Specifically an interactive command language must provide control for the following facilities.

1. The definition and verification of a data set.
2. The definition of transformations on the data.
3. The choice of multivariate statistical techniques.
4. The definition of parameters for these techniques.
5. The initiation of a procedure, and in some cases the interruption of a procedure.
6. The definition of what is to be displayed and how it is to be displayed within certain prescribed limits.
7. Dynamic picture manipulation.
8. The comparison of results.
9. The filing of intermediate results for use on subsequent occasions.

This list is not exhaustive, but indicates the topics which had to be given prime consideration when designing this interactive graphics system and the command language which is an integral part of it.

Decisions as to exactly where to make provision for user control are based on the nature of the various techniques and experience in the use of these. It is often unnecessary to provide facilities to specify all the parameters involved in a calculation, although obviously options must exist for significant parameters. Similarly it is unnecessary to make provision for all forms of output at each point for each technique. Experience has shown that in some places, for instance during unavoidable lengthy calculations, additional facilities are required to allow for

user intervention and more specific control. In other instances the provision of additional control or output facilities either complicates operations too much from the user's point of view without adding any real enhancements, or else it is too complex and takes more core space or time than its inclusion warrants. The amount of control over processing and output must be tailored to each individual statistical procedure, and its implementation on a given machine configuration.

An important requirement of a command language is that it be easy to use. It must facilitate a user's specification of what is required of the system: it must be easy to indicate a particular technique, to set up parameters, to initiate calculations and to specify the form of output. If the relevance of commands is to be easily understood, instructions in the use of the individual procedures must be clear. How to issue to commands and the rules of syntax must be clear, as must the implications of issuing a command. For ease of use there must be uniformity in the format and the syntax of commands, and under similar conditions it should be possible to issue syntactically similar commands. Furthermore, since text can be rapidly produced, an interactive graphics system can be made much easier to use with a display of comprehensive and continually revised instructions.

Errors in the syntax of commands and errors of context should be recognised and reported to the user, with a clear concise message, as soon as they have been detected. There should be complete recovery from such errors and an error which has been detected in a command should not affect commands previously given.

Lastly, the language must be extendable. It should be designed in such a way that, in terms of the subject for which it is written,

additional facilities can be provided by making only minimal extensions to the syntax.

2.3. The organisation of this system

The broad aspects of the design of this system as well as the detail have been influenced by available hardware and software. In spite of this, it has been designed in such a way that it should be possible to run the system elsewhere with a small number of changes, provided that the basic hardware and software routines to interface with Fortran are available. To run it with a storage tube rather than a refresh tube would require further programming effort because some use, although not extensive, has been made of facilities which are only available on a refresh tube. Apart from these factors and bearing in mind that the programs would be unlikely to make efficient use of filing facilities elsewhere, this system should be transportable.

The size and complexity of this project as well as the available hardware and software meant that the necessary administrative and statistical facilities had to be provided in several separate programs with intermediate files. The overall design of the system involved decisions about the purpose of each program, the structure of the programs, the overall format of graphical displays, the format and syntax of commands and the structure and organisation of files. Once a framework had been defined, the administrative programs and programs for some statistical methods were written. It was then possible to provide additional facilities within these programs and to add new programs for further multivariate statistical methods without having to alter the overall design or extend the syntax. The fact that these extensions were not difficult to make was partly due to the fact that a large proportion of the rules of syntax had to be defined for the administrative programs and any additional definitions necessary for the statistical programs are common to many of them.

As already indicated, this system has had to be written as a set of individual programs which fall into two categories, administrative and statistical. Three programs fall within the administrative category. The first is a program known as RDDATA, which has to be run at the start of each session, and sets up data files and administrative files. Ideally, certain facilities such as transformations should be made universally available, but shortage of space meant that they had to be provided in one program only, and the results filed for subsequent use. Transformations are included in a second administrative program, FILREC, which also has facilities for extending an existing data file with related data from an external source. This program can be used to isolate subsets of observations for separate analysis. The third program in this category is designed to display the data as scattergrams and histograms. It can be used at any stage in the solution of a problem. Extensive facilities in the statistical programs for adding new derived variables to the data file means that this program can be used to examine these new variables in some detail. Again a shortage of space and the limitations of the operating system meant that these facilities had to be confined to one program when it would have been more useful to have had them readily available within each statistical program. Even if this had been possible, it would still have been necessary to be able to file the results for subsequent examination or input to other programs.

Turning now to the statistical programs, a discussion of which multivariate statistical methods have been chosen and why, is deferred until a later chapter, however a few general observations can be made at this point. Each statistical program covers one topic or several closely related topics, particularly where the similarities involve the use of the same code and where there is sufficient space. The size of store imposes a limit on the

number of different forms of output which can be provided, although overlaying program code increases the amount of code which can be included. However, too many overlays increases response time to such an extent that the response to each character typed on the console is unacceptably slow. Within each statistical program a subset of variables can be chosen for analysis, and a subset of observations can be chosen provided that they can be explicitly named. If a subset of observations is to be chosen on the basis of the values of the data for each observation, then the second of the administrative programs, FILREC, must be used to create a new input file for that subset. From the user's point of view the mode of operation of these programs is as uniform as possible within the limits imposed by the differences in the various techniques.

Administrative files which provide background information for each program are binary disc files. The data files which are used as input to each program are also held in binary files which can either be on magnetic tape or disc. The data files are created with each record containing all the information for one observation and the records are read sequentially. The reasons for filing the data in this way are largely historical. This is a traditional format for filing statistical data and the input and the statistical calculations for this system are derived from an earlier batch system which was written to accept data in this form. Sort/merge facilities have been required for some projects to maintain updated input files and the only sort/merge programs available required files to be in this format and they had to be on magnetic tape. For the future, improved file handling facilities will allow for more complex structures for the data file and possibly for the transposition of the data file for faster access. Currently when transformations are made or new variables added, a new file is created adding the values for the new variables to each record. Transposition of

the data matrix would facilitate filing both transformations and new variables, particularly the latter which would simply mean the addition of an extra row to the file. In some programs, subfiles of data are created, for use within those programs only, for faster access for display purposes.

Up to 512 variables may be held on the data files. The statistical programs all cater for a maximum of 60 variables. There is no limit to the number of observations which may be held on file. The only limits for these occur in the number of points which may be simultaneously displayed, 4000 except in a few instances when a lower limit has been clearly specified.

2.4. The structure of programs and the purpose of commands

There are two basic processes involved in these programs; the specification and initiation of statistical procedures, and the presentation of results, which in the main consists of the production and modification of diagrams. The structure of the programs and the methods of issuing commands reflect the identification of these two processes. Other factors have also influenced the structure, for instance the fact that this system had to be written as a set of separate programs and that the choice of programs cannot be interactive. To have been able to provide all the facilities in one program and interactively choose and load only those parts which are currently required would have simplified the structure. As already indicated the system is a set of interrelated programs each performing either a separate function or several related functions.

Once the purpose of some individual programs had been defined, it became apparent that the process of specifying and initiating procedures could be divided into different stages or levels. The necessary commands fall into clearly defined groups, and these groups must be specified in a given sequence. After the specification and initiation of procedures, the presentation of results occurs at the last level of operation.

At one level one of the following may be requested.

(a1) A choice of statistical method may be made.

(a2) Parameters may be defined and a calculation initiated.

(a3) Parameters may be defined which will remain constant at a subsequent level and which it would be inconvenient to have to keep redefining.

(a4) The form of output may be chosen and details defined.

The transference from one level, level I, to a subsequent level, level I+1 will represent in each case.

(b1) A further step in the relevant method.

(b2) The termination of the calculations.

(b3) The availability of facilities with which to make more detailed definitions.

(b4) The results will be displayed in the form requested.

In each of the instances (a2) - (a4) several commands may have to be given before transferring to the next level, and some may be fairly complex. Typing these commands on the graphics console allows sufficient flexibility for conveying all the necessary information. (a1) is a simple choice, a choice which could be made with a light pen, similarly commands specifying lists of identifiers could be issued with the light pen, but for consistency and ease of use these commands are all issued via the console. Once a diagram or table of values is displayed, any alterations which are required are specified by means of the switches, or in some cases, by use of these in conjunction with the light pen.

Given that the programs were to be structured in this way, it was decided that they would present two basic types of picture. The first type consists of lists of options to which the user must respond with commands typed on the console, these are used for specifying the items indicated in (a1) - (a4). The second type of picture consists of either diagrams or tables of numerical values, when these are displayed only the keys may be used and where relevant the light pen. To the first type of picture the user must respond with option commands, and to the second with key commands.

To minimise the number of user responses, when lists of options are displayed and several option commands may be given before transferring to the next level, default options have been introduced. These indicate the default values which parameters will take unless there are instructions to the contrary. Therefore if all the default options are satisfactory, only the command indicating a transfer to the next level need be issued. If not all the default options are satisfactory, commands need only be given for those which require changing.

If the transfer to the next level indicates that calculations are likely to be lengthy, facilities are provided to indicate how the calculations are progressing and to allow for user intervention. Whether or not these are provided, once the calculations are complete, a further list of options is displayed to indicate the facilities for displaying output and, if relevant, for filing results.

In some circumstances lists of options have to be provided to allow for the specification of details for diagrams, and these may be supplied and terminated with a command to transfer to the lowest level and display the diagram. Once the diagram is on the screen, if it is to be altered or a new diagram displayed and the change is straightforward and systematic, key commands are the most convenient way of specifying these changes. Examples of systematic changes of pictures are: displaying the data set in terms of the next variable in a list, displaying the next 'page' of a table, or a simple modification to a picture, for instance, altering the number of intervals for a histogram or superimposing a distribution curve on a histogram. If the change is more complex a list of options is displayed (by issuing a key command) and option commands typed in.

Each program is loaded in a conventional manner, by supplying cards or card images to the operating system. Once it is loaded it works interactively until the user specifies, with a command, that an exit should be made. The next program may then be loaded. Obviously, where it has been possible to include several related topics in one program it is unnecessary to exit and re-enter if two such topics are required consecutively. In order to make this possible and to provide more user control within each program, control may be passed back to an earlier level of operation, that is, from level I to level I-1.

2.5. The syntax and decoding of commands

The purpose of this section is to give a more detailed description of the operation of programs in so far as these details are common to all programs. This involves the syntax of commands and how commands may be used. The section ends with a brief discussion on the decoding of commands. Detailed specifications for each of the programs including definitions of exactly which commands may be issued in all circumstances are given in a User's Manual (Appendix 2).

As soon as each program is loaded, a list of options is displayed. Four flashing asterisks in the top right and left hand corners of the screen indicate that the system is waiting for a response from the user, either by giving an option command if a list of options is displayed, or by giving a key command if a diagram is displayed.

For each item in the list of options (each item is preceded by <letter>)) one or more commands may be given. The current level of operation is displayed at the top of the screen and a digit indicating this level is output to the console. An option command must then be given in the form

<letter>); or <letter>)<list of items>;

where the letter must be one of those which appear at the start of each option on the screen. This will relate the command to an option on the screen. They will be related in that the command is intended to have a function which has been indicated to the user by the text for the relevant option. There is no functional relationship between the text on the screen and the issuing of a command. The list of options merely indicates what will happen if certain commands are given while that list of options is displayed. Some of these commands may not be effective until the list of

options is erased from the screen. But in order that a command shall have the function indicated by an option, the command must be issued while that option is displayed.

The first form of option command

<letter>;

has several functions. Firstly, it can be used to indicate a choice of route through the program if such a choice is available. It is used to transfer to another level of operation, this may in some circumstances imply the initiation of a calculation, or that a diagram is to be displayed. The third use for this form of command is for specifying parameters which can only take the value Ø or 1. This applies to program switches for printing as well as for computational parameters.

The second form of option command

<letter>)<list of items>;

is provided for commands where additional information has to be conveyed to the program. These commands may require lists of variable identifiers, or lists of record identifiers, or assignment statements for transformations, or selective IF statements for forming subsets of observations, or simply numbers. This will appear in the list of items, and a definition of the syntactic elements which may appear in a list of items is given in Appendix 1.

Certain basic universal commands have been built into the system to serve the same operational function in similar situations. These are as follows.

The command

X);

is used to transfer control to a (numerically) higher level of operation

in circumstances where several commands may be given before this transfer is made. This command is used to display diagrams.

The command

```
Y);
```

is used to transfer control to a lower level.

The command

```
Z);
```

is used to exit from all programs.

Once a diagram is displayed the keys are used for control. Key 1 is always used to obtain hard copy and key 8 is used to redisplay the latest list of options.

The system provides some protection against user errors. Firstly, commands are held in a buffer and are not transmitted to the program until the semi-colon terminating the command is typed. There is an escape character which allows a command to be restarted if an error is detected before the semi-colon is typed. If syntax errors occur when option commands are typed in, an error message is output on the console and the command may be restarted. Similarly, if an option command is given starting with a letter which is not currently displayed at the start of an item, an error message is output. Error messages will be output if, in some circumstances, too many commands are given and there is insufficient space to store the information.

An integer function called ACTION is provided for use with the graphical display (ICL 4100 Technical Manual, 1969). This function can be set up to expect any combination of four types of interrupt; a single character typed on the console, a key depressed, a specified number of end-of-frame interrupts or a light pen 'see', of which only the first three are

relevant here. Within these programs if a list of options is displayed, ACTION is set up to expect either 8 end-of-frame interrupts or a character typed on the console. If the end-of-frame interrupts occur before the character is typed the intensity of the asterisks at the top of the screen is changed from dim to bright or vice-versa. A command or a line of a command is collected in a buffer with repeated calls to ACTION, allowing it to be easily restarted if the escape character is typed. A subroutine named GTITEM, which is used by all the programs is called every time an item is required (an item is defined in Appendix 1, §1).

3. ADMINISTRATIVE PROGRAMS: THE FACILITIES THEY PROVIDE

The administrative programs to prepare date files, to isolate subsets of data and to make transformations are described, also the program used to display histograms and scattergrams; a program extensively used in later chapters. The problems of displaying n-dimensional data on a 2-dimensional screen are discussed in the last section.

3.1. Input files and the program RDDATA

Data for input to all programs which accept raw data, rather than a matrix derived from the raw data, must be presented as a rectangular matrix. Each row of this data matrix represents an observation or case, which consists of measurements for a named set of variables. This simple data structure is the only form of input provided for since hardware and software limitations made the introduction of more complex structures impractical.

This matrix is preceded by specifications for the data. These specifications include the number of variables, the number of observations, the format of the data and, optionally, a list of user variable names. Variables can be of two types, either numeric or alphanumeric. The variables used for input to all statistical procedures must be numeric. There are therefore facilities, not provided at the time of initial input, but available in another program, for recoding alphanumeric variables into numeric form. Alphanumeric variables may also be used subsequently for identifying and isolating particular observations. System variable names which may be used in commands in the form

V<unsigned integer>

are supplied automatically and sequentially, these may be used

interchangeably with user variable names. Record identifiers, which may also be used in commands in the form

R<unsigned integer>

for individual record identification, are also supplied sequentially. (The syntax for variable identifiers and record identifiers for use in commands are defined in Appendix 1.)

In the first instance data may be input from any peripheral and the program RDDATA will create a master input file (m.i.f.) with this data. At the start of a session the m.i.f. is the current input file (c.i.f.) for all programs. If at any time new variables are added or data transformations specified, a new file is created which becomes the c.i.f. The m.i.f. can be preserved for later use. The new c.i.f. may also be preserved for use in a later session. RDDATA must be run at the start of each session either to create an m.i.f. or to establish and check an existing c.i.f., and the user must respond to console messages to indicate whether or not an m.i.f. exists. The source of commands is also given in response to a console message.

Background files, to which the user does not have access, containing information about the current input file (the variables and their identifiers in a hash table, also the observations) are created by this program to be read in at the start of each program subsequently loaded. Once RDDATA has been successfully run other programs can be loaded. If these involve the creation and filing of new variables, or the isolation of a subset of variables or observations, a new current input file is created. On a new c.i.f. variables and records preserve their original identifiers, thus if some variables are not included in the new file their system variable names will be absent from the sequence. If new

variables are added they will automatically be given system variable names and user variable names must be supplied. The background files have to be adjusted in these circumstances since they contain information for the c.i.f.

Matrices created from the file of records on the c.i.f. which may be required for input to the programs are filed. Lists of the variables and records used to create such a matrix are also filed.

3.2. Transformations and the selection of data

During the course of an analysis transformations may be required to create new variables which are functions of existing variables, or to recode variables where they are not in a form suitable for input to statistical procedures. In addition to providing these transformations the program FILREC provides facilities for the addition of new variables to the data file and for the selection of subsets of data for analysis. The implementation of these facilities is limited in that they are not universally provided in all programs and the amendments are made record by record to a sequential file, nevertheless they are adequate for the fundamental requirements of data analysis.

The creation of new variables which are functions of variables already defined on the current input file is achieved by means of Fortran-like assignment statements which may include references to functions such as ABS, AINT, LOG, SQRT, etc. These new variables may be used for input to statistical procedures or simply to examine, in a scattergram, a known bivariate relationship. The syntax for these statements is defined in Appendix 1, §4. One command will contain only one assignment statement, and commands for these transformations must be issued in the form

<letter>)<assignment statement>;

The assignment statement may be for the definition of new variables or for the redefinition of existing variables. Each assignment statement is executed once for each observation on the c.i.f.

Conditional transformations may be used for recoding alphanumeric variables or for classifying continuous variables. In this case the execution of the transformation depends on the result of a logical test.

Each assignment statement is preceded by a logical expression involving variables previously defined. Alphanumeric constants may be used in logical expressions to identify chosen values of alphanumeric variables. The syntax for a logical expression is defined in Appendix 1, §5. For each observation on the c.i.f. the logical expression is evaluated, if the result of this expression is true the transformation is executed, if it is false the transformation is not executed and the variable on the left hand side of the assignment statement retains its previous value if it is already defined, otherwise it remains undefined. Only one conditional transformation may appear in a command, and the syntax for these commands is

<letter>)IF <logical expression>,<assignment statement>;

New variables may be added to the c.i.f. from an external source, for instance co-ordinates or new variables derived elsewhere for a given data set. These are specified by giving the appropriate command which will include a list of new user variable names. The data may be introduced on any medium; cards, paper tape, disc file, etc.

A facility for the selection of subsets of observations for analysis is provided by means of commands which are given as follows

<letter>)IF <logical expression>;

If the logical expression is true for any observation then it will be included on the new c.i.f. By means of these commands observations are selected to form a subset whose members have a common property, rather than having to explicitly name the members of such a subset. Once a subset has been examined, a further subset can be defined by returning to the original input file and operating this program with a different set of commands.

Once all the amendments have been specified the c.i.f. is read, one record or observation at a time and a new c.i.f. created. For each observation any logical commands for the isolation of subsets are evaluated first, followed by commands for any unconditional or conditional assignment statements. After the first three types of commands have been dealt with any variables which are to be added are read in and appended to any records written to the new c.i.f.

To improve efficiency in the evaluation of arithmetic and logical expressions, the operators and operands which occur in these expressions are held in a stack in reverse Polish form. In this program the final stack for the reverse Polish is an integer array where the operands have positive entries. These entries are pointers either to the array of data values which will be taken from the c.i.f. when the expressions are evaluated, or they are pointers to a real array of constants. The operators and function references have coded negative entries in the stack. When these expressions are evaluated intermediate values have to be held. For logical expressions the logical results of relational expressions have to be held in addition to the numerical results of arithmetic expressions. A real array is equivalenced (in the Fortran sense) to a logical array for these intermediate values. Pointers to this array are maintained, indicating whether or not an entry is real or logical so that a program switch will direct control either to an arithmetic assignment statement or a logical assignment statement, whichever is appropriate.

Exit from this program is automatic and the file most recently created becomes the new c.i.f.

3.3. Displaying histograms and scattergrams

The program for displaying data as histograms and scattergrams is known as DISVAR. Data used for either of these forms of display is taken from the c.i.f. This file may contain, apart from the original data set, non-linear mapping co-ordinates, principal component scores, factor scores, co-ordinates for canonical variates and the results of cluster analysis. The results of cluster analysis will be in the form of cluster or group labels which indicate to which cluster each individual belongs. In view of the extensive use of this program for the examination of results, the facilities provided are described in some detail and their use is demonstrated in a preliminary way with Fisher's Iris data.

The facilities available in this program are listed below. Further details as to how these and other more specific facilities have been implemented for histograms and scattergrams are given later. The first three items in this list indicate different ways of displaying the same data values.

1) The display may be a standard one where each observation from the c.i.f. is included and there is no distinction between observations except in terms of the variable or variables which are used as the basis for the diagram.

2) Individuals may be uniquely identified by their record identifiers. This is only relevant for scattergrams.

3) Groups of individuals may be defined at run time, and later identified on the display, by means of a series of logical expressions (for the definition of these see Appendix 1, §5). The first command which contains a logical expression defines group 1, and all those observations for which this logical expression is true belong to group 1. The remaining observations are tested to see if they belong to group 2 and so on for a

maximum of ten groups. If an observation does not satisfy any of these logical expressions it will not appear. The logical expressions may include any numeric or alphanumeric constants and variable identifiers for any variable on the c.i.f. Thus if a variable is included on the c.i.f. which is, in essence, a label with k possible integer values, its variable identifier can be used in a series of k commands, each containing a logical expression, to define k groups. Not all k groups have to be displayed simultaneously, any subset can be displayed.

The remaining two items in this list indicate ways in which the pictures can be manipulated.

4) A portion of the picture displayed on the screen may be magnified to reveal more detail.

5) The data may be rapidly displayed in terms of the next variable in a user defined list and in the same mode as it is currently being displayed.

Fisher's Iris data (Fisher, 1936) consists of a set of four measurements obtained from Iris flowers; sepal length, sepal width, petal length and petal width. 50 sets of measurements were taken from each of three species of Iris; setosa, versicolor and virginica. A data file was created with these measurements and a fifth variable was included; a group identifier with values 1 - 3 indicating to which of the three species each observation belongs. NLM co-ordinates were evaluated for 2 dimensions and these were added to the data file as two further variables. The variables on the c.i.f. were therefore

	System variable name	User variable name
1. Sepal length	V1	SEPALLEN
2. Sepal width	V2	SEPALWID
3. Petal length	V3	PETALLEN
4. Petal width	V4	PETALWID
5. Species identifier	V5	IDENT
6. 1st set of NLM co-ords	V6	NLM1
7. 2nd set of NLM co-ords	V7	NLM2

Three groups were defined by typing the commands

```
C) IF IDENT EQ 1;
C) IF IDENT EQ 2;
C) IF IDENT EQ 2;
```

at the appropriate point in the program. For a given observation, if the first expression is true then that observation belongs to the species Iris setosa, if the second is true then it belongs to the species Iris versicolor and if the third is true it belongs to Iris virginica.

Histograms

One variable identifier has to be given initially to define the first histogram, in this case it could be V1 or SEPALLEN. For a standard run all the data is displayed in one histogram initially with k*10 intervals, where k is the number of groups defined. If no groups are defined k = 1. Whenever a histogram is displayed the number of intervals may be increased or decreased by repeatedly pressing a key. A normal distribution curve may be superimposed on the histogram, by pressing a key. The area under this curve is equal to the area of the histogram and the mean and standard deviation are determined by the data used for the histogram. Figure 3.1 shows histograms and normal curves for the first variable of the Iris data; sepal length. Figure 3.1(a) shows one histogram for all the data with the normal distribution curve.

If groups have been defined they may be displayed individually, but simultaneously, as in Figure 3.1(b) where the three species of Iris are represented by the three different histograms for sepal length. The normal curves in this case replace the histograms and are superimposed on one another. In the first instance, Figure 3.1(c) the sum of the areas under these curves is equal to the area under the curve in Figure 3.1(a). However, if there are 10 such curves and if some of them have large standard

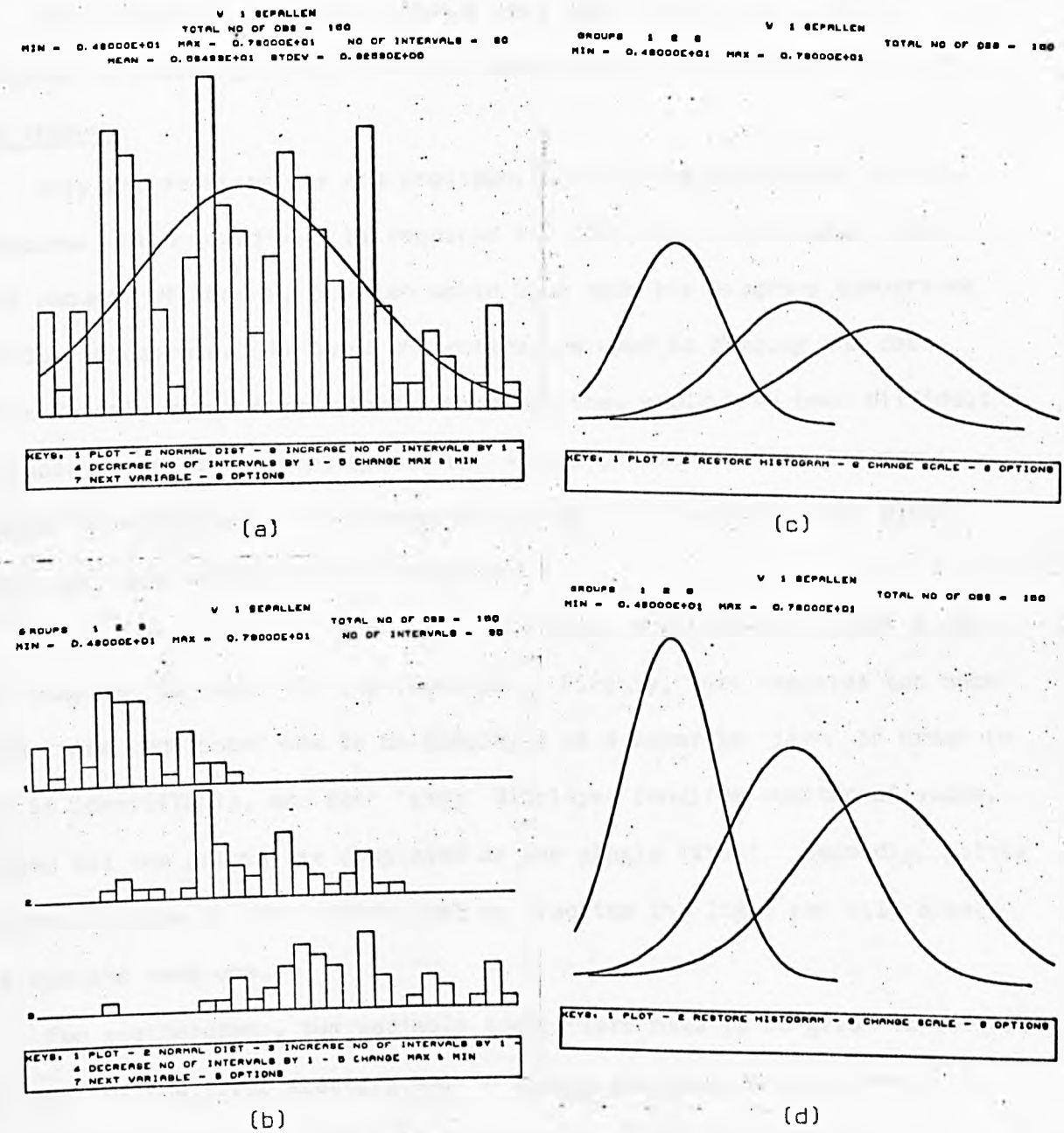


Figure 3.1. Iris data: histograms and normal curves for sepal length

deviations they may not be easily visible, and therefore they can be magnified by pressing a key, Figure 3.1(d).

Specified portions of the histograms can be displayed by typing new maximum and minimum values for the data which is to be included in the histogram.

Histograms for the next variable in a user defined list can be displayed by pressing a key. In this case such a list could be V1 TO V4.

Scattergrams

Only 2-d scattergrams are provided, since 3-d scattergrams and the procedures which would also be required for rotating 3-d pictures require large amounts of core store which would have made the programs cumbersome and slow to operate. Hardware characters are used to display the data points in only three fixed sizes, therefore they would have been difficult to display accurately in perspective. It was considered that the rapid display of successive 2-d scattergrams was of more value than the slower display of more complex 3-d scattergrams.

It is not practical to display individual data points in such a way that they can be 'seen' by the light pen. Firstly, this requires too much space since each point has to be displayed as a separate 'item' in order to make it identifiable, and each 'item' displayed requires additional space. Instead all the points are displayed as one single 'item'. Secondly, points can lie so close to one another that in practice the light pen will always pick out the same one.

For scattergrams, two variable identifiers have to be given to define the axes for the first scattergram. A single scattergram illustrates the Iris data and this shows the data transformed to NLM co-ordinates in Figure 3.2.

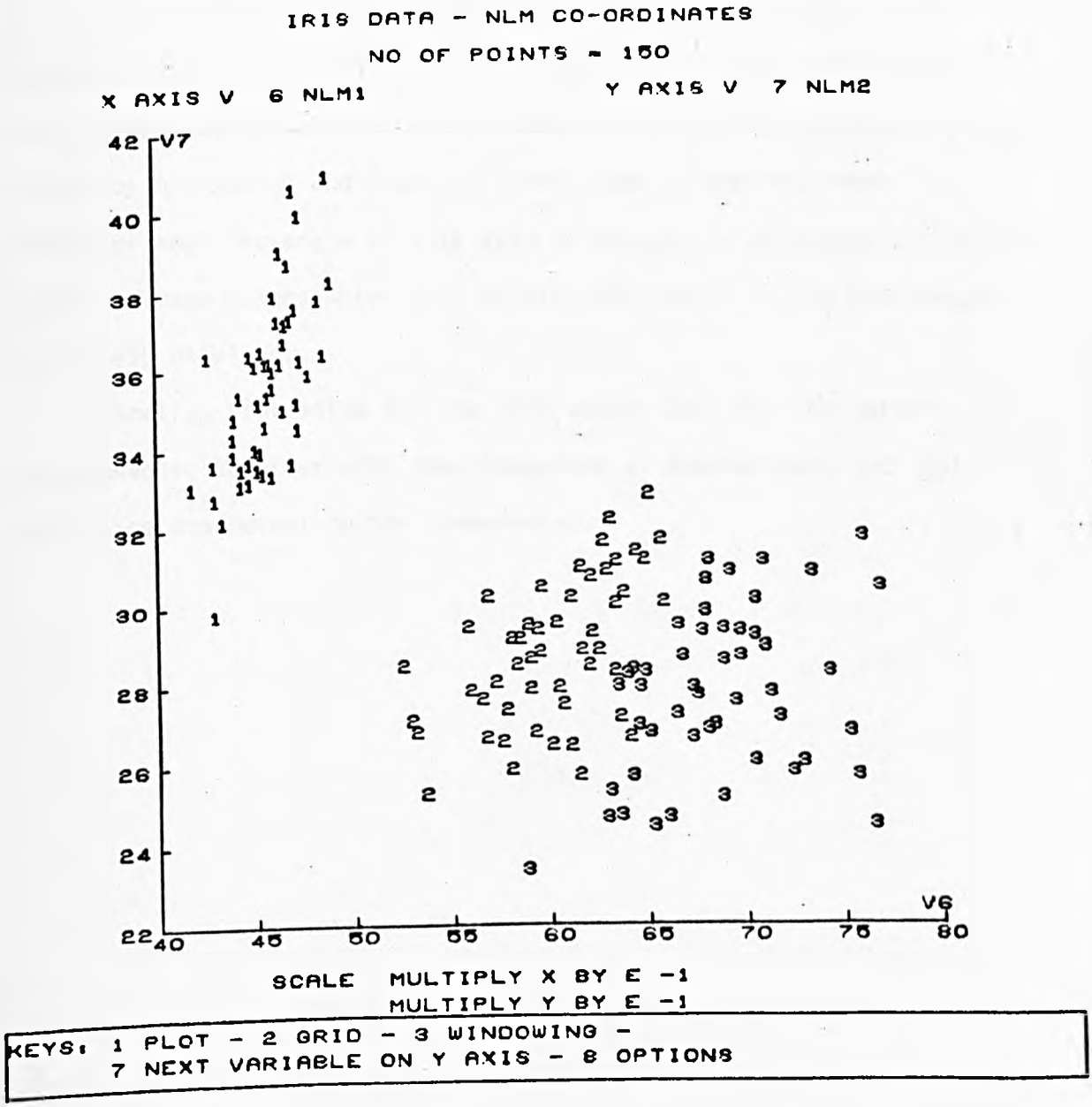


Figure 3.2. Iris data: NLM co-ordinates

Data points may be displayed by a single symbol, say O, or by the integer label appearing in their record identifier, or by their group labels if groups have been defined as in Figure 3.2. The light pen can be used (with a tracking cross) to define a rectangular portion of the scattergram to be magnified. Key 7 may be pressed to produce a new scattergram with the same variable for the X-axis, but a new variable for the Y-axis.

An additional facility is available for scattergrams; to count the number of data points which occur in each region of the scattergram. If key 2 is pressed the axes remain and the data points are replaced by a grid formed by horizontal and vertical lines drawn at each tic-mark. In the centre of each rectangle of this grid an integer is displayed giving the number of data points which fall within that region of the scattergram previously displayed.

Finally, statistics for the data values used for the current display are presented together with the histograms or scattergrams and additional statistics are output to the lineprinter.

3.4. Displaying n-dimensional data in less than n dimensions

We may define a data set as a set of n variables $x_1, \ldots x_n$ and for each variable a set of m observations with values

$$x_{ji},\ j = 1 \ldots m,\ i = 1 \ldots n.$$

A major problem in using graphics for data analysis arises in portraying properties of n-dimensional data in 1, 2 or at most 3 dimensions. When the data is projected onto, say, a 2-d subspace of the original n-dimensional space, although such pictures can be of great value, all the information contained in the data relating to the remaining variables is ignored. It requires a total of $\frac{1}{2}n(n-1)$ pictures to display the data in all possible 2-d subspaces of the original n-dimensional space.

Some statistical techniques including principal components, factor analysis and canonical analysis transform the data to a new set of variables. If any pair of these are chosen as axes for a 2-d scattergram for instance, then the information contained in the remaining new variables is similarly ignored. For some principal components solutions the two vectors associated with the two largest eigenvalues account for a very large proportion of the total variance, in which case a 2-d scattergram with these two vectors as axes might be considered an adequate representation of the data. However, in general a maximum of three such vectors will not provide an adequate representation of the data.

Scattergrams can be further developed to include metroglyphs, or symbols used to represent more than one variable. Instead of each data point being represented as a point in a 2-d picture, each point appears as a symbol. If then two perpendicular axes represent the first two variables and the symbol placed in the appropriate position, the values

of a number of other variables can be displayed simultaneously. The symbol may be a straight line whose length is proportional to a third variable and whose angle of tilt is proportional to a fourth (Ball and Hall, 1970). Alternatively it may be a triangle the length of whose base represents one variable and the height and orientation a further two (Pickett and White, 1966). The additional variables may be represented as lines radiating from a circle (Anderson, 1960), the lines for one variable having the same position on each circle and the length of the line representing the value of the variable for each observation. The resultant pictures consisting of a collection of symbols, may give an impression of different textures. Different textures, e.g. jaggedness, softness, may be confined to different regions of the picture indicating some form of homogeneity within a region and also between regions.

Symbols constructed as in these examples cannot easily be used to display more than ten to twelve variables and there is some loss of information particularly if parts of the symbol represent continuous variables.

A more intricate use of symbols is made by Chernoff in the design of faces to represent points in an n-dimensional space (Chernoff, 1971). The characteristics of each face are determined by the position of each point in the n-dimensional space. The illustrations of this technique allow at most eighteen variables to be represented.

Tryon and Bailey (Tryon and Bailey, 1970) use spheres to demonstrate the results of factor analysis. The factor loadings for each variable are normalised so that the sum of their squares is unity, and therefore if there are k factors and variables are displayed as points plotted on a hypersphere of k dimensions the points will all be on the surface.

Spheres depicting any combination of three factors can be displayed, and only those points which lie at or near the surface of each sphere are shown. Circular contour lines drawn on the sphere, and arcs joining the ends of the three axes provide visual aids to understanding the 3-dimensional picture. These spheres are produced by batch programs and can be rotated to display the maximum number of points before output.

Non-linear mapping (NLM) (Sammon, 1969) is a technique extensively used for this thesis for mapping the data points from an n-dimensional space onto a p-dimensional space, where p<n, while preserving the approximate structure of the data set. The object of this mapping is to preserve the relative magnitudes of interpoint distances. If

$$y_{jl},\ j = 1, \ldots m,\ l = 1, \ldots p$$

represent the co-ordinates in the new p-dimensional space then the error term

$$E = \frac{1}{\sum_{j<k}^{m} (d_{jk})} \sum_{j<k}^{m} \frac{(d_{jk} - d^*_{jk})^2}{d_{jk}}$$

where $d_{jk} = \left[\sum_{i=1}^{n} (x_{ji} - x_{ki})^2\right]^{\frac{1}{2}}$ and $d^*_{jk} = \left[\sum_{l=1}^{p} (y_{jl} - y_{kl})^2\right]^{\frac{1}{2}}$

will give a measure of the differences between the interpoint distances in the two configurations. The minimization of this error involves minimizing the function E with respect to the m x p variables y_{jl}. The computational technique used was minimization by the conjugate gradient method of

Fletcher and Reeves (Fletcher and Reeves, 1964). The output, a set of p (= 2) co-ordinates for each of the m observations in the data set was appended to the c.i.f. using the program FILREC. It was then possible to examine data sets in terms of these co-ordinates of the 2-d mapping. For a given data set different starting positions have nearly always produced different minima. But for the data for which these co-ordinates have been evaluated these different minima appear to have had little effect in terms of the relative positions of the data points in the resultant 2-d scattergrams. This procedure provides another way of examining a data set and the resultant pictures should be studied in conjunction with other pictures. Demonstrations of its use are given later when particular data sets are discussed.

A sequential mapping of points onto a plane such that 2m-3 of the original $\frac{1}{2}m\,(m-1)$ interpoint distances are exactly preserved is presented by Lee et al (Lee, Slagle and Blum, 1977).

Multidimensional scaling (Kruskel, 1964 a, b) is another technique for representing n-dimensional data in a reduced number of dimensions, p. Suppose there is an m x m matrix $\underline{A}$ of dissimilarities or similarities whose elements a_{jk} represent, in some sense, the experimental differences or similarities between the observations j and k. The object of this technique is to find co-ordinates for each observation in a p-dimensional space such that the distances between observations measured in this space are monotonically related to the corresponding elements of $\underline{A}$. The algorithm for this technique requires a large amount of store since the matrix $\underline{A}$ has to be held, rather than the raw data, ($\underline{A}$ may or may not be symmetric). Four other arrays of the same size are also required for the calculations. A program is available for up to 60 observations if $\underline{A}$ is symmetric, but it

has not been utilised for any of the data sets studied here since they all have more than 100 observations.

A technique for displaying n-dimensional data in two dimensions is described by Fukunaga and Olsen (Fukunaga and Olsen, 1971). This involves defining a new pair of co-ordinates for each point, these are the square of the Euclidean distance of each point from two vectors in the original space. These vectors may be, for example, the mean of two groups. This technique is most suitable for two class problems and since it would therefore not be of sufficiently general use it has not been included in this system.

Waveforms can be used to illustrate n-dimensional data in several different ways. The simplest of these is where the variables are represented at regular intervals along the horizontal axis (Ball and Hall, 1970). Each curve, or in this case, piecewise line joining the different values for each variable, represents an individual or group of individuals. This type of picture becomes very complex as the number of curves increases, and in this instance the nature of the picture could depend on the ordering of the variables.

Andrews (Andrews, 1970) maps each data point, or group of data points onto a function

$$f(t) = x_1/\sqrt{2} + x_2 \sin t + x_3 \cos t + x_4 \sin 2t + x_5 \cos 2t + \ldots$$

where t is plotted in the range $-\pi$ to $+\pi$. Using functions such as these any number of variables can be displayed in a 2-dimensional picture. If a set of functions form a band for all values of t, then the points these functions represent lie near to one another in the Euclidean space. However it cannot always be said that because, for a particular value of t, two functions have the same value then any of the coefficients x_i are

necessarily similar. The example given uses for the x_i the coefficients of the canonical variates for 9 groups of data; these groups fall into two distinct sets. Other individuals are examined in relation to these groups.

4. PROGRAMS FOR MULTIVARIATE STATISTICS

The first section of this chapter explains why particular multivariate techniques have been included in this system. For each of those which are included there follows the theory, including some details of how the relevant program functions, and a description of the type of output which may be obtained.

4.1. The choice of multivariate statistical techniques

The types of statistical problem listed below are all suitable for interactive computing and interactive graphics. The techniques involved in solving these problems are particularly suited to user interaction in that they provide a means of finding, rapidly and directly, an acceptable solution. In each case effective and informative output has been designed to display results. These problems may involve the use of related techniques or they may involve repeated use of the same technique with variation in parameters. They have been divided into three categories, but there may well be overlap in that the techniques indicated for the solution of one type of problem may also be used in the solution of others.

The three categories of problem are as follows.

1) Problems which involve transforming the data to a new basis. Factor analysis is used to explain multiple response data in psychological terms. Principal components can be used to describe n-dimensional data in less than n dimensions. Factor and component scores displayed as histograms and scattergrams allow the user to view the data in terms of the new co-ordinate systems. Graphical displays have been designed to show the contribution of the original variables to each of the new axes. Also the differences between an initial solution and an orthogonal rotation of that solution can be effectively demonstrated.

2) Problems which require the classification of data items. This involves deciding whether or not a given sample of individuals falls into groups, and if it does so, deciding on the number of groups and the membership of each group. The initial methods used in the solution of this type of problem have here been called cluster analysis. With an interactive system different starting configurations can be tried and the resulting solutions compared. When the classification is complete a label may be assigned to each individual to indicate to which group it belongs. This label may be used for several different purposes including the isolation of a single group for further analysis. As a direct consequence of the use of interactive graphics it has been possible, with this system, to make further investigations into results obtained from cluster analysis.

3) Problems of discrimination. In this type of problem the investigator is presented with samples drawn from known populations, and the object is to determine how the populations are separated for predictive purposes.

As already indicated one technique may be used for different purposes. Principal components analysis can be used to define the data in terms of a new set of co-ordinates, either to study the principal components solution or for input to cluster analysis. The results of cluster analysis may be investigated using the calculations involved in discriminant analysis.

Sammon (Sammon, 1968) has described an interesting and versatile system for pattern analysis and pattern classification which, in part, resembles the system described here. Sammon shows how transformations to alternative bases for representation, such as the transformation to principal component vectors, can provide information about the structure of a pattern. Similar principles are described here in relation to multivariate analysis and these ideas have been developed to show how data structures can be analysed in depth within an interactive graphical environment.

Multiple regression has not been included in this system in the first instance, since the techniques involved would not mean the addition of any interesting computer graphics features. Multiple regression programs could be added and could, in principle, allow users the interactive choice of variables and a choice of 2-d diagrams.

4.2. The correlation matrix

A program has been included to compute means, standard deviations and a correlation matrix for a set of variables defined by the user at run time. These values may be filed for subsequent input to another program. The results may also be displayed in tabular form with hard copy if requested.

The means, standard deviations and correlation coefficients which are computed may be defined as follows. Given a set of n variables $x_1, \ldots x_n$ for which there are m observations and an mxn matrix with elements x_{ji} which represent the jth observation for the ith variable, then the sample mean of variable x_i is

$$\bar{x}_i = \frac{\sum_{j=1}^{m} x_{ji}}{m}$$

The sample variance of variable x_i is

$$s_i^2 = \frac{1}{m}\left(\sum_{j=1}^{m} (x_{ji} - \bar{x}_i)^2 \right)$$

and the square root of this quantity gives the standard deviation, s_i, for variable x_i. The sample correlation coefficient between variables x_i and x_k may be defined as

$$r_{ik} = \frac{\sum_{j=1}^{m} (x_{ji} - \bar{x}_i)(x_{jk} - \bar{x}_k)}{m s_i s_k} \qquad (4.1)$$

Since the data is read in from the c.i.f. one observation at a time, the algorithm given by Herraman (Herraman, 1968) is used to maintain a current mean, which is updated as each observation is read in, in order that an accurate value for s_i^2 may be obtained.

The correlation matrix is displayed a portion or 'page' at a time and keys may be used to display the next page or the one previously shown. The means and standard deviations are similarly displayed with keys being used for other pages if they cannot all be displayed at once.

4.3. Principal components and factor analysis

Although the models for principal components and factor analysis are different, these techniques have been incorporated into one program since a large proportion of the program code is common to both methods. This program, known as FACTOR, is described in Section 4.3.4, with reference to Figure 4.1 (page 85) where it can be seen that the facilities for principal components and factor analysis are symbolically represented as two separate routes through this program.

4.3.1. Principal components

Given the definitions for the sample means and variances of a set of n variables x_i in Section 4.2, we may further define the standardised form of x_i as z_i where the particular values for each of the j observations are

$$z_{ji} = (x_{ji} - \bar{x}_i)/s_i$$

Each z_i has zero mean, therefore

$$\sum_{j=1}^{m} z_{ji} = 0 \qquad i = 1, \ldots\ldots n$$

and also each z_i has unit variance, therefore

$$\frac{1}{m} \sum_{j=1}^{m} z_{ji}^2 = 1 \qquad i = 1, \ldots\ldots n$$

The correlation coefficient defined in (4.1) can be rewritten

$$r_{ik} = \sum_{j=1}^{m} z_{ji} z_{jk}/m \qquad (4.2)$$

The object of principal components is to express the information contained in the set of n variables z_i more economically. This may be done by finding a set of q new uncorrelated variables v_k, k = 1, ... q (q ⩽ n) where each v_k may be expressed in terms of the n variables z_i. Each new v_k accounts for a decreasing proportion of the variance and all the variance is accounted for in the q new variables. The situation where q < n will be rare and will occur only if the variables are linearly dependent. More usually q = n and an approximate description of the data may be obtained by examining only those new variables which account for a given proportion of the total variance.

Following Kendall (Kendall, 1968) Chapter 2, if the m observations are considered as m points in the n-dimensional space whose co-ordinates are given in terms of the standardised values z_{ji}, we find a line such that the sum of the squares of the perpendicular distances from each point to that line is a minimum. The set of equations of a line passing through the general point $(X_1, \ldots\ldots X_n)$ and the fixed point $(p_1, \ldots\ldots p_n)$, with direction cosines c_i, i=1,..n is

$$\frac{X_1 - p_1}{c_1} = \frac{X_2 - p_2}{c_2} = \ldots = \frac{X_n - p_n}{c_n} \qquad (4.3)$$

where $\sum_{i=1}^{n} c_i^2 = 1.$

If mS is sum of squared perpendicular distances then

$$mS = \sum_{j=1}^{m} \left[\sum_{i=1}^{n} (z_{ji} - p_i)^2 - \{ \sum_{i=1}^{n} c_i (z_{ji} - p_i) \}^2 \right]$$

To find a stationary value of mS we take partial derivatives with respect to each of the p_i's, set these to zero, hence the set of n equations

$$- \sum_{j=1}^{m} (z_{ji} - p_i) + \sum_{j=1}^{m} c_i \sum_{i=1}^{n} c_i (z_{ji} - p_i) = 0 \qquad i=1, \ldots n.$$

Since $\sum_{j=1}^{m} z_{ji} = 0$ these equations hold only if

$$\frac{p_i}{c_i} = \text{constant} \qquad \text{for each } i.$$

Therefore each of the p_i's in the equations (4.3) can be set to zero and

$$mS = \sum_{j=1}^{m} \left[\sum_{i=1}^{n} z_{ji}^2 - \left(\sum_{i=1}^{n} c_i z_{ji} \right)^2 \right]$$

$$= mn - \sum_{j=1}^{m} \left(\sum_{i=1}^{n} c_i z_{ji} \right)^2 \qquad (4.4)$$

since $\sum_{j=1}^{m} z_{ji}^2 = m.$

We find the stationary values of S, a function in n variables, c_i, subject to the condition $\sum_{i=1}^{n} c_i^2 = 1$, by introducing a Lagrange multiplier λ and taking the partial derivatives with respect to each of the c_i's. This gives the n equations

$$- \frac{1}{m} \sum_{j=1}^{m} z_{jk} \left(\sum_{i=1}^{n} c_i z_{ji} \right) + \lambda c_k = 0 \qquad k=1, \ldots n. \qquad (4.5)$$

These can be rewritten

$$c_1(1 - \lambda) + c_2 r_{12} + \ldots + c_n r_{1n} = 0$$

$$c_1 r_{21} + c_2(1 - \lambda) + \ldots + c_n r_{2n} = 0$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$c_1 r_{n1} + c_2 r_{n2} + \ldots + c_n(1 - \lambda) = 0$$

Eliminating the c_i's from these equations gives the determinantal equation

$$|\underline{R} - \lambda \underline{I}| = 0$$

where $\underline{R}$ is the correlation matrix with elements $r_{ik} = r_{ki}$. There are, in general n roots, $\lambda_1, \ldots \lambda_n$, of the resultant polynomial in λ, the characteristic roots or eigenvalues of $\underline{R}$.

If $\underline{R}$ is real symmetric then there exists an orthogonal matrix $\underline{C}$ such that

$$\underline{C}'\underline{R}\,\underline{C} = \underline{D}$$

where $\underline{D}$ is the diagonal matrix of the characteristic roots of $\underline{R}$, and $\underline{C}$ contains the characteristic vectors or eigenvectors of $\underline{R}$ normalised to unity. The columns of $\underline{C}$ are the orthogonal components $\underline{c}_k$, k=1, ... n, each associated with a particular λ_k, and here $\underline{C}$ is known as the components matrix; elements c_{ik}.

The variables v_k are given by

$$v_k = \sum_{i=1}^{n} c_{ik}\, z_i \qquad (4.6)$$

If we multiply through each of the n equations (4.5) by c_k and sum over k

$$\sum_{j=1}^{m} \Big(\sum_{i=1}^{n} c_i\, z_{ji} \Big)^2 = \lambda m \qquad (4.7)$$

and from (4.4) $S = n - \lambda$.

Therefore if S is a minimum, λ is a maximum and the largest root, say λ_1, gives the line with the minimum S. The variance of v_1 is $\frac{1}{m} \sum_{j=1}^{m} \Big(\sum_{i=1}^{n} c_{i1}\, z_{ji} \Big)^2$ and using equation (4.7) this is equal to the largest eigenvalue λ_1.

Principal component scores

To understand and to be able to view the original data in terms of the new variables, v_k, principal component scores must be computed. If v_{jk} is the score for observation j on component k

$$v_{jk} = \sum_{i=1}^{n} c_{ik} z_{jk} \qquad j=1, \ldots m, \; k=1, \ldots n.$$

These equations may be rewritten in matrix form

$$\underline{V} = \underline{Z}\,\underline{C} \tag{4.8}$$

where $\underline{V}$ is the mxn matrix of component scores and $\underline{Z}$ is the mxn matrix of standardised data.

4.3.2. Factor analysis

In principal components all the variance is taken into account in the evaluation of the components matrix, although in the final analysis only a proportion of this may be examined. In factor analysis it is assumed that the correlations between the n variables z_i represent a number of factors, less than n, and that each variable z_i is a linear combination of these common factors, plus a unique factor which accounts for the remaining variance of that variable. The object of factor analysis is to determine the number of factors, $p < n$, and the proportion of the variance accounted for by the unique factors.

If each of the variables, z_i, is to be expressed in this way and if there are p common factors f_k, k=1, ... p and a unique factor u_i for each variable then

$$z_i = \sum_{k=1}^{p} a_{ik} f_k + t_i u_i \qquad i=1, \ldots n$$

or for each observation

$$z_{ji} = \sum_{k=1}^{p} a_{ik} f_{jk} + t_i u_{ji} \qquad j=1, \ldots m, \; i=1, \ldots n \tag{4.9}$$

The unique factor accounts for variance due to the particular choice of variables or the specific factor s_i, plus the error or unreliability factor e_i. Therefore

$$z_i = \sum_{k=1}^{p} a_{ik} f_k + b_i s_i + d_i e_i \qquad i=1, \ldots n$$

where the f_k, k=1, ... p, s_i and e_i all have unit variance and are all independent of one another.

For each value of j

$$z_{ji} = \sum_{k=1}^{p} a_{ik} f_{jk} + b_i s_{ji} + d_i e_{ji} \qquad i=1, \ldots n, \; j=1, \ldots m.$$

The variance of z_i is

$$\sum_{j=1}^{m} z_{ji}^2/m = \sum_{j=1}^{m} \left(\sum_{k=1}^{p} a_{ik} f_{jk} + b_i s_{ji} + d_i e_{ji} \right)^2/m.$$

If we evaluate the square and sum over j, the cross-product terms vanish because of the independence of the f_k, k=1, ... p, s_i and e_i. Thus

$$\text{var}(z_i) = \sum_{k=1}^{p} a_{ik}^2 \left(\sum_{j=1}^{m} f_{jk}^2/m \right) + b_i^2 \sum_{j=1}^{m} s_{ji}^2/m + d_i^2 \sum_{j=1}^{m} e_{ji}^2/m.$$

Since f_k, s_i and e_i have unit variance

$$\text{var}(z_i) = \sum_{k=1}^{p} a_{ik}^2 + b_i^2 + d_i^2 = 1$$

since the variance of z_i is also unity.

$$\sum_{k=1}^{p} a_{ik}^2 = h_i^2$$

where h_i^2 is known as the communality of variable z_i.

The elements a_{ik}, i=1, ... n, k=1, ... p form an nxp matrix $\underline{A}$ which is known as the factor matrix.

In order to discuss how to determine p, the number of factors, and h_i^2 and hence the elements of the factor matrix, two results are required.

1) If the matrix with elements x_{ji} j=1, ... m, i=1, ... n is of rank p and the m points are represented in an n-dimensional space, they will all lie within a p-dimensional subspace, and the n variables can be described in terms of p vectors.

2) The rank of the product of a matrix by its transpose is equal to the rank of the matrix.

If the rank of $\underline{Z}$ is p then the m points lie in a p-dimensional subspace and the variables, z_i, can be described in terms of p vectors, also from the second result the rank of $\underline{R}$ is p. In the case of the principal components model the rank of $\underline{R}$ is usually n, therefore p=n, and n vectors are required to describe the n variables. For factor analysis p<n vectors or factors are required to describe the correlations between the n variables. Therefore a matrix $\underline{R}^1$ (sometimes called the reduced correlation matrix) of rank p is required, where $\underline{R}^1$ is the correlation matrix with the ones in the diagonal replaced by the communalities.

The number of conditions for a symmetric matrix of order n to be of rank p is $\frac{1}{2}(n-p)(n-p+1)$. These are independent conditions (Ledermann, 1937). For the matrix $\underline{R}^1$ there are n unknowns, h_i^2, i=1, ... n. The number of independent conditions the h_i^2 must satisfy in order that $\underline{R}^1$ has rank p is $\frac{1}{2}(n-p)(n-p+1)$. In general there will therefore only be solutions for h_i^2 if

$$n \geq \tfrac{1}{2}(n-p)(n-p+1)$$

If this inequality holds then for each n there is a minimum value of p for which the conditions can be satisfied. If this inequality does not hold

then there must exist relationships amongst the correlations in order that the number of independent conditions is reduced to the number of unknowns. If this case p may fall below the minimum value mentioned above.

One method of determining the communalities is to assume the rank of $\underline{R}^1$ and to compute the communalities using the conditions imposed on the correlationsin order that the matrix may have the assumed rank. This procedure becomes impractical when the rank is greater than two. In practice one way in which the communalities can be determined is firstly, to assume that the rank of $\underline{R}^1$ is p and that therefore there are p factors, then to insert estimates of the communalities in the diagonal of the correlation matrix. This matrix is diagonalised and new communality estimates are obtained from the first p of the resultant eigenvectors. These new estimates are then placed along the diagonal of the correlation matrix and the process is repeated until each of the communalities has converged to within a specified criterion. There is no formal proof that this process converges but there is evidence from tests that it does converge (Harman, 1960) (p.86). By this means we cannot say that the unique variance has been estimated; it has only been estimated given the number of factors. The factor matrix $\underline{A}$ with elements a_{ik}, i=1, ... n, k=1, ... p is the first p vectors of the matrix of eigenvectors resulting from the final diagonalisation in the iterative process described above. The communalities h_i^2 for variable z_i is the sum of squares of the ith row of this matrix, $\sum_{k=1}^{p} a_{ik}^2$.

In the program FACTOR, p may be specified by supplying an integer value or by using a criterion known as Kaiser's criterion (Kaiser, 1960); that p is the number of eigenvectors of the matrix $\underline{R}$ which have eigenvalues greater than one.

There are many procedures for choosing initial communality estimates of which three are included in this program. The first of these is to use

the squared multiple correlation of each variable with the remaining variables. The squared multiple correlation coefficient for variable i is $(1 - \frac{1}{r^{ii}})$ where r^{ii} is the ith diagonal element of the inverse of $\underline{R}$. Alternatively, the initial communality estimates may be taken from the eigenvectors of $\underline{R}$. If c_{ik} is the weight for the ith variable on the kth eigenvector, then the initial communality estimates will be $\sum_{k=1}^{p} c_{ik}^2$ for each i where p has already been chosen. Lastly, communality estimates may be supplied by the user if there is already some empirical evidence about communalities for a particular set of variables.

Factor scores

If $\underline{F}$ is the mxp matrix of factor scores, f_{jk}, $\underline{T}$ is a diagonal matrix with elements t_i and $\underline{U}$ is the mxn matrix of unique factor scores u_{ji}, equation (4.9) can be written

$$\underline{Z} = \underline{F}\,\underline{A}' + \underline{U}\,\underline{T}$$

It is not possible to solve this equation directly for $\underline{F}$ since a solution would involve $\underline{A}^{-1}$ which does not exist, moreover $\underline{U}$ and $\underline{T}$ are unknown. The elements of $\underline{F}$ therefore have to be estimated and conventional multiple regression procedures can be used for this. Each factor f_k is a linear function of the n original standardised variables and estimates of f_k, $\bar{f}_k$ may be obtained from the equation

$$\bar{f}_k = \beta_{k1} z_1 + \beta_{k2} z_2 + \ldots\ldots + \beta_{kn} z_n \qquad k=1, \ldots p \qquad (4.10)$$

The normal equations for evaluating the β coefficients are

$$\begin{aligned}
\beta_{k1} &+ r_{12}\beta_{k2} + \ldots + r_{1n}\beta_{kn} = a_{1k} \\
r_{21}\beta_{k1} &+ \beta_{k2} + \ldots + r_{2n}\beta_{kn} = a_{2k} \\
&\vdots \\
r_{n1}\beta_{k1} &+ r_{n2}\beta_{k2} + \ldots + \beta_{kn} = a_{nk}
\end{aligned} \qquad (4.11)$$

where r_{ij} is the correlation between variables z_i and z_j and a_{ik} is the correlation between variable z_i and factor f_k. These a_{ik} are the elements of the factor matrix $\underline{A}$ since an alternative description for $\underline{A}$ is that it gives the correlations between each of the variables and each of the factors. The equations (4.11) must be solved for β_{ki} i=1, ... n. If $\underline{\beta}_k$ is the column vector $\begin{pmatrix} \beta_{k1} \\ \cdot \\ \cdot \\ \cdot \\ \beta_{kn} \end{pmatrix}$ and $\underline{a}_k$ the column vector $\begin{pmatrix} a_{1k} \\ \cdot \\ \cdot \\ \cdot \\ a_{nk} \end{pmatrix}$ then $\underline{R}\ \underline{\beta}_k = \underline{a}_k$ and $\underline{\beta}_k = \underline{R}^{-1}\ \underline{a}_k$ or more generally for all k, k=1, ... p the solution is $\underline{R}^{-1}\ \underline{A}$. By substituting in (4.10) and rewriting this equation in matrix form, estimates of f_{jk} may be obtained from the equation

$$\underline{\bar{F}} = \underline{Z}\ \underline{R}^{-1}\ \underline{A} \qquad (4.12)$$

where $\underline{\bar{F}}$ is the matrix of estimates $\bar{f}_{jk}$.

4.3.3. Orthogonal rotations

Principal components analysis involves the definition of a new set of orthogonal axes, where the axes can be ordered in such a way that each successive axis accounts for a decreasing proportion of the variance. These new axes can be understood as a new set of variables or components. Once it has been decided to examine only a certain proportion of the total variance or only say, r, components and to ignore the remainder, the new n-dimensional space may be projected onto the r-dimensional subspace. The information contained in the last n-r dimensions will be ignored. The r components may then be rotated in order that they may be more easily interpreted. If each of the n original variables is considered as a point in the r-dimensional space, then finding the required rotated solution may be understood as rotating these r axes until as many of the points as possible have either

a high absolute loading or a near zero loading on each axis. The variables with high loadings on a particular axis or component are associated with that component and may be examined to see if they are related in terms of the problem under consideration.

A factor analysis solution may be regarded as a description of the original variables in terms of p new factors. There are an infinite number of ways of selecting factors for this description. The problem is to choose the axes to represent the correlations between the original variables in a way which has most meaning to the investigator. Each factor can be interpreted by examining those variables which have a high absolute loading on that factor, and the relationships between these variables. The methods by which these new axes are chosen involve rotating the initial factor solution until some criteria have been satisfied. The program FACTOR provides two methods for orthogonal rotations.

For these rotations the same theory applies to the principal components solutions and to the factor analysis solution. Therefore, in the discussion which follows, r is set equal to p and the n x p matrix $\underline{A}$ refers in the case of principal components to a matrix consisting of the first r(= p) columns of the components matrix and in the case of factor analysis $\underline{A}$ is the n x p factor matrix.

If $\underline{B}$ is an n x p matrix describing the original variables in terms of the rotated axes then there is a p x p transformation matrix $\underline{T}$ such that $\underline{B} = \underline{A}\,\underline{T}$.

If the initial principal components or factor solution is seen as points representing the variables in a p-dimensional space, then the new rotated axes should, where possible, lie close to any cluster of these points. The criteria for deciding on the elements of the matrix $\underline{B}$ can be put more precisely. (Thurstone, 1947) (Chap. 14)

1. Each row of the matrix $\underline{B}$ should have at least one zero.
2. Each column of the matrix $\underline{B}$ should have at least p zeros.
3. For every pair of columns of the matrix $\underline{B}$ there should be several variables whose entries vanish in one column but not in the other.
4. For every pair of columns $\underline{B}$ a large proportion of the variables should have vanishing entries in both columns where there are four or more factors.
5. For every pair of columns of $\underline{B}$ there should be only a small number of variables with non-vanishing entries.

One method of rotating the axes, which has been included here because of its particular suitability for use with a refresh tube, is to project the variables onto a plane defined by two of the original axes. These axes may then be rotated on the display by using the switches. When the rotation is satisfactory, this is a subjective judgement, the angle of rotation in the plane currently displayed will be recorded and the matrix $\underline{A}$ adjusted accordingly. The matrix $\underline{A}$ will by stages be transformed to the matrix $\underline{B}$. Two further axes may then be displayed and rotated. If, for example, there are originally three factors f_1, f_2, f_3 and initially f_2 and f_3 are rotated and the new axes in this plane are f_2' and f_3', the next axes to be displayed may be f_1 and f_2'. This pair will be rotated to f_1' and f_2''. The last set of axes to be rotated will then be f_1' and f_3'. When this process is complete the matrix stored is the matrix $\underline{B}$, the rotated factor matrix.

Such methods have been superceded by objective methods for rotating the axes to satisfy the criteria listed above.

The quantity $\sum_{k=1}^{p} a_{ik}^2$ is constant for each variable under the transformation $\underline{A}\,\underline{T} = \underline{B}$, (this is constant because $\underline{T}$ is an orthogonal matrix). Therefore, if the elements of the matrix $\underline{B}$ are b_{ik} for the ith variable on

the kth rotated component or factor,

$$\sum_{k=1}^{p} b_{ik}^2 = \sum_{k=1}^{p} a_{ik}^2 = h_i^2$$

Therefore $\left(\sum_{k=1}^{p} b_{ik}^2 \right)^2 = \left(h_i^2 \right)^2$ and summing over i

$$\sum_{i=1}^{n} \left(\sum_{k=1}^{p} b_{ik}^2 \right)^2 = \sum_{i=1}^{n} \left(h_i^2 \right)^2 = \text{constant}$$

$$\sum_{i=1}^{n} \left(\sum_{k=1}^{p} b_{ik}^2 \right)^2 = \sum_{i=1}^{n} \sum_{k=1}^{p} b_{ik}^4 + 2 \sum_{i=1}^{n} \sum_{k=1}^{p-1} \sum_{l=k+1}^{p} b_{ik}^2 b_{il}^2$$

Since the expression on the R.H.S. is constant, minimising the second term is equivalent to maximising the first and vice-versa. This relationship provides the basis for orthogonal rotations and if for instance we maximise the first term this will ensure that, for each variable, the loadings have either as large or as near zero values as possible.

One variation of this is to maximise the variance of the squared loadings for each factor, or to maximise the scatter of the squared loadings along each factor. Thus for each k, V_k should be maximised where

$$V_k = \left[n \sum_{i=1}^{n} \left(b_{ik}^2 \right)^2 - \left(\sum_{i=1}^{n} b_{ik}^2 \right)^2 \right] \Big/ n^2 \qquad k=1, \ldots p$$

Summing over k gives

$$V = \sum_{k=1}^{p} V_k = \sum_{k=1}^{p} \left[n \sum_{i=1}^{n} \left(b_{ik}^2 \right)^2 - \left(\sum_{i=1}^{n} b_{ik}^2 \right)^2 \right] \Big/ n^2 \qquad (4.13)$$

The rotation given as the result of the maximisation of this function is known as the varimax rotation (Kaiser, 1958) which is included here. The criterion in (4.13) is known as the raw varimax criterion and

each term will be weighted according to the square of the communality of the variable with which it is associated. In order to reverse this bias each term is divided by the relevant communality and the following function, the normal varimax criterion is maximised

$$V = \sum_{k=1}^{p} \left[n \sum_{i=1}^{n} \left(b_{ik}^2 / h_i^2 \right)^2 - \left(\sum_{i=1}^{n} b_{ik}^2 / h_i^2 \right)^2 \right] \Big/ n^2 \qquad (4.14)$$

Facilities are provided to maximise both (4.13) and (4.14). The method used is to maximise the function for one pair of axes at a time, and to repeat this procedure for all $\frac{1}{2}p(p-1)$ pairs, and then evaluate V. This constitutes one iteration. Iterations are continued until the difference in V between iterations is less than a specified criterion.

Rotated principal component scores

Equation (4.8) gives $\underline{V}$ the matrix of unrotated principal component scores. If $\underline{G}$ is the m x p matrix of rotated principal component scores, where p is the number of components rotated, then

$$\underline{G} = \underline{Z}\,\underline{B}$$

Rotated factor scores

Equation (4.12) gives $\underline{\bar{F}}$ the matrix of the estimates of the unrotated factor scores. If $\underline{\bar{G}}$ is the m x p matrix of estimated rotated factor scores then

$$\underline{\bar{G}} = \underline{Z}\,\underline{R}^{-1}\,\underline{B}$$

4.3.4. The program FACTOR

A schematic representation of how the program FACTOR is structured from the user's point of view is shown in Figure 4.1. The lists of options appear on the screen exactly as they are shown, the graphical and tabular displays are shown schematically. Not all the paths through which control

FACTOR

ENTRY

(1)

FACTOR ANALYSIS & PRINCIPAL COMPONENTS

OPTIONS AVAILABLE AT LEVEL 1

A) FACTOR ANALYSIS

B) PRINCIPAL COMPONENTS

Z) EXIT

EXIT

(2)

FACTOR ANALYSIS

OPTIONS AVAILABLE AT LEVEL 2

DEFAULT VALUES IN ()

A) INPUT FACTOR MATRIX - (CORRELATION MATRIX FROM CORR USED AS STARTING MATRIX)

B) INPUT USERS OWN CORRELATION MATRIX - (CORRELATION MATRIX OUTPUT FROM CORR IS USED)

C) SAVE FACTOR MATRIX - (MATRIX NOT SAVED)

D) DISPLAY VARIABLE NAMES FOR SELECTION - (ALL)

E) TYPE IN NO. OF FACTORS - (ALL)

F) KAISERS CRITERION USED TO DETERMINE NO. OF FACTORS - (ALL)

G) TYPE MINIMUM PERCENT OF VARIANCE TO BE EXPLAINED BY FACTOR STRUCTURE - (TEST WILL NOT BE PERFORMED)

H) COMMUNALITY ESTIMATES TAKEN FROM ORIGINAL EIGENVECTORS - (SQUARED MULTIPLE CORRELATIONS, RSQD, ARE USED AS COMMUNALITY ESTIMATES)

I) TYPE IN COMMUNALITY ESTIMATES - (RSQD)

J) MAXIMUM NO. OF ITERATIONS - (50)

K) CONVERGENCE CRITERION TO DETERMINE WHEN COMMUNALITIES HAVE CONVERGED - (.005)

L) COMMUNALITIES NOT PERMITTED TO FALL BELOW RSQD - (NO MINIMUM RESTRICTION)

X) START

Y) RETURN TO LEVEL 1

Z) EXIT

(3)

FACTOR ANALYSIS

CONVERGENCE OF COMMUNALITIES

KEYS: 1 PLOT - 2 CHANGE CONV. CRITERION - 3 CHANGE MAX. NO. ITERATIONS - 8 OPTIONS LEVEL 3

(4)

FACTOR ANALYSIS

OPTIONS AVAILABLE AT LEVEL 3

DEFAULT VALUES IN ()

A) DISPLAY EIGENVALUES

B) DISPLAY FACTOR WEIGHTS

C) DISPLAY VARIABLES IN FACTOR SPACE, TYPE IN 2 FACTOR NOS. FOR AXES

D) DISPLAY FACTORS WITH WEIGHTS AS AMPLITUDES, TYPE IN FACTOR NOS., MAX. 4

E) DISPLAY CUMULATIVE SUMS OF SQUARES OF FACTOR WEIGHTS

F) VARIMAX ROTATION

G) SCORES

H) GRAPHICAL ORTHOGONAL ROTATIONS, TYPE IN FACTOR NOS. - (1 & 2)

I) DISPLAY COMMUNALITIES

Y) RETURN TO LEVEL 2

Z) EXIT

(9)

PRINCIPAL COMPONENTS

OPTIONS AVAILABLE AT LEVEL 2

DEFAULT VALUES IN ()

A) INPUT COMPONENTS MATRIX

B) INPUT USERS OWN CORRELATION MATRIX - (CORRELATION MATRIX OUTPUT FROM CORR IS USED)

C) SAVE COMPONENTS MATRIX - (MATRIX NOT SAVED)

D) DISPLAY VARIABLE NAMES FOR SELECTION - (ALL)

X) START

Y) RETURN TO LEVEL 1

Z) EXIT

(10)

PRINCIPAL COMPONENTS

OPTIONS AVAILABLE AT LEVEL 3

DEFAULT VALUES IN ()

A) DISPLAY EIGENVALUES

B) DISPLAY COMPONENT WEIGHTS

C) DISPLAY VARIABLES IN EIGENVECTOR SPACE, TYPE IN 2 EIGENVECTOR NOS. FOR AXES

D) DISPLAY EIGENVECTORS WITH WEIGHTS AS AMPLITUDES, TYPE IN EIGENVECTOR NOS. MAX 4

E) DISPLAY CUMULATIVE SUMS OF SQUARES OF EIGENVECTOR WEIGHTS

F) VARIMAX ROTATION

G) SCORES

H) GRAPHICAL ORTHOGONAL ROTATIONS, TYPE IN EIGENVECTOR NOS. - (1 & 2)

Y) RETURN TO LEVEL 2

Z) EXIT

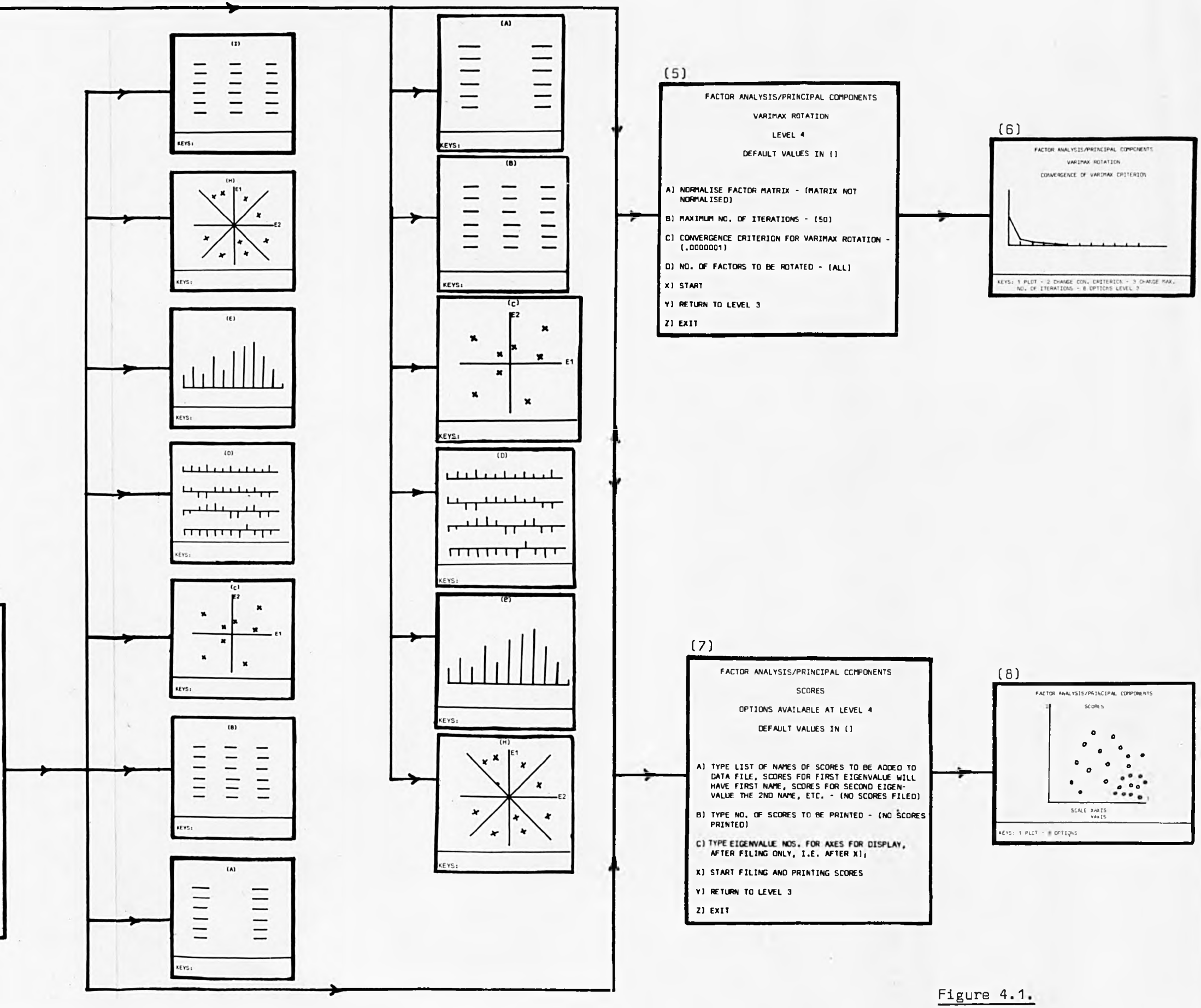

(I)
KEYS:
(H)
E1
E2
KEYS:
(E)
KEYS:
(D)
KEYS:
(C)
E2
E1
KEYS:
(B)
KEYS:
(A)
KEYS:
(A)
KEYS:
(B)
KEYS:
(C)
E2
E1
KEYS:
(D)
KEYS:
(E)
KEYS:
(H)
E1
E2
KEYS:
(5)
FACTOR ANALYSIS/PRINCIPAL COMPONENTS
VARIMAX ROTATION
LEVEL 4
DEFAULT VALUES IN ()
A) NORMALISE FACTOR MATRIX - (MATRIX NOT NORMALISED)
B) MAXIMUM NO. OF ITERATIONS - (50)
C) CONVERGENCE CRITERION FOR VARIMAX ROTATION - (.0000001)
D) NO. OF FACTORS TO BE ROTATED - (ALL)
X) START
Y) RETURN TO LEVEL 3
Z) EXIT
(6)
FACTOR ANALYSIS/PRINCIPAL COMPONENTS
VARIMAX ROTATION
CONVERGENCE OF VARIMAX CRITERION
KEYS: 1 PLOT - 2 CHANGE CON. CRITERION - 3 CHANGE MAX. NO. OF ITERATIONS - 8 OPTIONS LEVEL 3
(7)
FACTOR ANALYSIS/PRINCIPAL COMPONENTS
SCORES
OPTIONS AVAILABLE AT LEVEL 4
DEFAULT VALUES IN ()
A) TYPE LIST OF NAMES OF SCORES TO BE ADDED TO DATA FILE, SCORES FOR FIRST EIGENVALUE WILL HAVE FIRST NAME, SCORES FOR SECOND EIGENVALUE THE 2ND NAME, ETC. - (NO SCORES FILED)
B) TYPE NO. OF SCORES TO BE PRINTED - (NO SCORES PRINTED)
C) TYPE EIGENVALUE NOS. FOR AXES FOR DISPLAY, AFTER FILING ONLY, I.E. AFTER X);
X) START FILING AND PRINTING SCORES
Y) RETURN TO LEVEL 3
Z) EXIT
(8)
FACTOR ANALYSIS/PRINCIPAL COMPONENTS
SCORES
SCALE XAXIS
YAXIS
KEYS: 1 PLOT - 8 OPTIONS

Figure 4.1.

may pass are shown and an indication of, for instance, how the results of a varimax rotation may be examined is given in the description which follows. However, in this description the emphasis is on the different forms of graphical display and possible user interaction.

The user makes an initial choice of principal components or factor analysis at level 1 and then the relevant list of options appears. Input must be a correlation matrix either from the end of the c.i.f. or else from a device indicated by the user. For principal components analysis the user may set the parameters, initiate the analysis and when it is complete display the results. For factor analysis after the parameters are set and the analysis initiated, a graph, (3) in Figure 4.1, showing the convergence of the communalities is displayed. Each interval along the X-axis represents one iteration or one change in the current value of h_i^2 (each iteration involves the diagonalisation of the current n x n reduced correlation matrix). The Y-axis represents the squares of the differences in the communalities between successive iterations summed over all the variables. The keys will be interrogated after each iteration to enable the user to interrupt the iterative process in order to change the maximum number of iterations, the convergence criterion, or to stop the process and display the results. Once the communalities have converged or the maximum number of iterations has been reached or the iterations have been terminated interactively, results may be displayed. Both the components matrix and the factor matrix may be saved for future inspection, for rotations or for the evaluation of scores.

The different forms of output provided are largely the same for both techniques, although there is one which is only relevant for factor analysis. The description of each item is preceded by a definition of the command which must be given at level 3 in order to obtain the particular form of output.

A);

displays the eigenvalues.

For principal components these will be the eigenvalues of $\underline{R}$ and for factor analysis the eigenvalues of the final reduced correlation matrix $\underline{R}^1$.

B);

displays the first column of either the components matrix or the factor matrix. Subsequent and previous columns of the matrix can be displayed by pressing keys.

C)<integer 1>,<integer 2>;

displays the original variables as points projected on to the plane defined by the two eigenvectors <integer 1> and <integer 2>. <integer 1> and <integer 2> gives the eigenvector number for the X and Y-axes respectively. The co-ordinates of the points will represent the weights of the variables when the eigenvectors are normalised to unity, and initially the maximum and minimum on both axes will be ± 1.0. If all the points lie close to the origin and cannot be clearly distinguished the scale can be changed by pressing a key.

D)<integer 1>,<integer 2>,<integer 3>,<integer 4>;

displays a maximum of four components or factors with the variable weights or loadings shown as amplitudes. Sketches of this are shown in (D), below (4) and (10), in Figure 4.1. Vertical lines represent the amplitudes, +ve above the line, -ve below. For each vector the length of the lines depends on the maximum absolute value for all variables for that vector, which means that each vector is drawn to a different scale. To enable the user to identify individual variables, a cursor, a feint vertical line, may be displayed and moved backwards and forwards across the screen by pressing keys.

E);

is to display the cumulative sums of squares for each eigenvector. Initially

a horizontal line appears at the bottom of the screen. The first time key 2 is pressed vertical lines appear, one for each variable, representing the square of the weight of that variable on the first vector. The second time key 2 is pressed each vertical line represents the sum of the squares of the weights of each variable on the first two vectors. If c_{ik} are the elements of the components matrix, when the switch has been pressed n times, each of the lines will be of equal length and will fill the available space and will represent unity since $\sum_{k=1}^{n} c_{ik}^{2} = 1$. If a_{ik} are the elements of the factor matrix then when the switch has been pulled p times the vertical lines will represent the communalities since $\sum_{k=1}^{p} a_{ik}^{2} = h_i^2$.

F); and G); are discussed below.

H)<integer 1>,<integer 2>;

will display a picture similar to that obtained with the command C)<integer 1>, <integer 2>; but the user may now rotate the axes, using the switches, for an orthogonal rotation. The next pair of axes in a defined sequence of axes to be rotated, may be displayed by pressing a key.

I);

is relevant only to factor analysis and simply displays the numerical values of the communalities.

F);

is for varimax rotation and since parameters have to be set a further list of options is displayed, (5) in Figure 4.1. Once these have been set and the rotation initiated, a graph is displayed similar to that used to display the convergence of the communalities with similar interactive facilities. The Y-axis represents the difference in the varimax criterion between iterations. When the rotation is complete the list of options for level 3 is displayed and the commands starting B), C), D) and G) may be used to

display the results. This is also true for the results of the rotation obtained with the command H) etc.

G);

is for scores. Whether these will be for principal components or factor analysis and whether they will be rotated or unrotated scores depends on the most recent calculations. The list of options at level 4, (7) in Figure 4.1, is to allow the user to specify if the scores are to be filed on the c.i.f., and to specify axes to display the scores as points in 2-d scattergrams with any two variables on the c.i.f. as axes. All the points will be displayed with the same label; a small O. For more complex scattergrams and histograms the scores must be filed and input to the program DISVAR.

4.4. Cluster analysis

Given the usual set of n variables, x_1, ... x_n and m observations on each of these with values x_{ji}, the object of cluster analysis is to define clusters or groups of these observations where the members of each cluster closely resemble one another in terms of the data used to describe the observations. In this type of problem, sometimes called unsupervised classification, there are no preclassified samples and the problem is one of deciding how many clusters there should be and which observations belong to which clusters.

There are many methods or strategies for cluster analysis. Only a few of these are provided here, the object being not to provide a comprehensive set of clustering methods, but to demonstrate how a graphical display can be used to illustrate and compare the results of various methods. Four well known hierarchical methods are included. The form of input required for these methods, the methods themselves and the single program which implements them all are described first. Secondly, there is a description of one non-hierarchical method and the program to implement it.

4.4.1. Hierarchical clustering methods

For hierarchical clustering methods it is necessary to form an m x m matrix whose elements are measures of similarity, s_{jk}, or dissimilarity coefficients, d_{jk}(j,k=1, ... m). Similarity coefficients will usually have the value 1 if the individuals j and k are identical and 0 if they are complete opposites. Dissimilarity coefficients are 0 when j and k are identical and become larger as the differences between j and k increase. Many such coefficients have been defined and Sokal and Sneath (Sokal and Sneath, 1963) have made a comprehensive review of these coefficients. Only one is provided in this system; the distance coefficient

$$d_{jk} = \left[\sum_{i=1}^{n} (x_{ji} - x_{ki})^2 \right]^{\frac{1}{2}} \quad d_{jj} = 0, \; d_{jk} = d_{kj}$$

The matrix of these coefficients is calculated in a separate program, DISTANCE, which runs in a similar manner to the program for correlation coefficients, filing the matrix at the end of the c.i.f. It is a simple matter to provide programs to compute other similarity or dissimilarity coefficients and to output the matrix in a format acceptable to the hierarchical clustering program.

The first two of the hierarchical algorithms included here work with a matrix of dissimilarities or distances, and the second two with a matrix of similarities. However, initially either type of matrix may be input, if necessary, the following transformation is used

$$d_{jk} = \left[2(1 - s_{jk}) \right]^{\frac{1}{2}}$$

This particular transformation is chosen because, as Gower has shown (Gower, 1966), if the eigenvalues and eigenvectors of the matrix of similarity coefficients are calculated and each of the eigenvectors normalised to the relevant eigenvalue, then each individual can be represented as a point in the m-dimensional eigenvector space and the distance between any two of these points, say, p_j and p_k, is $\left[2(1 - s_{jk})\right]^{\frac{1}{2}}$.

"A hierarchic classificatory system may be considered as a nested sequence of partitions of a set of objects". (Jardine and Sibson, 1968). The output from a hierarchic classificatory system is a hierarchical dendrogram or tree diagram where each branch point of the tree is associated with a numerical value. Each branch point represents either the partitioning of a cluster or the amalgamation of clusters depending on whether the algorithm used is divisive or agglomerative. Divisive algorithms start with all m individuals in one group which is successively subdivided until ultimately there are m groups each with one individual. Agglomerative algorithms start with m individuals or groups and proceed by fusing these groups until ultimately there is only one group.

The algorithmsfor the four hierarchical methods included here are all agglomerative and start with m clusters each of one item. At each cycle two clusters denoted by C_j and C_k are joined to form a new cluster until after m-1 cycles there is only one cluster. The criterion for deciding which clusters should be joined and how the resultant cluster is defined in relation to the other clusters depends on the strategy being used.

For the first method, known as nearest neighbour or single-link clustering, dissimilarity coefficients are used and the two clusters which have the smallest inter-cluster distance are joined. For this method the inter-cluster distance is the smallest dissimilarity coefficient between items from each cluster. If the two clusters which are joined have m_j and m_k items respectively, then the new cluster will have $m_j + m_k$ items. The subroutine SLINK (Sibson, 1973) which is both compact and fast has been used to implement this method.

For the second method, known as furthest neighbour or complete link, dissimilarity coefficients are used and again the two clusters with the smallest inter-cluster distance are joined. In this case the inter-cluster distance is the largest dissimilarity coefficient between items from each cluster, and again the new cluster has $m_j + m_k$ items.

The third hierarchical technique is the weighted mean pair method described by Gower (Gower, 1967) and Sokal and Sneath (Sokal and Sneath, 1963). This is calculated using similarity coefficients. At any one cycle the two clusters joined will be those which have the highest absolute value of similarity coefficient. When two clusters C_j and C_k are merged a new cluster C_e is formed such that if the similarity coefficient between C_e and another cluster C_f is s_{ef} then

$$s_{ef} = \tfrac{1}{2} s_{fj} + \tfrac{1}{2} s_{fk} - \tfrac{1}{4} (1-s_{jk})$$

This is if the analysis is "weighted", the nomenclature used is the same as that of Gower (Gower, 1967) (p. 626-8). The point defining the new cluster is the midpoint of the line joining C_j and C_k. The fourth method is a variation of this.

If the analysis is "unweighted"

$$s_{ef} = \frac{m_j}{m_j + m_k} s_{fi} + \frac{m_k}{m_j + m_k} s_{fk} + \frac{m_j m_k}{(m_j + m_k)^2} (1 - s_{jk})$$

and the point defining C_e is the centroid of clusters C_j and C_k.

Jardine and Sibson (Jardine and Sibson, 1971) make the criticism that only the first of these techniques can be guaranteed to give a unique result for a given matrix of dissimilarities. Sibson (Sibson, 1971) has shown that elements of equal value in the dissimilarity matrix can give different results for all the methods implemented here except the first, if an arbitrary choice is made between the equal elements. A further major criticism is that all these methods other than the single-link are discontinuous. This means that a small change in the dissimilarity matrix, which occurs as the result of the fusion of two clusters, causes large changes in the coefficients associated with the new cluster and hence large changes in the resultant dendrogram. Therefore the effects of rounding errors and experimental errors can be magnified by these methods. Both these factors also mean that the results of different hierarchical clustering methods used on the same similarity or dissimilarity matrix may not be strictly comparable.

With the single-link method once the nucleus of a cluster is formed it tends to have items added to it one by one, rather than new clusters forming and large clusters joining together at a later stage. Jardine and Sibson (Jardine and Sibson, 1968) have developed a non-hierarchic method, which is a generalisation of the single-link method, and which overcomes this problem and allows for overlapping clusters. Implementation of an

algorithm for this method would have involved a discussion of the general topic of overlapping clusters, which was considered to be too large to be within the scope of this thesis.

The diagrammatic output from the hierarchical clustering program CLUSTER is firstly, a dendrogram. This may be displayed when the algorithm for a particular method is complete. It is drawn with the fusions made first at the top of the screen and those made last at the bottom. The numerical values associated with these fusions increase as one moves down the tree from the top. The light pen and tracking cross may be used to draw a horizontal line across the screen, thus defining g groups, where g is the number of vertical lines this horizontal line crosses. Each item then belongs to a group or cluster and therefore has a cluster label associated with it. A scattergram may then be displayed with each point represented by its cluster label. The axes for the scattergram may be any pair of variables on the c.i.f. including, for instance, the NLM co-ordinates. The cluster labels may be filed on the c.i.f., which will mean the addition of one variable to that file. For each observation this variable will have an integer value k, where $1 \leqslant k \leqslant g$, indicating to which cluster each individual belongs.

4.4.2. Non-hierarchical clustering

The major drawback with hierarchical clustering methods is that once two clusters have been joined they cannot be separated and a decision made early in the analysis may not prove to be a correct one in the latter stages.

The one non-hierarchical method included here is a method which Beale (Beale, 1969) calls "Euclidean Cluster Analysis". This method does not involve using a matrix of inter-element similarities or dissimilarities but the original data values x_{ji}. The object is to divide the data set into g disjoint clusters $C_1, \ldots C_g$, with $m_1, \ldots m_g$ items respectively, such that the sum of squared deviations of all observations from their cluster centres

is a minimum. If the cluster centres are defined as the means of the clusters and if $\bar{x}_{k_i}$ denotes the mean of the ith variable for cluster k then

$$\bar{x}_{k_i} = \frac{1}{m_k} \sum_{j \epsilon C_k} x_{ji},$$

and the sum of the squared deviation of each point from its cluster centre for all g clusters is

$$S = \sum_{k=1}^{g} \left[\sum_{j \epsilon C_k} \sum_{i=1}^{n} (x_{ji} - \bar{x}_{k_i})^2 \right]$$

The object of this method is to attempt to find the set of clusters for which S is a minimum.

The basic idea of minimising S is used by several authors including MacQueen (MacQueen, 1967) and Ball and Hall in ISODATA (Ball and Hall, 1965). Different procedures are used for choosing initial cluster centres and MacQueen and Ball and Hall allow the number of clusters, g, to vary according to some user defined criteria. Clusters can be split as they become too large and fused together if they are too small.

Beale's algorithm, which is used here, is to start with g higher than will eventually be required and to find a minimum for the sum of squared deviations of observations from their cluster centres as follows. Initial cluster centres are provided (a description of the options provided for the choice of initial cluster centres in the program which implements this algorithm is deferred until the description of the program), and each observation is allocated to its nearest cluster centre. Each observation is examined in turn and moved to another cluster if the total sum of squared deviations is reduced by doing so. If d_k is the distance of an observation from the centre of cluster k and d_l its distance from the centre of cluster l, the criterion for moving a point is not simply if

$$d_1^2 < d_k^2$$

An observation is reassigned if the squared distance from the centre of cluster 1 is less then the squared distance from the centre of cluster k even when the clusters are simultaneously repositioned or if

$$\frac{m_1}{m_1 + 1} d_1^2 < \frac{m_k}{m_k - 1} d_k^2$$

A minimum, although it may only be a local minimum, is achieved when no further improvement can be made in the total sum of squared deviations by moving a single observation.

Two clusters can then be amalgamated by finding the pair of clusters which when joined cause the smallest increase in the sums of squares. The centre of this cluster and the other cluster centres can be used as a set of initial clusters for a further solution with the number of clusters reduced by one. This process can be repeated until the required number of clusters is achieved.

Beale suggests as a significance test to determine the optimum number of clusters, a modified F-statistic to determine whether a division into g_1 clusters is significantly better than a division into g_2 clusters where $g_1 < g_2$. If S_{g_1} denotes the residual sum of squares when m observations are partitioned into g_1 clusters then the statistic

$$F(g_1, g_2) = \frac{S_{g_1} - S_{g_2}}{S_{g_2}} \Bigg/ \left[\left(\frac{m - g_1}{m - g_2} \right) \left(\frac{g_2}{g_1} \right)^{2/n} - 1 \right]$$

can be computed and treated as an F statistic with $n(g_1 - g_2)$ and $n(m - g_1)$ degrees of freedom for all $g_1 < g_2 \leq g_{max}$. If for a given g_1 it is significant for any g_2 then the configuration of g_1 clusters is not entirely adequate.

The program EUCLID

The choice of initial cluster centres.

Three methods are provided for making a choice of initial cluster centres. If g_{max} is the initial number of clusters, firstly the co-ordinates for the first g_{max} observations may be used. Secondly, random starting positions may be used, by requesting the use of a pseudo-random number generator with an option command. Facilities are provided to ensure different random starting configurations if these are required. The third method involves the interactive use of the graphical display. The data may be presented as a scattergram using any two variables on the c.i.f. as axes; the most relevant in this case would be a pair which in some way represent information about all the n dimensions which are to be used for the cluster analysis, for instance, principal component scores or NLM co-ordinates. The light pen and tracking cross may then be used to define and draw g_{max} circles, by specifying the centre and radius of each circle. The co-ordinates of the mean of all the points which lie within each circle define the initial cluster centres.

The algorithm for Euclidean Cluster Analysis and representation of results.

The algorithm given in Applied Statistics Algorithms (Sparks, 1973) is used to attempt to minimise S. First, points are allocated to their nearest initial cluster centre, the means of these clusters are calculated and these represent the new cluster centres. Scattergrams may be used to present the results where each data point is represented by its current cluster label. These labels will only be changed when the value of g is altered. At this stage the axes for the scattergram may be changed to present the data in terms of different variables, and clusters may be viewed individually again in terms of a scattergram. In this case the data points are represented by their record identifiers. The modified F-statistic can also be displayed and

printed. The values of the total squared deviation for each value of g are recorded and these may be presented as a graph with the values of g plotted against the total squared deviation. Each time a solution is found for a given value of g, it is filed, and the user may rapidly redisplay earlier solutions.

4.5. Discriminant analysis

Given a set of measurements on individuals drawn from k populations, we may wish to determine, with the minimum possibility of error, to which of these populations a newly observed individual belongs. A related form of analysis, canonical analysis, can be used to demonstrate the relative positions of the population means.

This section starts with a definition of the discriminant scores by means of which an individual can be assigned to one of the k populations. This is followed by a definition of the canonical variates. Finally there is an outline of the program which implements these techniques.

4.5.1. Discriminant scores

If there are k populations and n measurements, $x_1, \ldots x_n$, on each member of each population, then we require a division of the n-dimensional space into k mutually exclusive regions which are such that the probability of wrongly classifying an individual is a minimum.

If the probability density functions for each of the populations are $f_1, \ldots f_k$, the k regions which minimise the chances of wrongly classifying an individual are such that $f_1 \geqslant f_2, \ldots, f_1 \geqslant f_k$ and similarly for each of $f_2, \ldots f_k$, (Rao, 1952).

If the k populations are multivariate normal with the same dispersion matrix with elements α_{ij} then the probability density function for the rth group is

$$f_r = C \exp - \tfrac{1}{2}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha^{ij}(x_i - \mu_{ri})(x_j - \mu_{rj})\right]$$

where the α^{ij} are the elements of the inverse of the dispersion matrix and μ_{ri} is the mean of the ith variable for the rth population. The surface of

constant likelihood ratio between populations r and s is defined by

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha^{ij}(\mu_{ri} - \mu_{si})x_i - \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha^{ij}(\mu_{ri}\,\mu_{rj} - \mu_{si}\,\mu_{sj}) = \text{const} \qquad (4.15)$$

If we consider the k functions, or discriminant scores

$$L_r = \sum_{i=1}^{n}\Big(\sum_{j=1}^{n} \alpha^{ij}\,\mu_{ri}\Big)x_i - \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha^{ij}\,\mu_{ri}\,\mu_{rj}$$

then an observation is allotted to that population for which L_r is a maximum. For if, say L_s is the maximum score then from (4.15) $f_s > f_t$ for $t = 1, \ldots k$, $t \neq s$. If the probabilities of occurrence of the k populations are $\pi_1, \ldots \pi_k$ where $\sum_{r=1}^{k} \pi_r = 1$, then an individual is assigned to the population for which

$$L_r + \log_e \pi_r$$

is a maximum.

The scores L_r may be calculated by inserting the sample values;

$$\sum_{i=1}^{n}\Big(\sum_{j=1}^{n} v^{ij}\,\bar{x}_{ri}\Big)x_i - \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} v^{ij}\,\bar{x}_{ri}\,\bar{x}_{rj}$$

where v^{ij} are the elements of the inverse of the average within group dispersion matrix and $\bar{x}_{ri}$ is the sample mean for the ith variable in the rth group.

4.5.2. Canonical variates

If there are only two groups or if the means all lie on a straight line, then the line through the means can be used to measure the discriminant function. If the means do not lie on a straight line, the discriminant function can be measured along the line which is such that the sum of

squared perpendicular distances from the means to that line is a minimum. The equation for this line, which may be written $\sum_{i=1}^{n} l_i x_i$, is found by maximising the ratio of the between group variance to the within group variance or solving the equations

$$(\underline{B} - \lambda \underline{W})\underline{l} = 0$$

where $\underline{B}$ is the between group dispersion matrix and $\underline{W}$ the pooled within group dispersion matrix. λ is a root of the determinantal equation

$$|\underline{B} - \lambda \underline{W}| = 0 \qquad (4.16)$$

The vector $\underline{l}$ associated with the largest λ defines the required line which is the first canonical variate. A second line, orthogonal to the first, associated with the second largest root of (4.16) defines the second canonical variate. The number of canonical variates is equal to the minimum of n and k-1. Therefore if there are two groups there will only be one variate; the line joining the means.

Mahalanobis' distance, D^2, between two populations r and s as estimated from the sample is

$$\sum_{i=1}^{n} \sum_{j=1}^{n} v^{ij}(\bar{x}_{ri} - \bar{x}_{si})(\bar{x}_{rj} - \bar{x}_{sj})$$

Rao (Rao, 1952) derives the canonical variates by finding, for the first variate, the line with respect to which the sum of all $\frac{1}{2}k(k-1)$ distances is a maximum, and for the first two variates the first two mutually orthogonal lines for which this sum is a maximum. The sum of all the eigenvalues which are solutions to (4.16) is the sum of all possible D^2. If the means are represented with, say, the first two of these variates as co-ordinate axes, then the ratio of the sum of the first two eigenvalues to the sum of all

the eigenvalues is a measure of the adequacy of the picture in representing all the distances between the populations.

4.5.3. A program for discriminant analysis

The program for discriminant analysis allows the user to define k sample populations or groups of observations from the current input file, either by supplying lists of record identifiers or by defining groups with logical IF statements.

Once the groups have been defined, the weights and constants for evaluating discriminant scores are calculated, provided that the inverse of the within group dispersion matrix exists, and these may be displayed on the screen. A contingency table may also be displayed showing how the data used to evaluate the discriminant functions was allocated using these same discriminant functions. Additional sets of n measurements for further observations may be typed in and will be assigned to groups using the discriminant scores.

A scattergram may be displayed with the group means referred to any two canonical variates. The coefficients defining each of the variates are normalised so that the average within group variance along each variate is unity. Data points may then be superimposed on the diagram by pressing a key; they will be displayed with a group identifier. The scores for each of the data items on the canonical variates may be filed as additional variables on the c.i.f. If this data is then input to the program DISVAR, the canonical variates can be displayed individually as histograms (see Figure 6.22, page 162). The same groups can be defined for this program as were defined for the discriminant analysis in order to be able to inspect the distribution of the groups along each canonical variate. This is particularly relevant in instances where there are only two groups and therefore only one canonical variate.

5. THE PROGRAM FACTOR IN PRACTICE

Three sets of data have been used to demonstrate the program FACTOR; two for principal components and one set for factor analysis.

5.1. 24 psychological tests

The first set for principal components is a well known set of data consisting of 24 psychological tests performed on children and collected by Holzinger and Swineford. A factor analysis model, which would include error factors is, strictly, a more suitable model for the analysis of this data. However, in spite of this, clear results have been obtained by using this program for principal components analysis and the results are particularly useful for illustrating the value of graphical output.

The data was taken from Harman (Harman, 1960) (pp. 124-5) who gives a definition of each of the variables and also the intercorrelations between them. The variables fall into five psychologically distinct sets; spatial tests (tests 1 - 4), verbal tests (tests 5 - 9), speed tests (tests 10 - 13), memory tests (tests 14 - 18) and tests of mathematical ability (tests 19 - 24). These sets of variables emerge clearly in the diagrams displayed. This clarity is partly helped by the fact that the variables are ordered into the five sets in the correlation matrix and variables in the same set appear next to one another.

The eigenvalues from the principal components analysis are given in Table 5.1. Eigenvectors are displayed graphically with the weights of variables shown as amplitudes. Amplitude diagrams for the eight eigenvectors associated with the eight largest eigenvalues are shown in Figures 5.1 and 5.2.

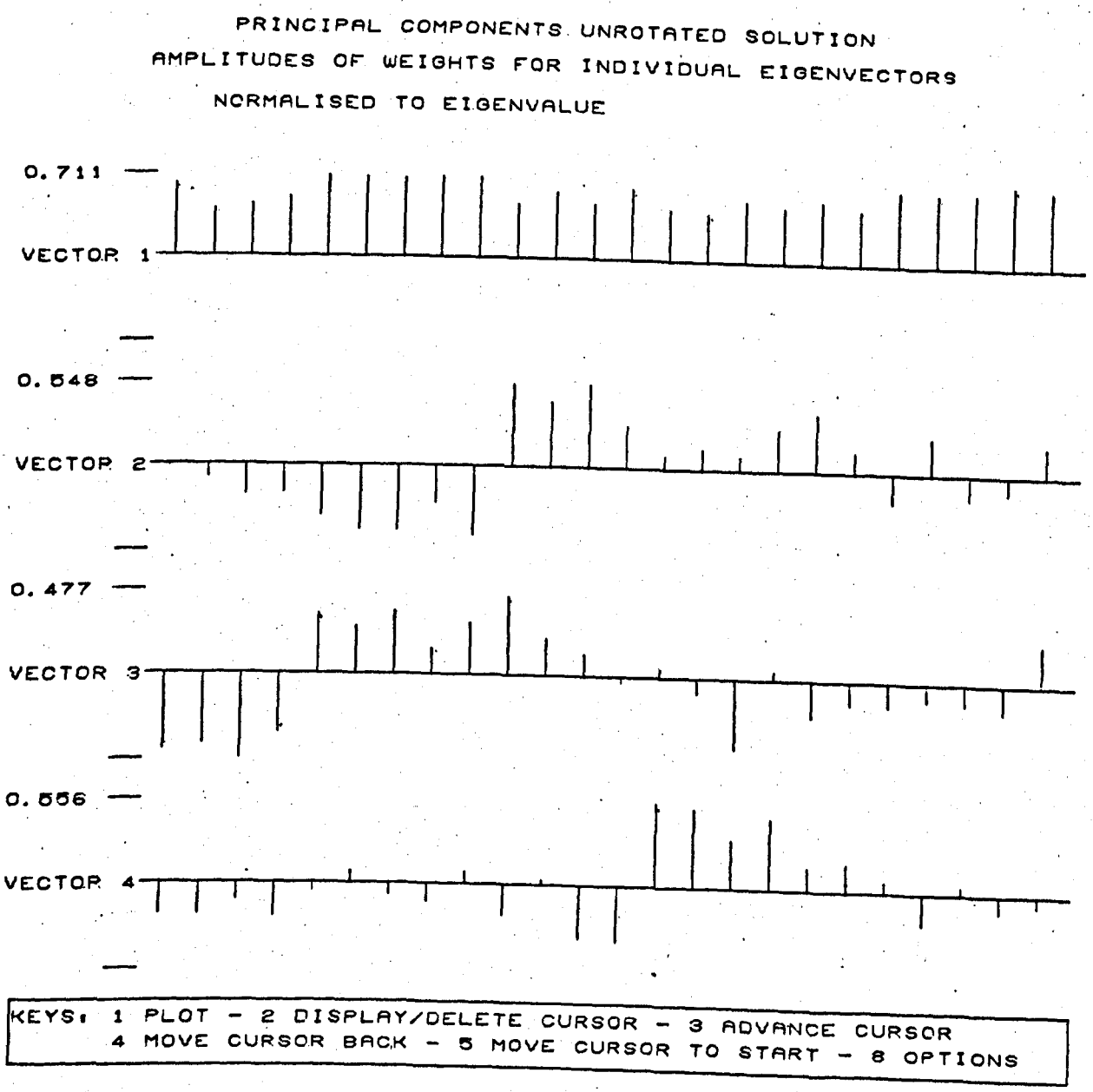


Figure 5.1. 24 psychological tests: unrotated components 1 - 4

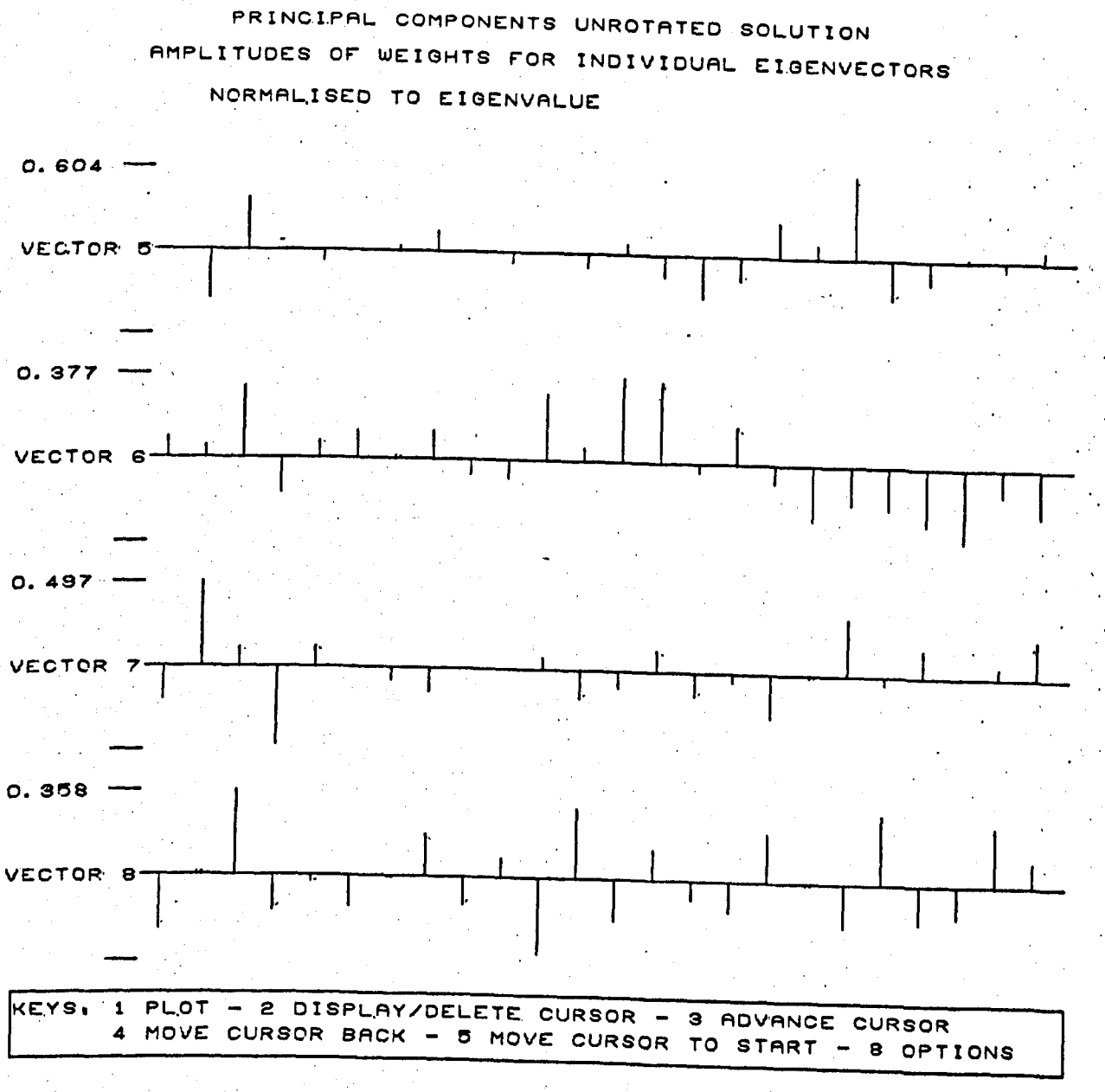


Figure 5.2. 24 psychological tests: unrotated components 5 - 8

	Eigenvalues	%	Cum. %		Eigenvalues	%	Cum. %
1	8.13957	33.9	33.9	13	0.52891	2.2	84.7
2	2.09782	8.7	42.6	14	0.50209	2.1	86.8
3	1.69156	7.0	49.6	15	0.47677	2.0	88.8
4	1.50254	6.3	55.9	16	0.39441	1.6	90.4
5	1.02400	4.3	60.2	17	0.38416	1.6	92.0
6	0.94677	3.9	64.1	18	0.33731	1.4	93.4
7	0.90450	3.8	67.9	19	0.33327	1.4	94.8
8	0.81410	3.4	71.3	20	0.31663	1.3	96.1
9	0.79537	3.3	74.6	21	0.29794	1.2	97.3
10	0.70693	2.9	77.5	22	0.26346	1.1	98.4
11	0.63662	2.7	80.2	23	0.18968	0.8	99.2
12	0.54366	2.3	82.5	24	0.17194	0.7	99.9

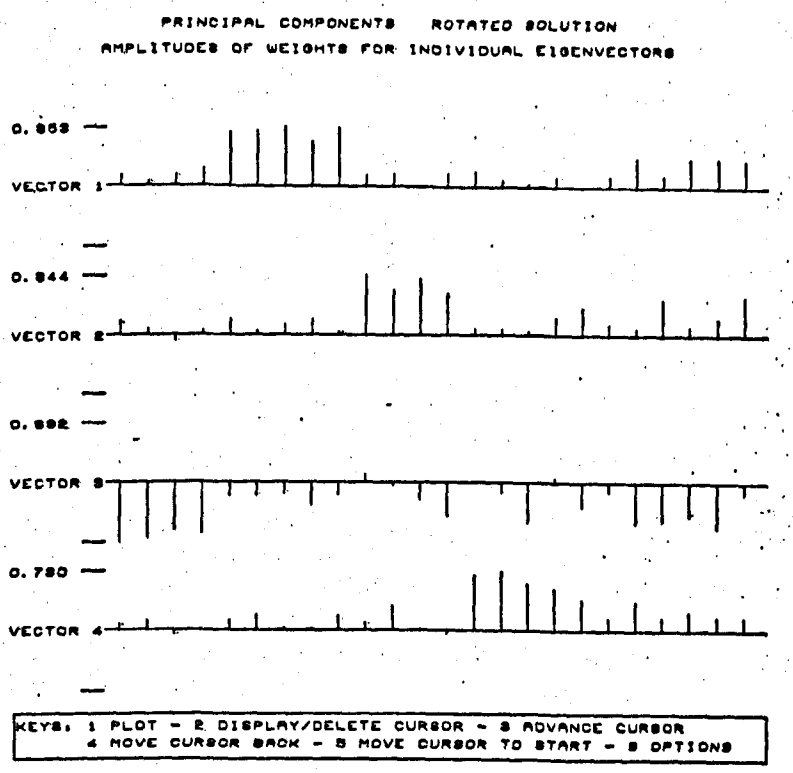
Table 5.1.

Eigenvalues for 24 psychological tests

Some grouping of variables can be observed within the vectors shown in these two figures. For vector 3 the first four tests have relatively large amplitudes but so also do other tests on this vector. Also in vector 4 the first four members of the fourth set of tests have relatively large amplitudes, but the fifth member of this set has a small amplitude when compared with other tests in this vector.

Although the varimax rotation was developed primarily for rotating factor analytic solutions, it can be used to advantage to rotate principal components solutions. The results of this rotation presented graphically illustrate how effectively rotation clarifies the partitioning of the variables into different sets. This technique is also particularly appropriate within the context of interactive graphics, since the number of components or axes to be included in the varimax rotation can be varied and the different solutions quickly and efficiently examined.

If the first five eigenvectors are rotated using the normalised varimax criterion, groups emerge clearly on individual vectors. (See Figure 5.3 for these rotated vectors, vectors 2, 3 and 4 are repeated in



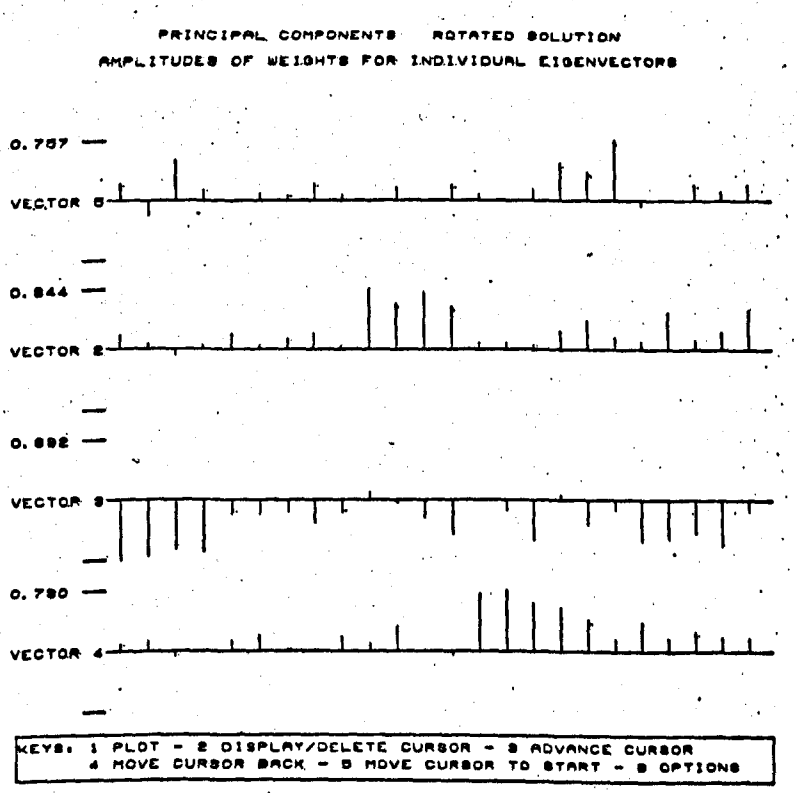
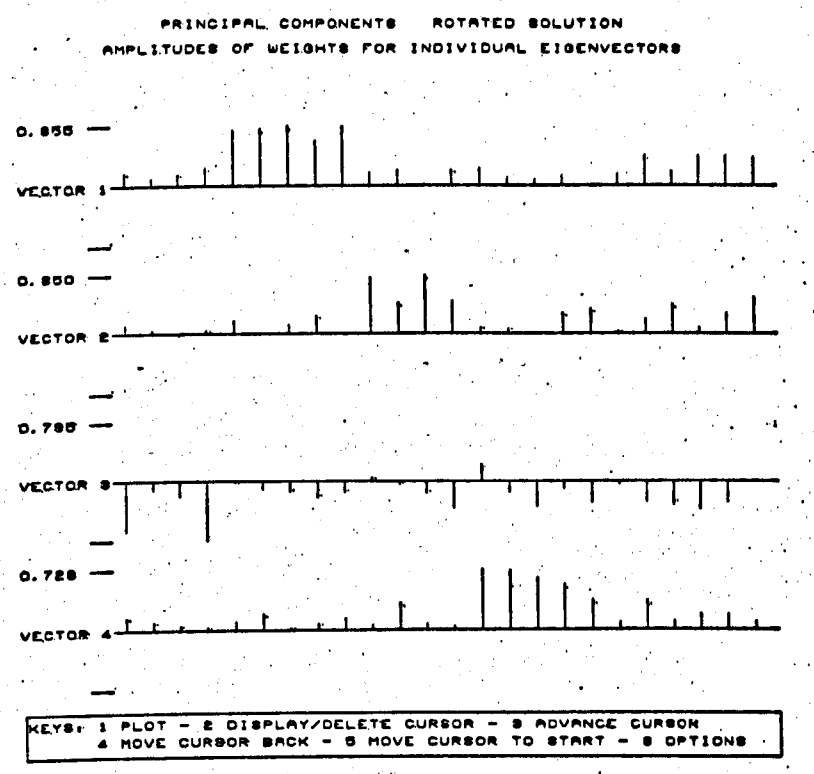


Figure 5.3. 24 psychological tests:
5 component rotation

the lower picture). In vector 1 group 2 (tests 5 - 9) dominates, in vector 2 group 3 (tests 10 - 13), in vector 3 group 1 (tests 1 - 4) and in vector 4 group 4 (tests 14 - 18). Group 5 is not associated with any one particular vector, but the variables of this set have moderately large amplitudes for several different vectors. Therefore this particular rotation associates four sets of variables each with a separate vector, and these sets are the sets one would expect to find from the definition of the tests. This is with the exception of the last set which is not associated with any particular vector.

If now the first eight vectors are rotated, again using the normalised varimax criterion, (Figure 5.4 shows the amplitude diagrams for this rotation), the fifth group still does not emerge. Also in this rotation the first group of tests (tests 1 - 4) is split between vectors 3, 7 and 8. As further vectors are included the groups gradually split up until finally, when 24 vectors are included only one test is associated with each vector. It is therefore of some importance to be able to examine the properties of the process until an acceptable and meaningful form of resolution has been obtained. Interactive graphics, with suitably designed and rapidly produced displays, proved useful for the examination of these properties.



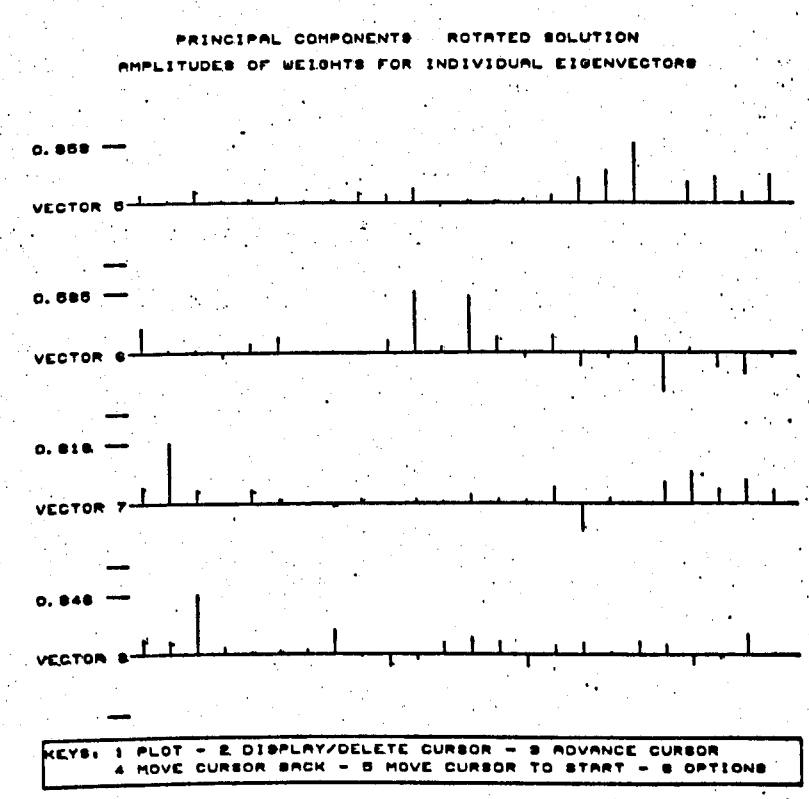


Figure 5.4. 24 psychological tests:
8 component rotation

5.2. Children's palate study

The second set of data used to demonstrate the use of this program for principal components is a set of measurements associated with a study of children's palates which is part of a project related to cleft palate surgery. The data under consideration refers to normal palates and was used as control data in the study. A large number of geometric variables defining the structure of the palate were specified by the originators of the problem, and Figure 5.5 is a diagram of a typical normal palate indicating the points used to identify these variables. Some of the variables are related to others and some appear to be redundant. Nevertheless, the problem was taken as defined and a correlation matrix for all 53 variables was constructed, and this matrix with unities in the diagonal was diagonalised as the first step of a principal components analysis. The resultant eigenvalues are given in Table 5.2. Seven eigenvalues greater than unity were obtained, however there

	Eigenvalues	%	Cum. %		Eigenvalues	%	Cum. %
1	24.81480	46.8	46.8	19	0.00708	0.0	100.0
2	9.81423	18.5	65.3	20	0.00509	0.0	100.0
3	7.29815	13.8	79.1	21	0.00322	0.0	100.0
4	3.44494	6.5	85.6	22	0.00232	0.0	100.0
5	2.12405	4.0	89.6	23	0.00152	0.0	100.0
6	1.63565	3.1	92.7	24	0.00130	0.0	100.0
7	1.15516	2.2	94.9	25	0.00092	0.0	100.0
8	0.98673	1.9	96.8	26	0.00060	0.0	100.0
9	0.65292	1.2	98.0	27	0.00034	0.0	100.0
10	0.40093	0.8	98.8	28	0.00029	0.0	100.0
11	0.33490	0.6	99.4	29	0.00021	0.0	100.0
12	0.12415	0.2	99.6	30	0.00013	0.0	100.0
13	0.07489	0.1	99.7	31	0.00010	0.0	100.0
14	0.04730	0.1	99.8	32	0.00009	0.0	100.0
15	0.03068	0.1	99.9	33	0.00005	0.0	100.0
16	0.02132	0.0	100.0	34	0.00003	0.0	100.0
17	0.00855	0.0	100.0	35	0.00001	0.0	100.0
18	0.00735	0.0	100.0	36-53	0.00000	0.0	100.0

Table 5.2.

Eigenvalues for normal palate data

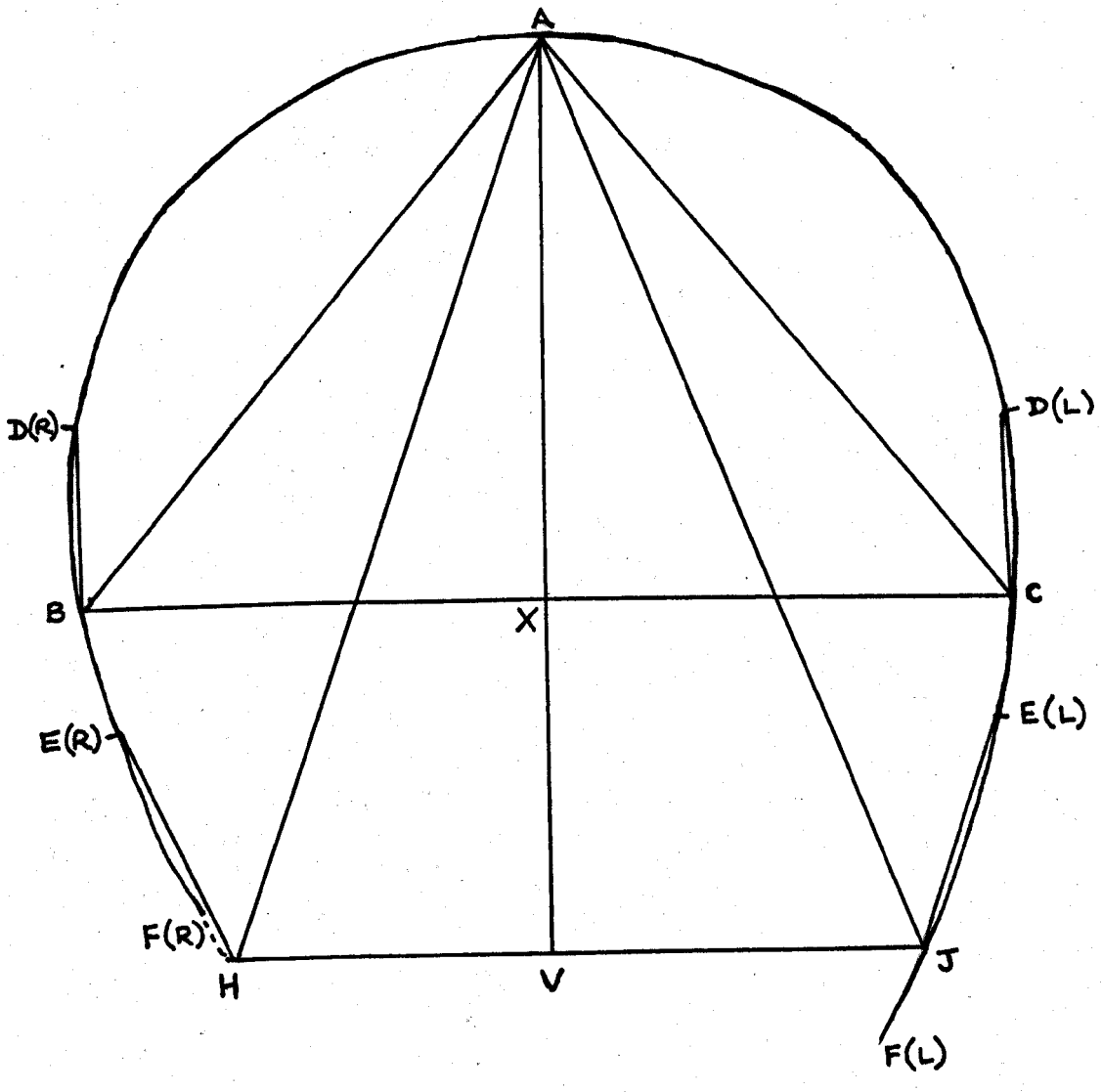


Figure 5.5. Diagram of a normal palate

is a greater difference between the 8th and 9th eigenvalues than between the 7th and 8th, and the analysis was therefore developed by considering rotations for both 7 and 8 components using the normalised varimax criterion. There were not many differences between the results of the 7 and 8 component rotations, and amplitude diagrams for the first eight vectors resulting from the 8 component rotation are given in Figures 5.6 and 5.7. (A cursor, which can be moved interactively with switches, to identify individual variables is shown in Figure 5.6. The variable currently pointed to is identified in the top right hand corner).

Two sets of measurements were given for each palate, one set with respect to a base line BC, in Figure 5.5, whose mid-point X is 20mm below A with AX (the midline plane) perpendicular to BC, and the second set with respect to the line HJ. If F(R) and F(L) are, respectively, the right and left 'end' points of the crest of the alveolar ridge, then the line HJ is a transverse line equidistant from F(R) and F(L) perpendicular to the midline plane. E(R) and E(L) are points on the crest of the alveolar ridge 15mm anterior to H and J respectively. Similarly D(R) and D(L) are points 15mm anterior to B and C respectively.

The eight rotated components and their associated variables are each briefly discussed below. (The cursor in Figure 5.6 is currently pointing to a variable which has only small amplitudes on vectors 1 - 4).

In all the rotated solutions, as well as the unrotated solution, the first component has many variables with relatively large amplitudes, therefore it is impossible to come to any conclusions about this component.

For the second component V9, V10, V23 and V24 have large amplitudes. V9 and V10 measure the angles D(R)BX and D(L)CX and V23 and V24 are derived from these. V23 is the mean of V9 and V10 and V24 is the sum of V9 and V10.

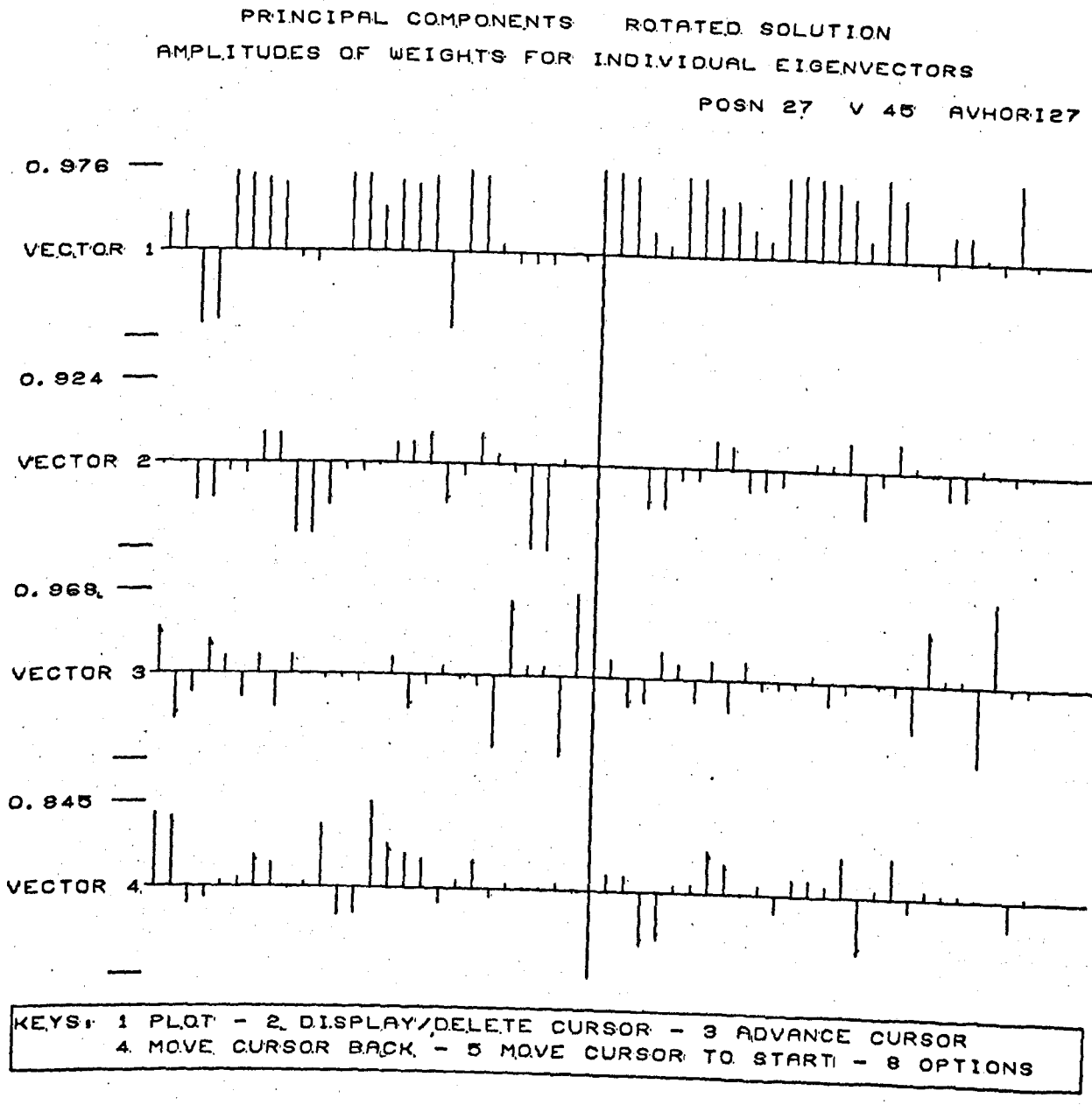


Figure 5.6. Childrens' palate study: rotated components 1 - 4 (with cursor)

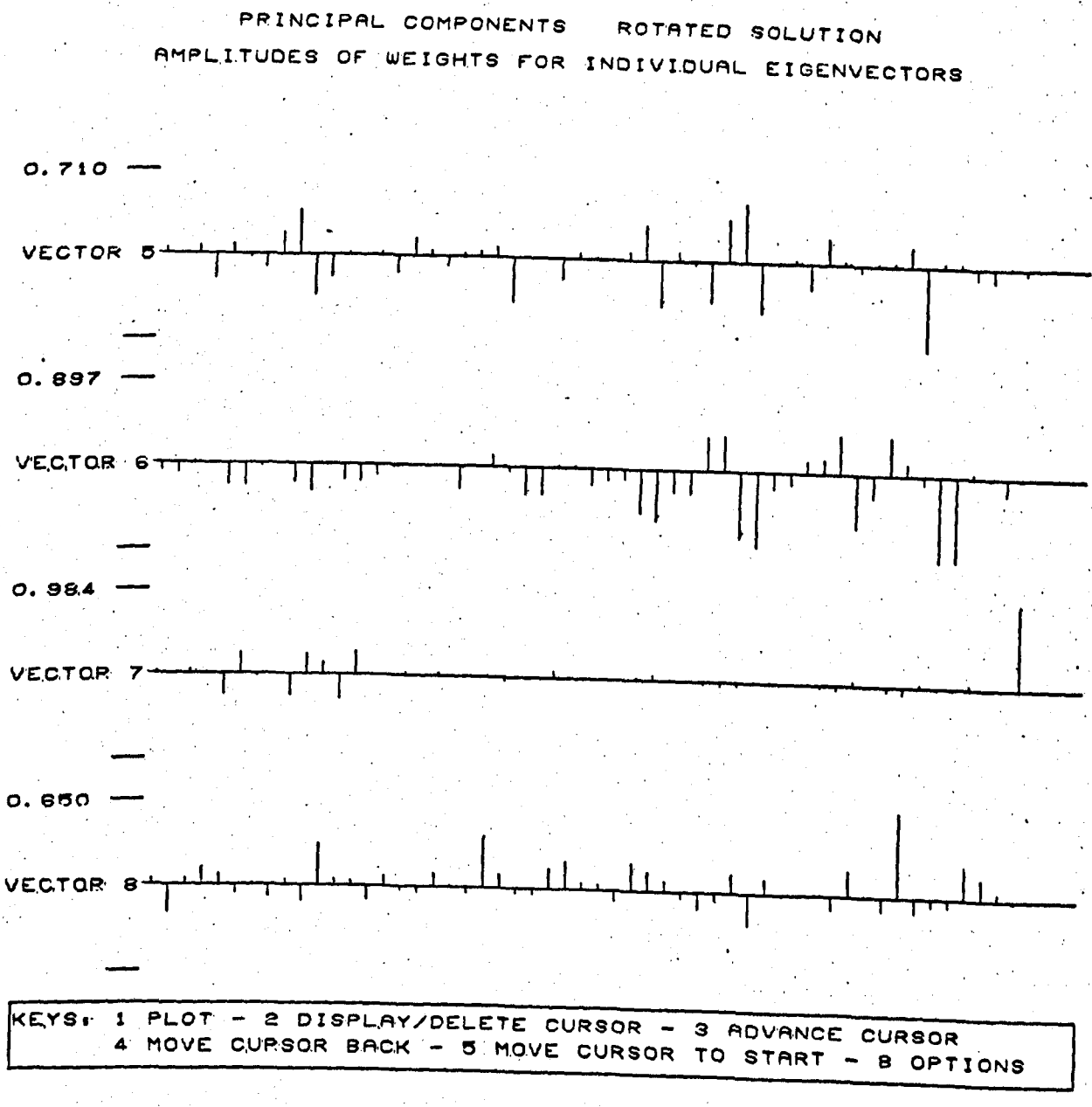


Figure 5.7. Childrens' palate study: rotated components 5 - 8

The variables with relatively large amplitudes for the third component are all ratios representing asymmetry; V21, V22, V25, V26, V46, V47, V50 and V51. The first four are measured with respect to BC and the second four with respect to HJ. V21 and V46 are the greater half-widths over the lesser half-widths, V22 and V47 are the right side-widths over the left side-widths. Each of these pairs of variables has opposite signs for this component.

For component four, all the variables appear to relate to the overall size of the palate. V1 and V2 are the right and left half areas ABXA and ACXA, V14 is the sum of these two. V30 and V31 are the angles AHJ and AJH, and for these palates if these angles are large the palates will be large. V43 is the mean of these two angles. V11 is not a variable quantity; it measures AX, a fixed length of 20mm for all observations.

For component five, V9, V10, V36, V37 are all angles of alveolar convergence, right and left, with respect to the two different base lines. For instance V36 is the angle E(R)HV and V37 is the angle E(L)JV. V34 and V35 are right and left side widths at HJ and V47 is the ratio of these two. V22 is the ratio of the right and left side widths at BC. V30 and V31 are the angles AHJ and AJH. These variables indicate the relative widths of the right and left sides of the palate. Where there are pairs of variables for right and left measurements, these have opposite signs.

For component six, the seven variables with the largest amplitudes, V30, V31, V36, V37, V43, V48 and V49, are all variables which measure angles at the lower base line HJ, or variables derived from these angles.

Component seven has only one variable with a large amplitude, V53. This measures the mean vertical distance between F(R) and F(L).

Component eight has three variables with large amplitudes, V21 and V46. These are again both asymmetry ratios; lesser half width/greater half width measured at the two different base lines. There is also V11 which measures AX (20mm).

The last variable, V11, appears in several components for the eight component rotation. However when ten components are rotated it appears only in component 4 and it is the only variable of any significance in that component. The other variables that were in component 4 and also relate to overall size appear in component 9 with V34, V35, V42 and V45 which are all related to the measurement of the right and left side widths measured at the base line HJ.

In conclusion, for the rotation with eight components, some of these components are clearly associated with particular types of measurements. Specifically, component three with ratios of asymmetry, component four with overall size and component five with the relative widths of the right and left sides of the palate.

Without pursuing the questions raised by these results in relation to cleft palate surgery, it may be useful to point out that in a statistical experiment comparing normal palates used as controls and cleft palates which have been repaired, the results of the surgery may be better described by one group of variables such as those representing angular displacements rather than other groups. This comment is incidental to the present discussion, which seeks simply to demonstrate how interactive graphics can be used efficiently to find groups of variables from rotated solutions on the assumption that these groups will prove to be relevant to the experiment they describe.

5.3. An investigation among women at work in the electronics industry

The data used to demonstrate how the program FACTOR functions for factor analysis is data collected by Hill et al. (Hill, Wild and Ridgeway, 1969). This data was for an investigation into motivation, job satisfaction and labour turnover among women working in seven different plants in the electronics industry. As part of this investigation a questionnaire was designed with questions to cover the following areas.

1. Wages
2. Supervision
3. Training
4. Induction into the firm
5. Social relations with peers
6. The firm itself
7. Physical working environment
8. The work itself

The questionnaire was divided into four parts of which only the second and third parts are relevant to this exercise. The third part required each respondent to indicate overall satisfaction or overall dissatisfaction with her job. The second part of the questionnaire consisted of a set of 47 questions and the respondents were asked to endorse one of four answers to each question, to indicate their level of satisfaction with various aspects of their job. These responses were divided into two groups depending on what the respondent had indicated in section 3, i.e. whether she was generally satisfied or dissatisfied. This particular set of data is for 208 satisfied respondents.

This is a conventional factor analysis study to determine if respondents answered questions in ways which were meaningful to the psychologists who designed the questionnaire. It is clear, that in carrying out this analysis

with interactive graphics rather than with a conventional batch program with no displays, results are more easily assimilated and that the investigator may obtain a clearer understanding of the data and the results.

The initial principal components solution produced 13 eigenvalues greater than unity accounting for 62.7% of the total variance, it was therefore assumed that there were 13 factors. Squared multiple correlation coefficients were used as communality estimates. The communalities converged in 9 iterations and are shown in tabular form in Table 5.3 and as

COMMUNALITIES

1	0.58006	13	0.68404	25	0.60955	37	0.60607
2	0.32648	14	0.17084	26	0.61193	38	0.62977
3	0.67048	15	0.66601	27	0.32178	39	0.51701
4	0.60029	16	0.69391	28	0.39739	40	0.56107
5	0.78775	17	0.42211	29	0.61340	41	0.45800
6	0.62421	18	0.70149	30	0.25205	42	0.46089
7	0.45494	19	0.57943	31	0.68185	43	0.44255
8	0.57629	20	0.31541	32	0.70778	44	0.58875
9	0.53341	21	0.59002	33	0.75688	45	0.55537
10	0.55309	22	0.49508	34	0.62464	46	0.67499
11	0.45765	23	0.39795	35	0.61042	47	0.42154
12	0.65409	24	0.49494	36	0.67925		

Table 5.3.

Communalities for study on women at work

amplitudes in Figure 5.8. The initial factor solution was then rotated using the normalised varimax criterion and amplitude diagrams for the first twelve of the thirteen factors are shown in Figures 5.9 - 11. For the majority of these factors, sets of related questions all have large amplitudes on the same factor in such a way that each factor can be associated with one aspect (and in some cases more than one aspect) of the job. The factors and their associated questions or variables are as follows:

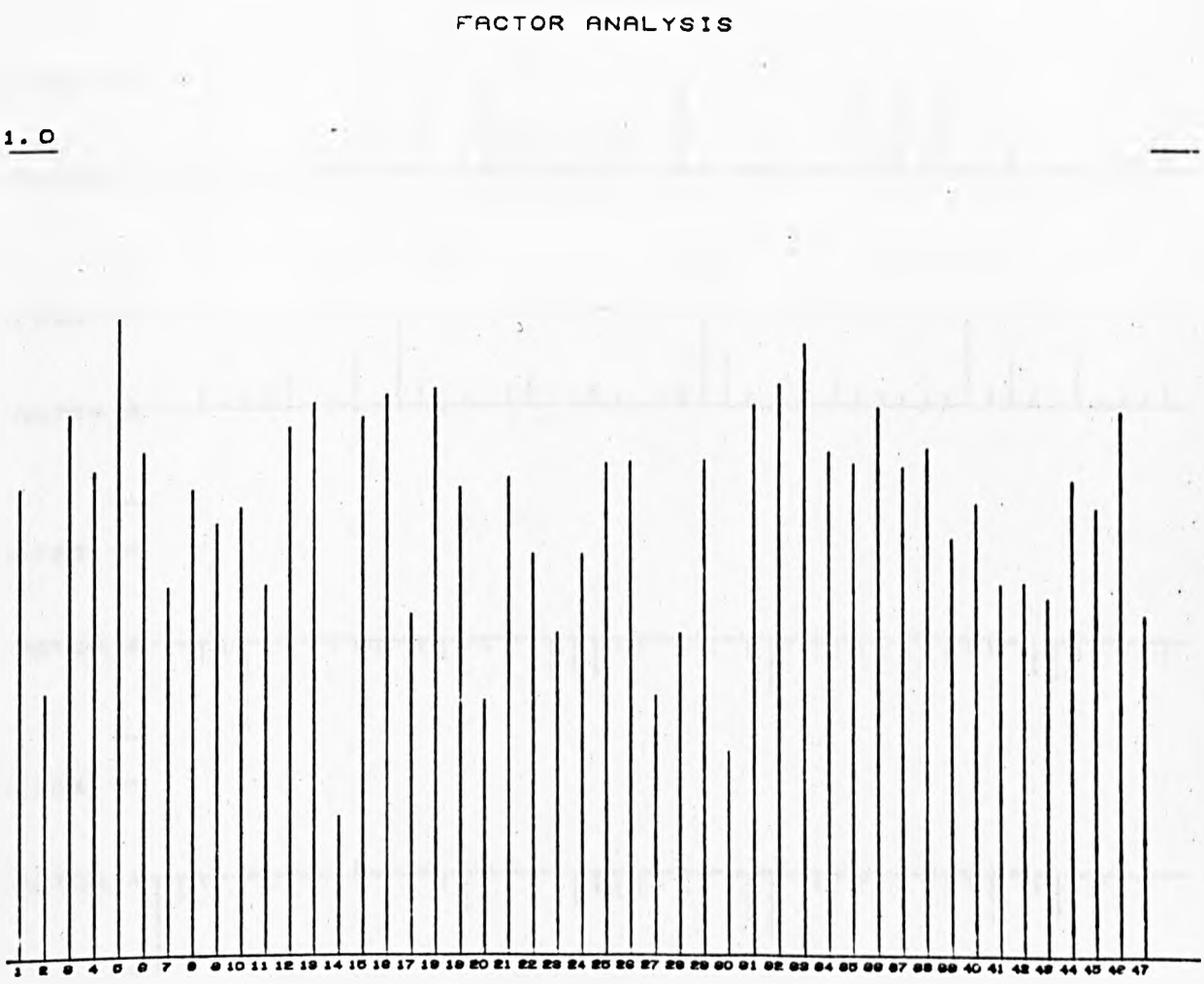


Figure 5.8. Women at work: communalities

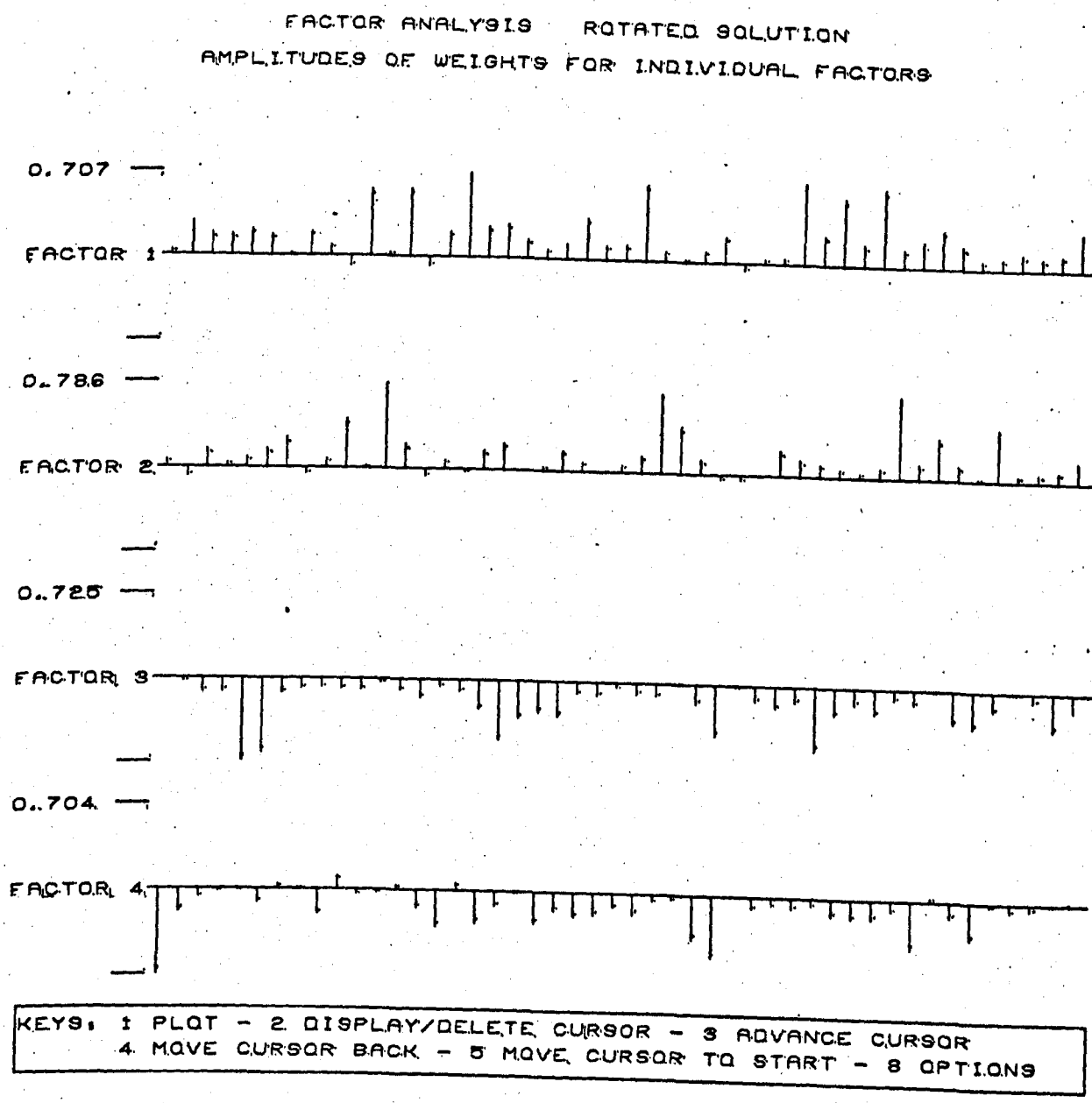


Figure 5.9. Women at work: rotated factors 1 - 4

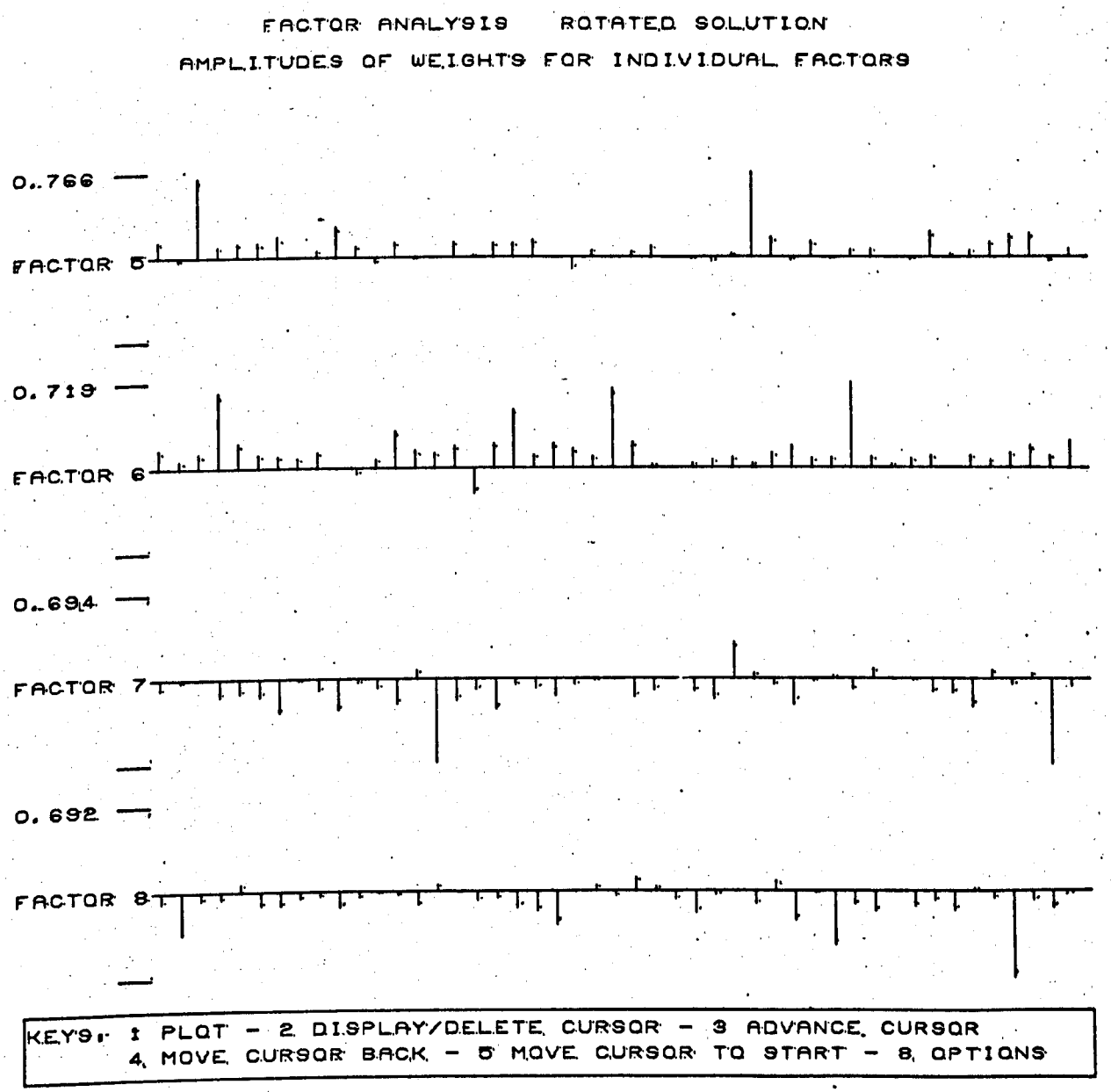


Figure 5.10. Women at work: rotated factors 5 - 8

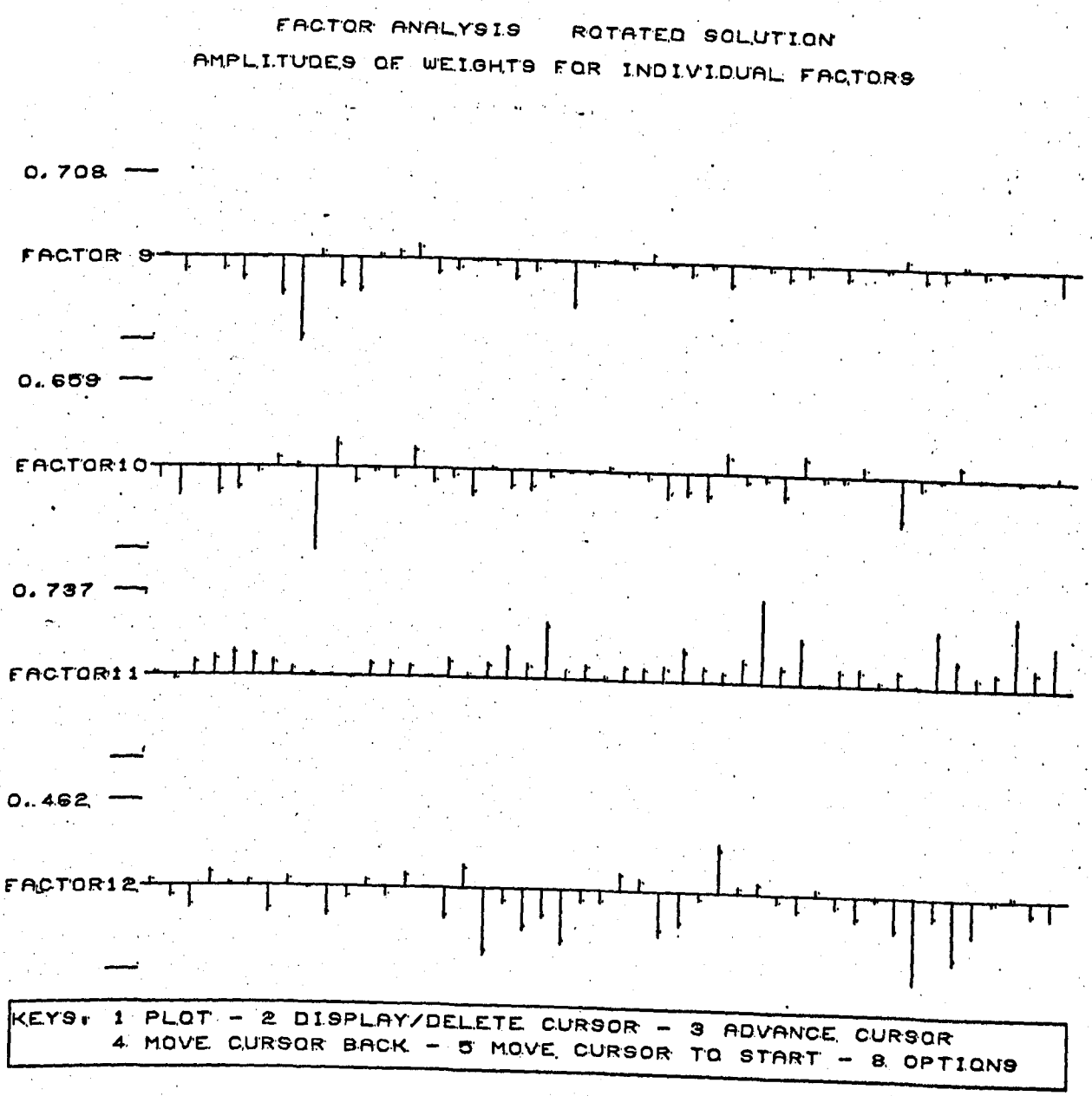


Figure 5.11. Women at work: rotated factors 9 - 12

Factor 1. V2, V11, V13, V16, V25, V33, V35, V37.

Each of these belongs to one of the following closely allied categories; the respondent's attitude to her work and the amount of control she feels that she has over the way in which she can do her work.

Factor 2. V10, V12, V26, V27, V38, V40, V43.

All these questions are associated with the respondent's relationships with her colleagues and how she settled down socially, with the exception of V40 which is a question about settling to the work. All except V10 refer to initial period of employment with the firm, either during training or immediately afterwards, on the shop floor.

Factor 3. V5, V6, V18, V29, V34.

All except V29 are questions about supervision. Question 29 refers to the condition of equipment.

Factor 4. V1, V15, V17, V20, V28, V29, V39, V42.

All except V42 are about working conditions, although question 42 is related to these.

"Is it possible to choose your own workmates, (i.e. to choose the people you want to sit next to)?"

V23 which is a question about lighting is not amongst those with large amplitudes for this factor.

Factor 5. V3, V31.

Both these questions are about training.

Factor 6. V4, V19, V24, V36.

All these questions refer to pay.

Factor 7. V15, V46.

Questions 15 and 46 are concerned with the physical effort involved in doing the job.

Factor 8. V2, V35, V44.

Each of these questions is concerned with a different aspect of the job; how much control the respondent has over the speed at which she may work, how much variety there is in her work and her prospects for earning more in the future.

Factor 9. V7, V8, V10, V11, V22.

All these questions, with the exception of V10 ("Will your workmates help you out if you get into difficulties on an "off" day?"), are about the amount of control the respondent has over the way in which she may do the job. However, question 2 ("How much control have you personally over the speed at which you work?") is not among this set.

Factor 10. V9, V39.

Both these questions are about working conditions. Question 9 is about noise and has the largest amplitude (- .659). Question 39 which refers to the atmosphere in the shop has a relatively small amplitude (- .371). The first does not appear in factor 4, which is also about working conditions, however the second does.

Factor 11. V21, V32, V34, V41, V45, V47.

All these questions are concerned with the attitude of the firm and the management to employees, although question 34 is about the attitude of the supervisors; this also appears in factor 3.

Factor 12. V18, V20, V22, V27, V30, V40, V42.

This set of questions does not cover any one clearly defined aspect of the respondents' attitude to their work.

Factor 13. (The amplitude diagram for this factor is not given)

V23.

Lighting. This question,which is about working conditions,

appears on its own in this factor. It does not appear in factors 4 or 10 both of which are concerned with working conditions.

Factor scores could subsequently be obtained and displayed as scattergrams within this program, they could also be filed for more detailed display and analysis.

6. CLUSTER ANALYSIS - A STUDY OF CENSUS DATA

A study of some data taken from the 1971 census was intended, primarily to involve the use of the Euclidean cluster analysis program, EUCLID, although as the analysis developed it required the use of several of the programs already described. In view of the extensive use of cluster analysis in this study it was necessary to compare the different sets of results obtained. Various graphical techniques were used to help make these comparisons and therefore this chapter begins with a description of the various means, within this system, for comparing the results or the solutions of the available clustering methods.

6.1. The comparison of cluster analysis solutions

If we take a cluster analysis solution to mean the way in which observations are grouped together as a result of using a cluster analysis algorithm on either the original data matrix, or a matrix of similarity or dissimilarity coefficients, then this section describes how, say two solutions A and B, may be compared using the facilities available within this interactive graphics system. The two solutions are assumed to have the same number of clusters, g, and although they will involve the same set of observations, they need not necessarily involve the same set of variables; for the first solution the set of variables S_A is used and for the second the set S_B.

If the two solutions to be compared are the results of hierarchical clustering methods then the resultant dendrograms may be examined. The following discussion refers to solutions where dendrograms are not available, although it is relevant in instances where they are. If, as a result of two analyses, individuals are assigned two cluster labels, two similar clusters (or clusters with predominantly the same members), one from each solution, will not necessarily have the same label.

The most obvious method of comparison is simply to examine the record identifiers of the members of each cluster. In practice, comparisons are difficult to make in this way unless the solutions are very nearly the same, since the user has to examine and remember to which cluster each individual observation belongs and there is no overall picture to be studied.

Another method is to examine scattergrams and histograms based on variables used in both solutions. Within these scattergrams and histograms the individual cluster can be identified, and examples of histograms for two variables taken from one solution are shown in Figure 6.3. These would have to be compared with histograms for the same variables taken from another solution. In making comparisons in this way only the variables which are common to S_A and S_B can usefully be examined. Examination of these scattergrams and histograms is a lengthy process, and it is difficult to retain the information contained in so many pictures.

A third technique, available within this system, which may be used to compare cluster analysis solutions is the grid which can be displayed in place of a scattergram. If, initially, a modified scattergram is displayed with the axes representing two sets of cluster labels or two solutions, each point in this scattergram will have a pair of co-ordinates with integer values. Any points which belong to the same pair of clusters will appear superimposed on one another, and it will be impossible to see how many there are at any one point of the scattergram and therefore how many observations belong to the same pair of clusters. A key can be used to display a grid (Figure 6.7), formed by horizontal and vertical lines drawn at each tic-mark, in this case at each of the values 1, ... g, along both axes. Instead of the data points integer values are displayed, indicating the number of points which appeared in each region of the scattergram, and

if there are no points in a region the relevant section of the grid is left blank.

If the two solutions are identical there will be g entries; one in each row and one in each column. Any entries in the grid over and above the g entries give an indication of the number of differences in the two solutions. This grid gives no indication of precisely which observations are in which cluster.

Labels attached to observations as a result of cluster analysis are arbitrary. If two solutions are similar the labels can be changed interactively, using the grid, and adjusted so that the same labels represent the 'same' clusters.

Scattergrams with NLM co-ordinates proved the most useful for comparing cluster analysis solutions in the census data study. These scattergrams give a more realistic representation than scattergrams for the basic variables, since they take into account all the variables used in the cluster analysis.

The non-linear mapping routine and the resultant two dimensional co-ordinates for mapped observations can be used to compare two cluster analysis solutions as follows. First two solutions A and B are obtained, using the sets of variables S_A and S_B, and also two sets of cluster labels. The same set of variables S_A, which was used for the cluster analysis for solution A is input to the NLM routine and a set of co-ordinates C_A obtained, and likewise co-ordinates C_B for the set of variables S_B. A scattergram is then displayed using the co-ordinates C_A with the cluster labels obtained from the cluster analysis using S_A. This is compared with a similar scattergram using the co-ordinates C_B with the cluster labels obtained from a cluster analysis using S_B.

Two such scattergrams are shown in Figures 6.5 and 6.6, where the same set of individuals has been used for the cluster analysis and for the non-linear mapping, but the variables used for Figure 6.6 form a subset of those used for Figure 6.5. The labels for similar clusters will not necessarily be the same in these two scattergrams, and also it may be difficult to identify individuals from the relative positions of data points, since, if S_A is not equal to S_B, the two mappings differ. If the two solutions are similar the cluster for one solution can be relabelled using the grid technique described above. It is then possible, in principle, to examine the relative positions of clusters. The clusters shown in Figure 6.6 have been relabelled using the grid in Figure 6.7 and the relabelling is shown in Figure 6.8. A further useful technique is to display the labels for the cluster analysis solution using S_B with the NLM co-ordinates C_A. In Figure 6.9 the co-ordinates are those shown in Figure 6.5, but the labels are those for the solution given in Figure 6.8.

Which of these techniques is most suitable for the comparison of solutions depends on the data involved and the solutions obtained. It requires many, rapidly produced pictures to find the display or displays which best demonstrate any similarities which exist.

6.2. A cluster analysis solution

Data from the 1971 census used for this study is socio-economic data for 144 enumeration districts (E.D.) in Newcastle-under-Lyme (National Population Census, 1971). The initial object of the study was to determine clusters within the 144 E.D.'s, which could be used as a basis for sampling for a subsequent survey on recreational activities. For this survey, for practical reasons, 6 to 12 clusters was considered optimum. The 37 variables chosen to describe the E.D.'s are listed in Table 6.1.

Initially the hierarchical clustering program, CLUSTER, was used, with a matrix of distance coefficients for input, and although the results obtained from this program did not ultimately prove to be of value, two results are briefly mentioned. For the single-link method chaining occurs, and therefore, if the number of clusters is chosen to be 8, there is one cluster with 137 members and seven clusters each with one member. The dendrogram in Figure 6.1 is the result of furthest neighbour clustering. The interactive graphics program allows a line to be drawn across the dendrogram, in this case cutting 8 vertical lines thus defining 8 clusters. After displaying this a scattergram, using for instance NLM co-ordinates, could be displayed with the data points represented by their eight cluster labels.

For the non-hierarchical clustering, using the program EUCLID, five different solutions were obtained using different configurations for the initial cluster centres, each starting with 20 clusters and systematically reducing the number of clusters down to 3. These solutions used for the initial cluster centres, a set of random co-ordinates, three sets of user defined co-ordinates using three different scattergrams and lastly the first 20 data points. These five runs all gave the same solutions for 9 clusters and therefore also for 8 clusters. There was a further run using random co-ordinates for the initial cluster centres but starting with 12 clusters

No.	Identifier	Variable Description
V1	TOTPOP	Total population
V2	EAMALES	Economically active males/1000 over 15 yrs
V3	EAFEMS	Economically active females/1000 over 15 yrs
V4	CHOTO4	Children between 0 - 4 yrs/1000
V5	CH5TO14	Children between 5 - 14 yrs/1000
V6	STDGT15	Students over 15 yrs living at home/1000
V7	MALEGT65	Males over 65/1000
V8	FEMSGT60	Females over 60/1000
V9	GBBORN	British born/1000
V10	HHOLDS	Total number of households
V11	TWOCARS	Total of households with 2 or more cars
V12	OWNERS	Owner occupied households/1000
V13	COUNCIL	Council tenant households/1000
V14	UNFURN	Unfurnished tenancies/1000
V15	SHARED	Shared dwellings/1000
V16	AMSEXC	Households with all amenities/1000
V17	NOBATH	Households sharing or lacking a bath/1000
V18	NOWC	Households with no inside W.C./1000
V19	GT1P5PPR	Households with more than 1.5 persons a room/1000
V20	LTOP5PPR	Households with less than 0.5 persons a room/1000
V21	ROOM1TO3	Households with 1-3 rooms/1000
V22	ROOMSP7	Households with more than 7 rooms/1000
V23	ONEPHH	Single person households/1000
V24	TWOPHH	Two person households/1000
V25	NOCAR	Households without a car/1000
V26	YR1MIN	Married male migrants within local authority area during last yr
V27	YR5MIN	Married male migrants within local authority area during last 5 yrs
V28	YR1MOUT	Married male migrants into area during last yr
V29	YR5MOUT	Married male migrants into area during last 5 yrs
V30	HIGHSEG	Economically active males in SEG 1, 2, 3, 4, 13
V31	LOWSEG	Economically active males in SEG 7, 10, 11, 15
V32	OND	Total population with OND, School Cert., A level
V33	DEGREE	Total population with HND or degree
V34	CTRAVIN	Travel to work by car within local area
V35	CTRAVEX	Travel to work by car outside local area
V36	BTRAVIN	Travel to work by bus within local area
V37	BTRAVEX	Travel to work by bus outside local area

Table 6.1. Variables for census data study

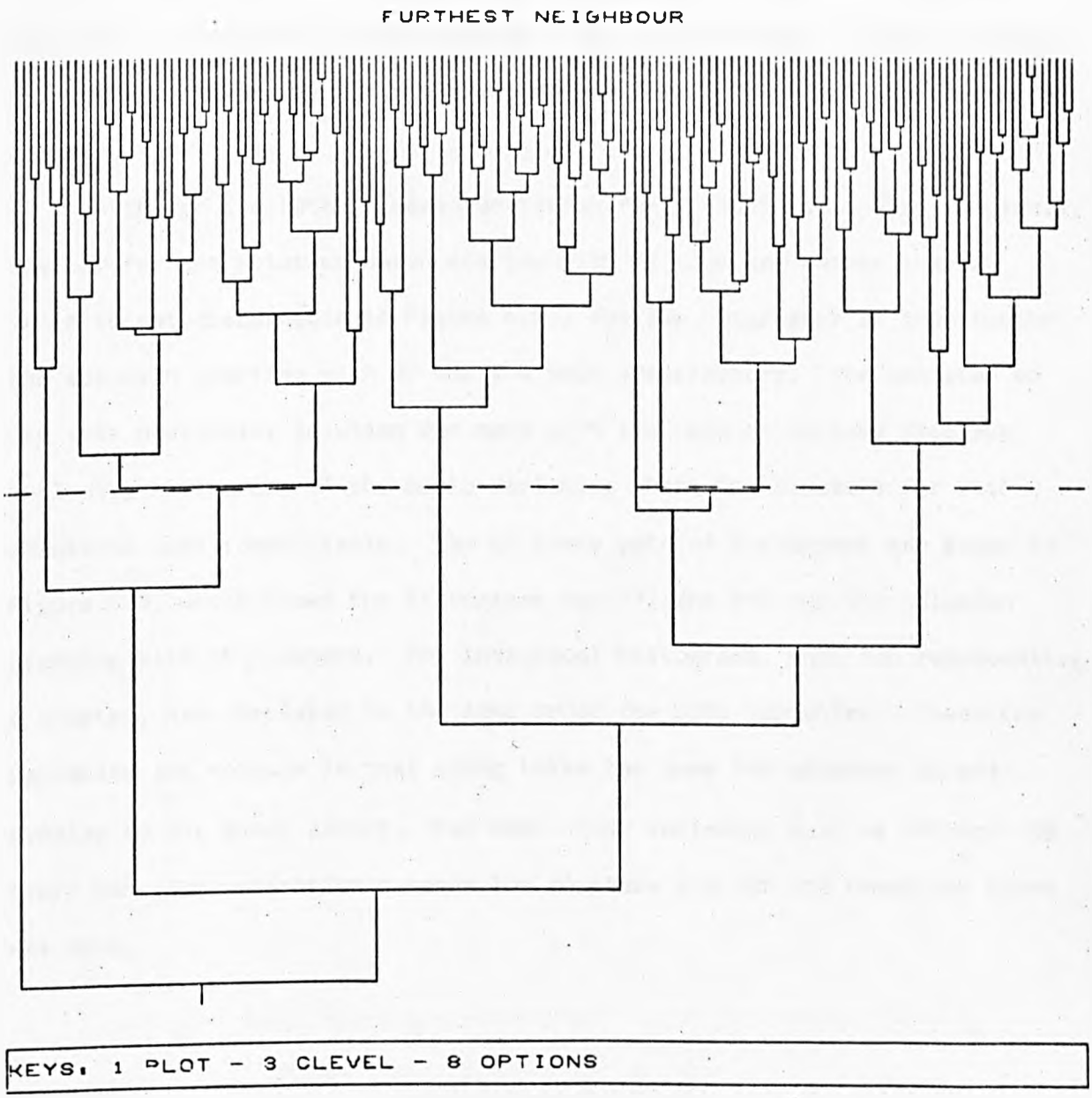


Figure 6.1. Census data: dendrogram for furthest neighbour clustering

instead of 20. These solutions were all examined interactively and graphically and as an example Figure 6.2 shows a display where the number of clusters has been plotted against the total squared deviation of all points from their respective cluster centres. The 1's represent a random starting configuration starting with 20 clusters and the 2's a random start with 12 clusters.

Although the total squared deviation for 8 clusters is very marginally smaller for the solution which started with 12 clusters rather than 20 (this is not discernable in Figure 6.2), for the originator of the problem the solution starting with 20 was the most satisfactory. The decision to use this particular solution was made with the help of various displays including histograms of the basic variables where the clusters for both solutions were identifiable. Two of these sets of histograms are given in Figure 6.3, which shows the histograms for V12 and V13 for the solution starting with 20 clusters. The individual histograms, each one representing a cluster, are displayed in the same order for both variables. These two variables are notable in that along these two axes the clusters do not overlap to any great extent. For some other variables such as V19 and V20 there was some separation between the clusters and for the remainder there was none.

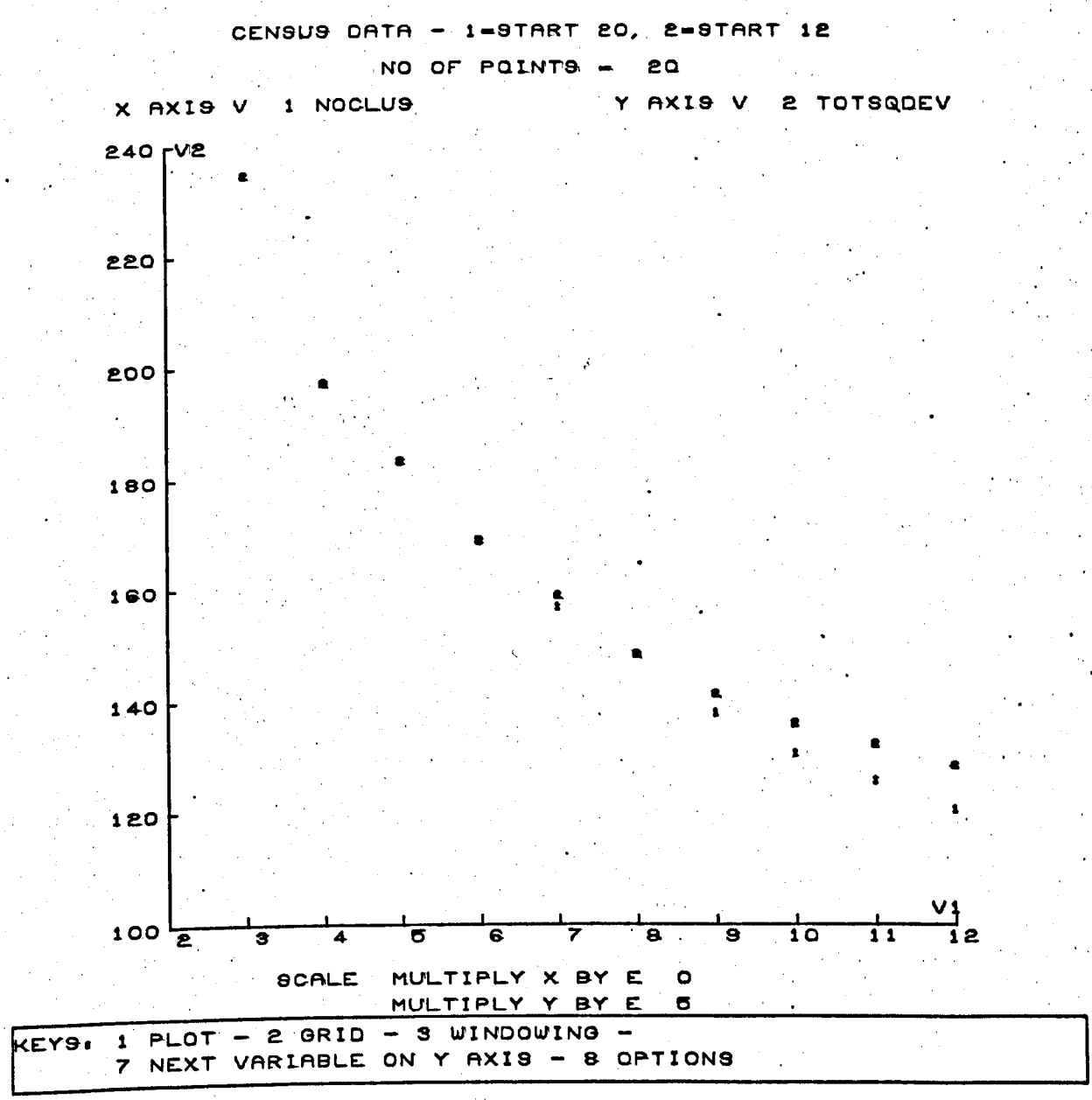


Figure 6.2. Census data: No. of clusters v. total squared deviations

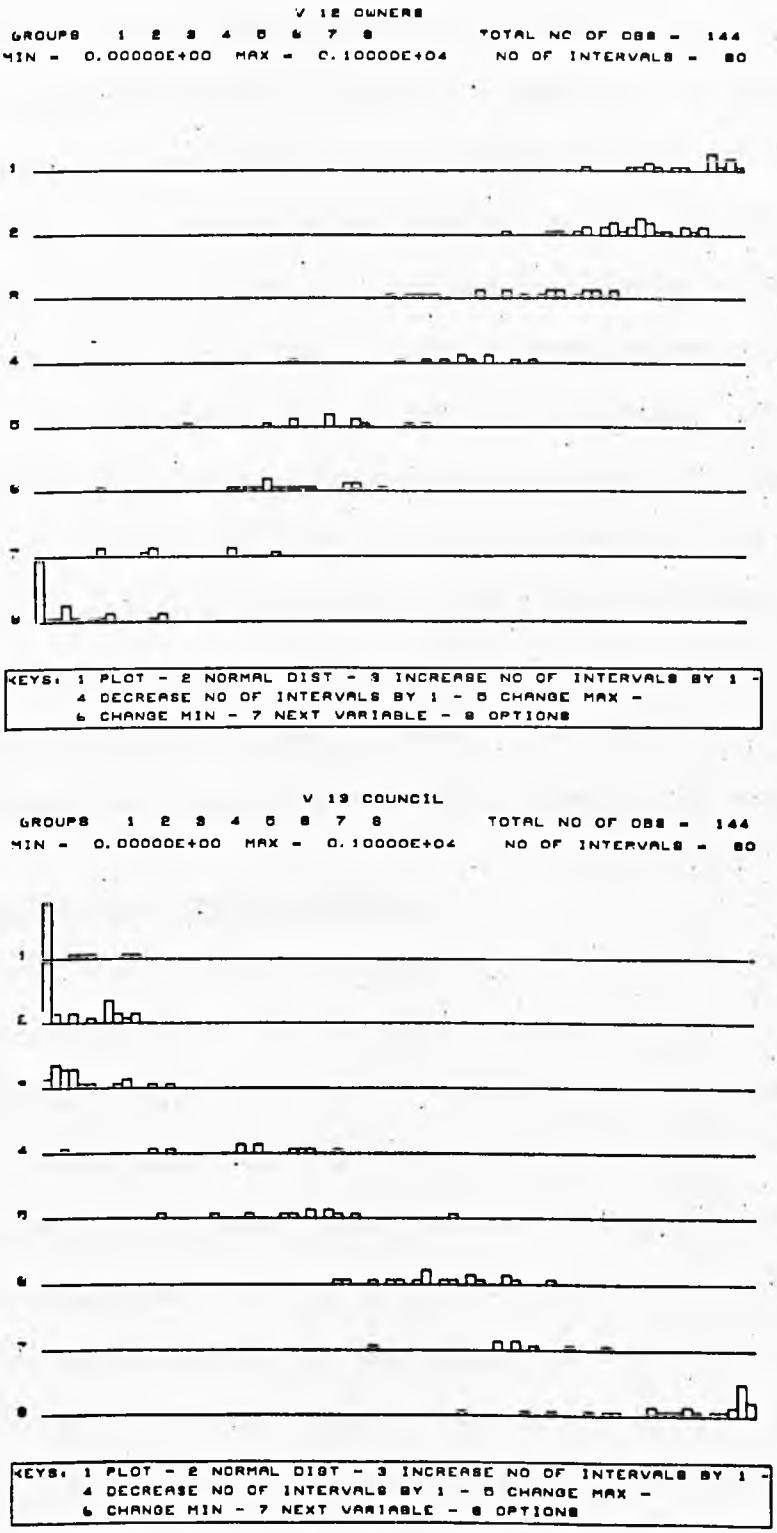


Figure 6.3. Census data: histograms for V12 and V13 for initial cluster analysis solution

6.3. The nature and origin of clusters

The subsequent analysis of this data developed beyond what had been originally planned, largely as a result of the facilities provided by the interactive graphics system, which enabled properties of the cluster analysis solutions to be examined in detail, easily and rapidly. As already mentioned, analysis by scanning sets of histograms for each of the 37 variables showed good separation of the clusters on some variables, some overlapping of clusters on others and complete overlapping on the rest. These properties were also examined in terms of scattergrams with the data projected onto planes defined by chosen pairs of variables. It became apparent that certain variables dominated the solution in determining the clusters obtained and that others had no influence at all. This is not an unexpected result for data of this kind. What is important for present purposes, is the ease with which the situation can be analysed, provided that the appropriate facilities are provided. The rest of this discussion on clustering shows how interactive graphics influenced the analysis.

6.3.1. Finding the first set of clusters

Given that there was good separation of the clusters for this solution for variables 12 and 13, the next step was to see if it was these and other associated variables which had dominated the solution. The first problem was to decide which were the associated variables.

The histograms for several other variables besides V12 and V13 showed reasonably good separation, but how to make the choice of a subset using these histograms was not clear. It was decided to try a principal components analysis as a means of finding groups of related variables. The diagonalisation of the correlation matrix computed with the complete data set (37 variables and 144 observations) gave eight eigenvalues greater than one accounting for 75.9% of the total variance. On inspection variables 12

and 13 were not unambiguously associated with a single vector; they both had relatively large amplitudes on two vectors. The first eight components were rotated to try and resolve this ambiguity and to see if variables 12 and 13 could be associated with just one vector. Amplitude diagrams showing the result of rotating the first eight components using the normalised varimax criterion are shown in Figure 6.4. Here variables 12 and 13 appear with relatively large amplitudes in rotated component 6, and only in component 6.

If S denotes the original set of 37 variables, and set I is arbitrarily chosen as the set of 6 variables with the highest absolute loadings on rotated component 6 then

set I = {V5, V12, V13, V19, V20, V24}

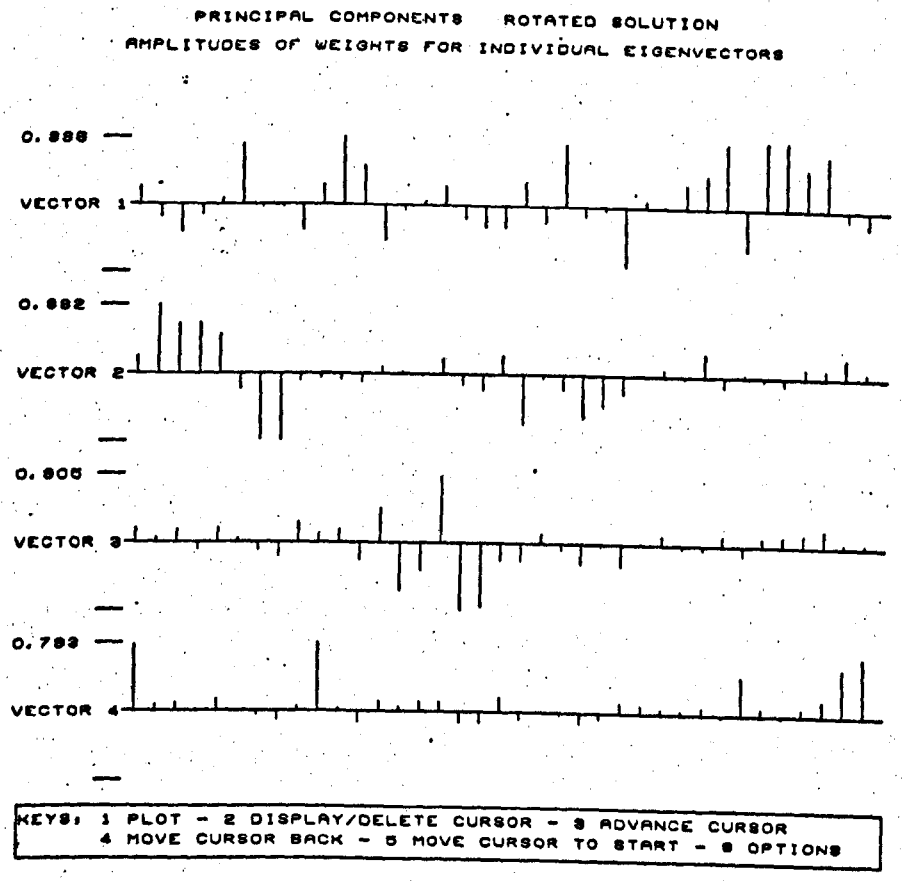
A further arbitrarily chosen subset may be defined;

set IA = {V1,V5,V7,V8,V12,V13,V19,V20,V21,V24,V25,V31,V35,V36},

these are the 14 variables with the highest absolute loadings on the same vector 6.

At this point and subsequently there were further cluster analyses using all 144 observations but different sets of variables. In order to simplify the analysis and to make comparisons easier, in each case a random starting configuration was used for 20 clusters and only the solution for 8 clusters was examined.

The next step was to compare the two cluster analysis solutions, the first using the complete set of variables S and the second using set IA, to see if they are similar and if in fact it was the subset of variables, set IA, which largely determined the solution for the total set S. These two solutions were compared as follows. Figure 6.5 shows the NLM co-ordinates



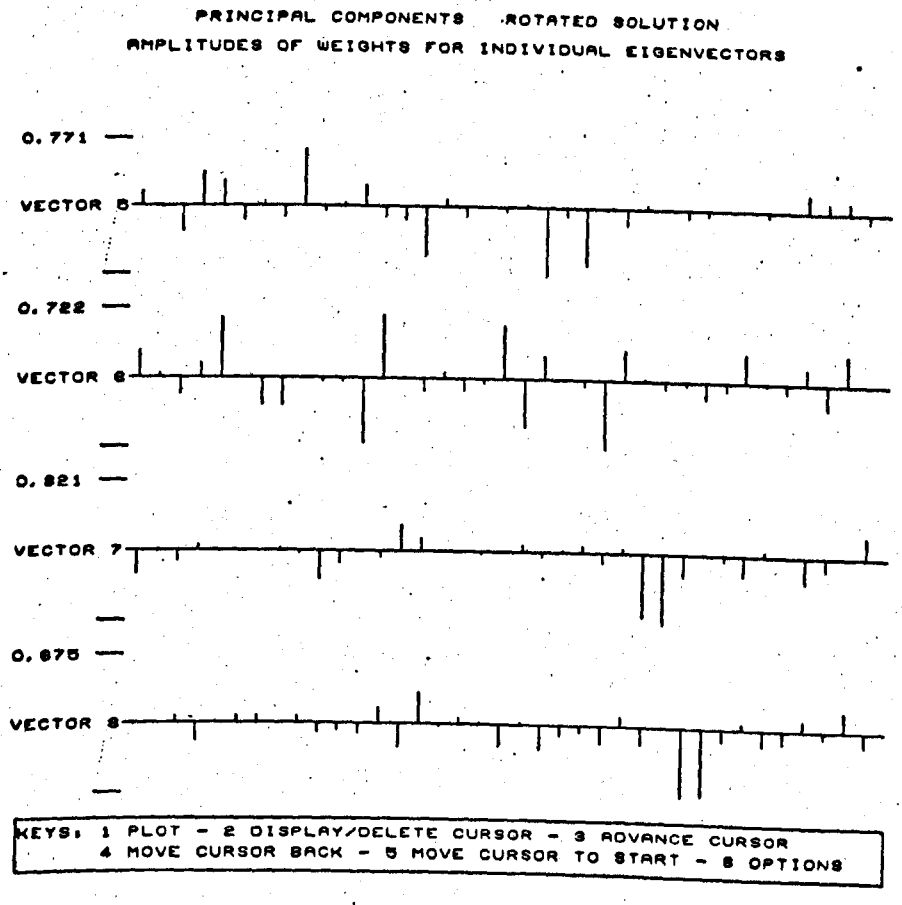


Figure 6.4. Census data: rotated principal components

for the total set with labels for the cluster analysis using the total set. Figure 6.6 shows a similar picture but for set IA. The cluster labels are arbitrary, therefore the grid (Figure 6.7) was used to relabel the clusters for one of these solutions. This was straightforward since the clusters are predominantly the same in both solutions. Figure 6.8 is identical to Figure 6.6 except that the clusters have been relabelled. A comparison of Figures 6.5 and 6.8 shows that similar clusters appear in the same relative positions. A further picture, Figure 6.9, shows the NLM co-ordinates for set S, but with the labels for the cluster analysis using set IA. This again may be compared with Figure 6.5.

Several totally different subsets of variables to set IA were used for cluster analysis solutions and these were compared with the solution for set S, to see if there were any similarities and if an alternative set to set IA would give a similar solution as that for set S. Figure 6.10 shows the data points with the same co-ordinates as Figure 6.5 and 6.9 but they have cluster labels for a solution using a different set of variables to set IA, namely {V2, V3, V4, V23} (= set III). The definition of these clusters is not at all clear, particularly when compared with the definition of clusters in Figure 6.9 which is the equivalent picture for the solution using set IA. Other sets of variables gave similarly ill-defined clusters in terms of these co-ordinates.

In the analysis which followed, extracting set IA from the total set gave more meaningful results than extracting set I, however the NLM scattergram for set I with labels for a solution using set I, Figure 6.11 (the clusters have not been relabelled), is similar in shape to the equivalent scattergram for set IA. The data points are more compact and the clusters more clearly defined, also investigation showed that the solution is similar to those for the total set and set IA. The solution for set IA more closely resembled the solution for set S than did the solution for set I.

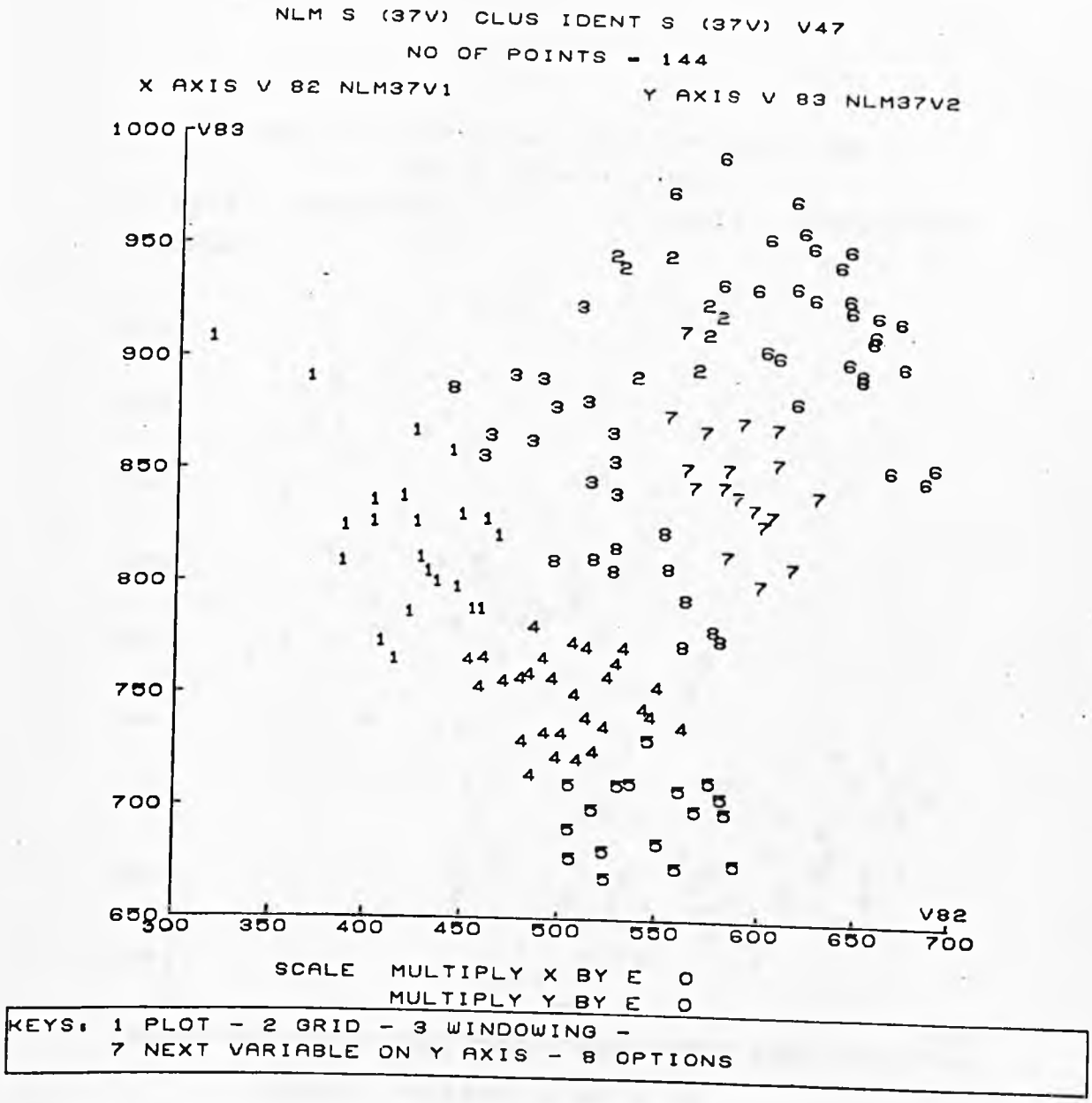


Figure 6.5. Census data: cluster labels set S, NLM set S

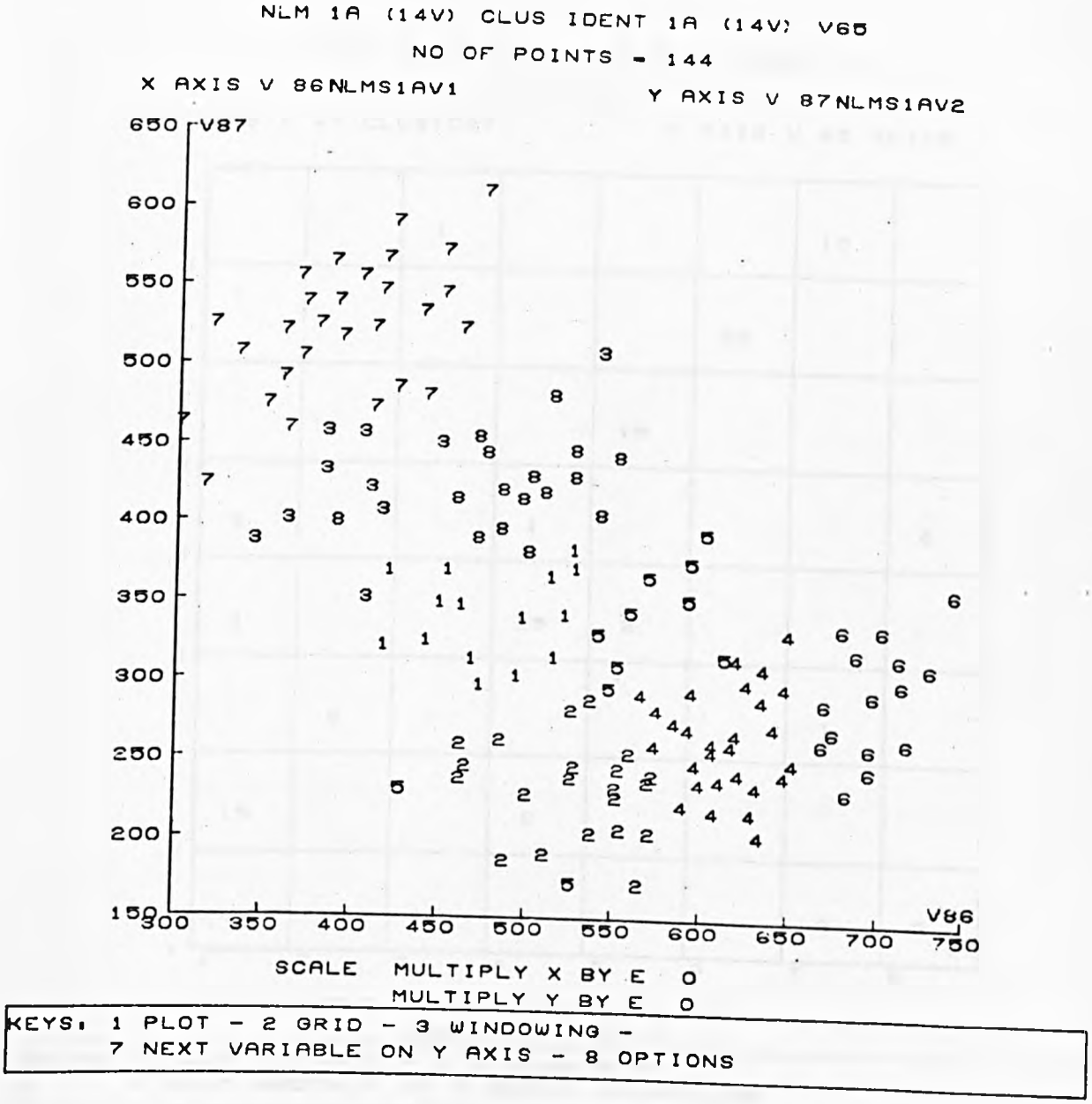


Figure 6.6. Census data: cluster labels set IA (original labels), NLM set IA

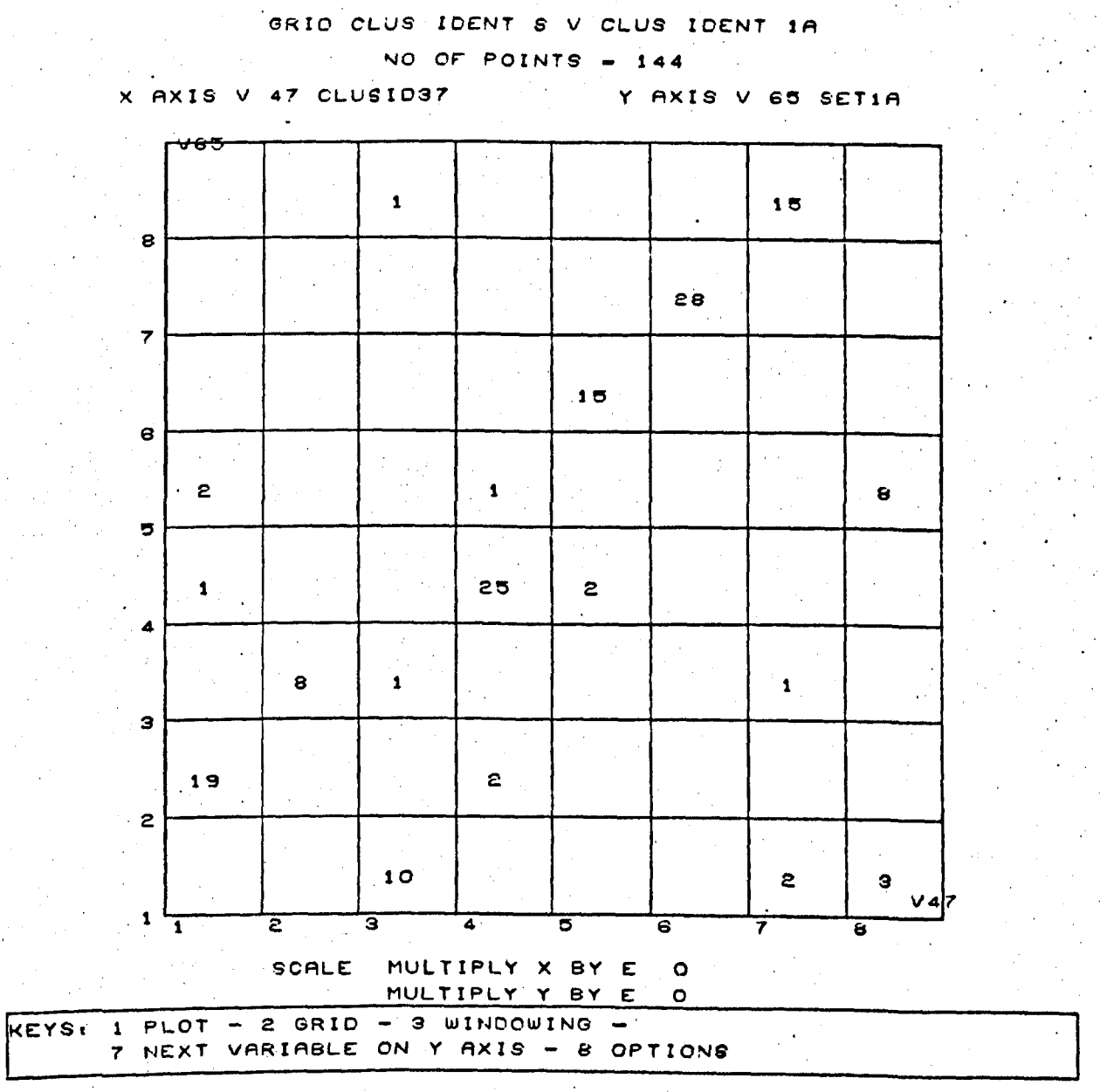


Figure 6.7. Census data: grid for solutions for set S and set IA

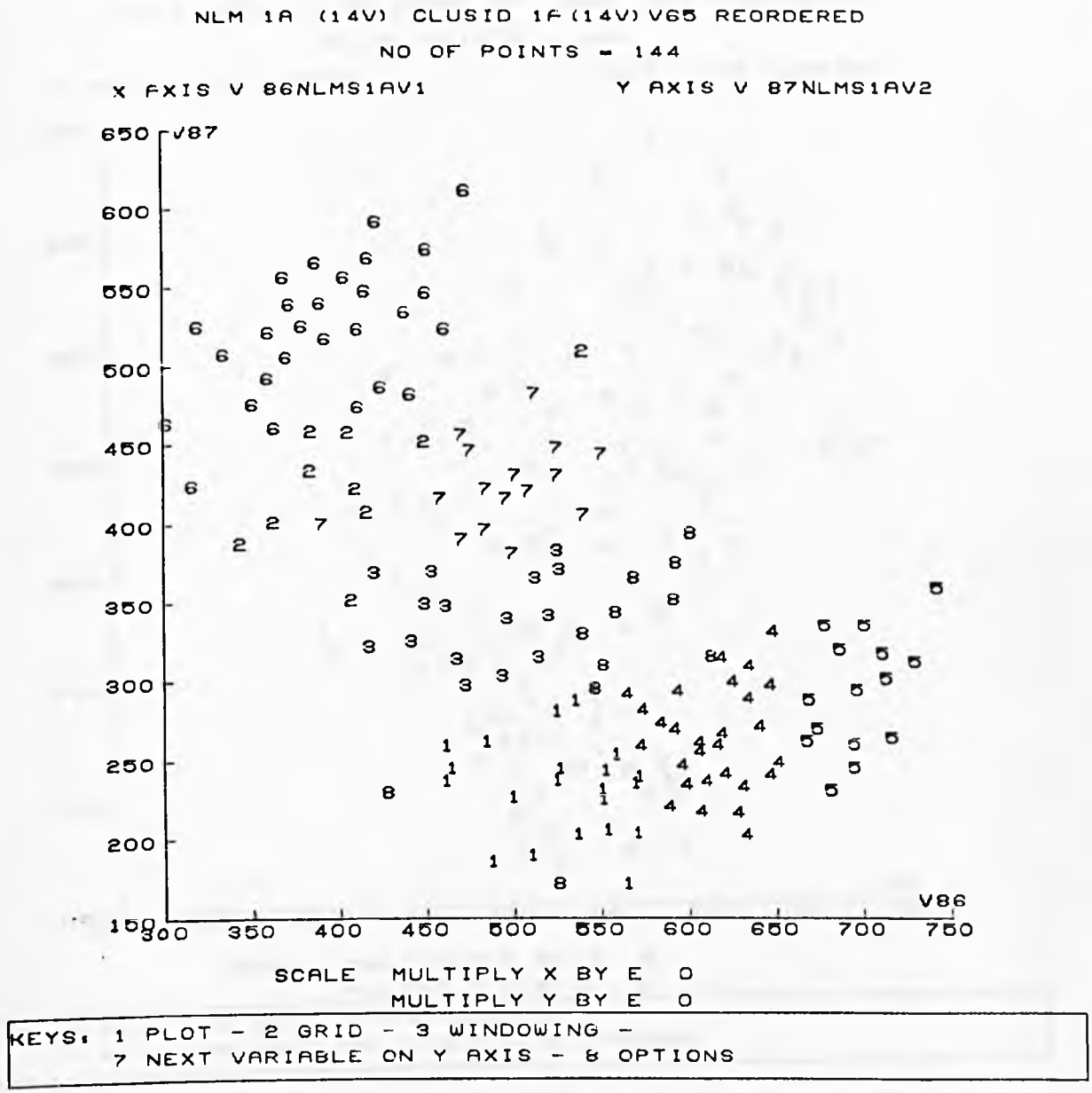


Figure 6.8. Census data: cluster labels set IA (relabelled), NLM set IA

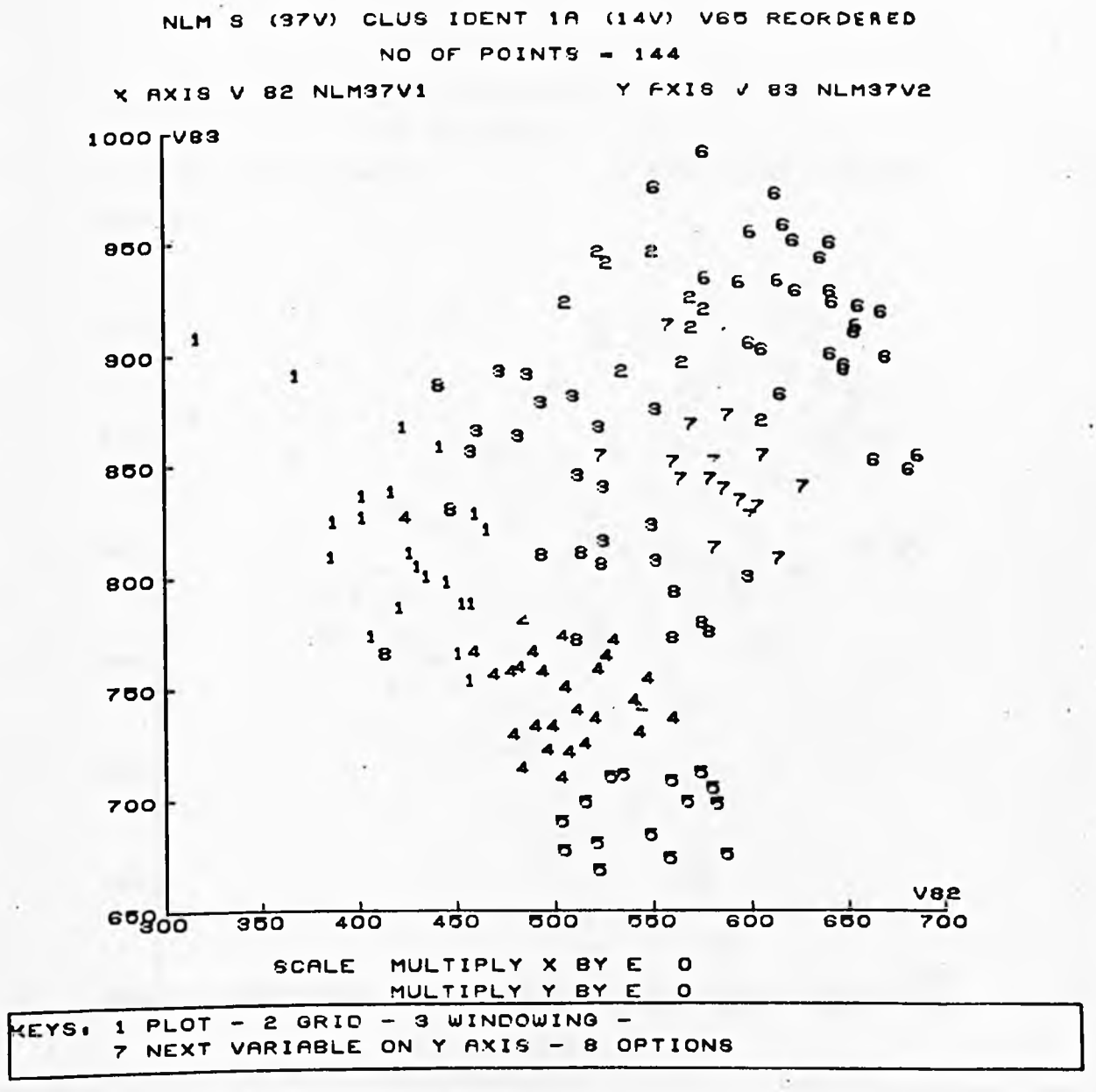


Figure 6.9. Census data: cluster labels set IA (relabelled), NLM set S

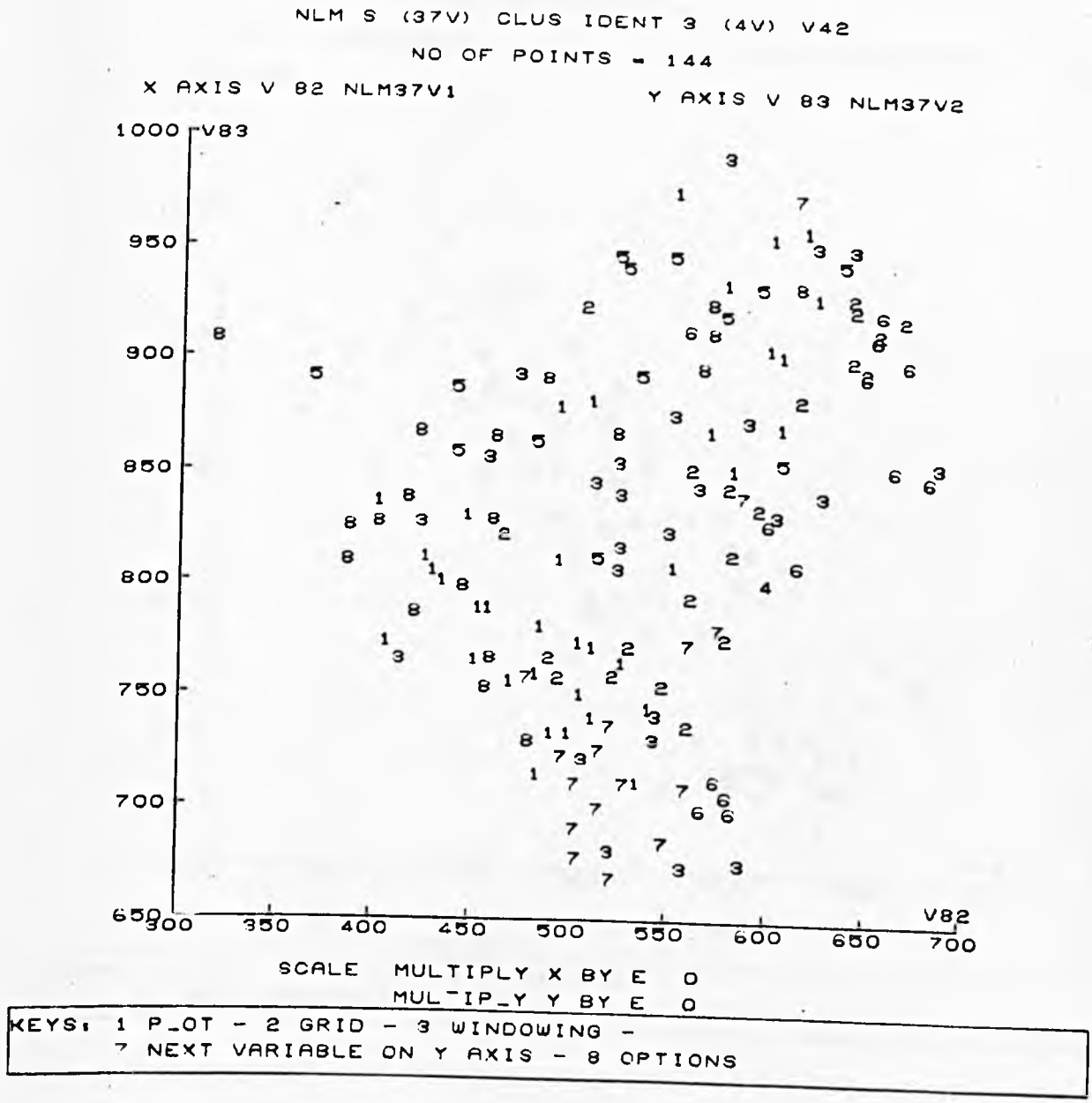


Figure 6.10. Census data: cluster labels set III, NLM set S

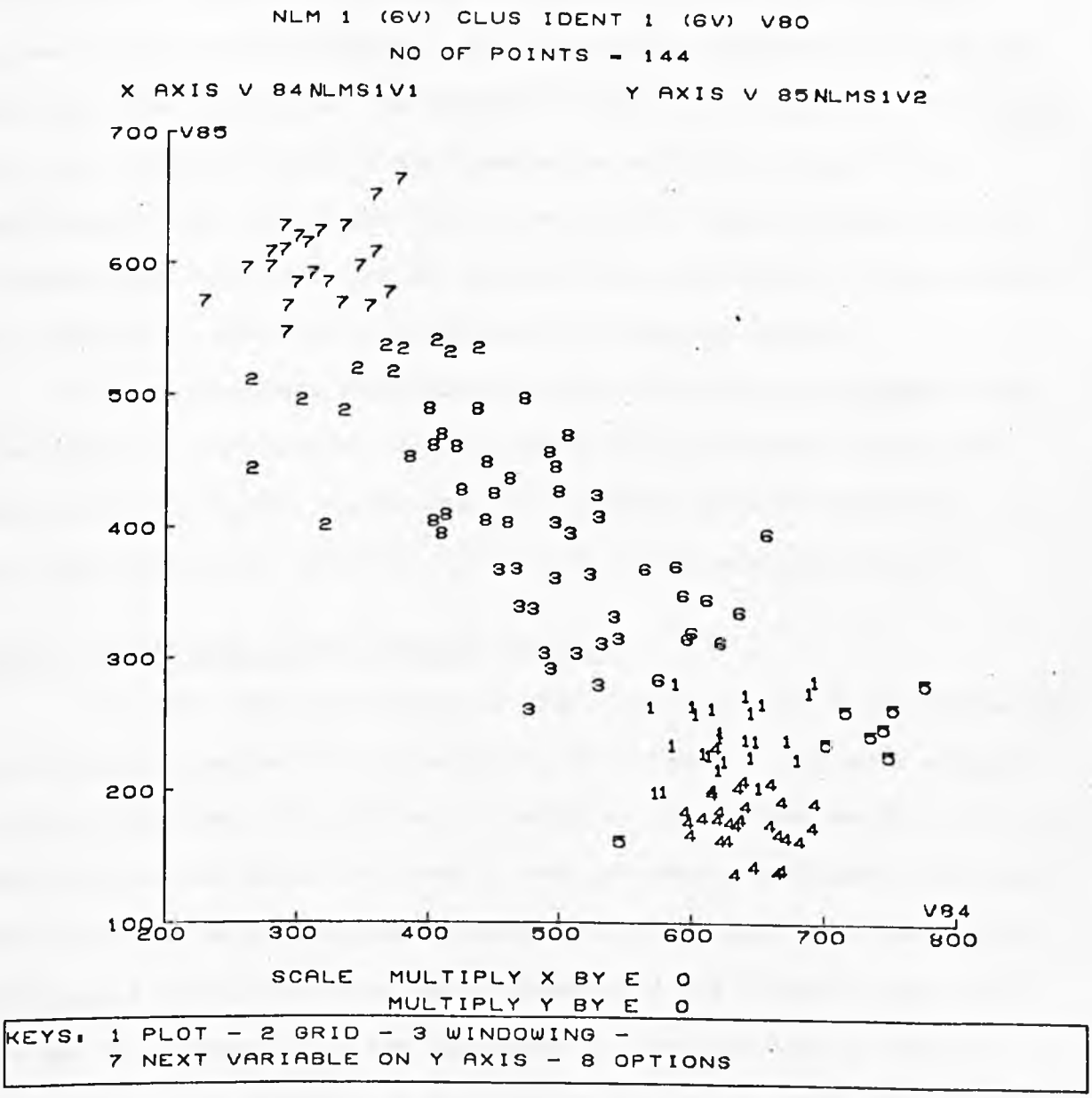


Figure 6.11. Census data: cluster labels set I (original labels), NLM set I

The rotated principal component scores were computed, filed and displayed as eight histograms, one for each cluster of the initial cluster analysis solution. The histograms for the scores on the first and the sixth rotated components are shown in Figure 6.12, (the sixth component appears at the top of the page). The individual histograms are in the same vertical order as they are for variables 12 and 13 in Figure 6.3. The first and sixth rotated components are those where variables 12 and 13 have amplitudes of any significant size. There is very little overlap for the clusters along these axes and the clusters which are adjacent to one another on variables 12 and 13 are also adjacent on these two scores.

These pictures and the comparisons which have been made suggest that the initial cluster analysis solution using set S is almost the same as the solution using set IA, and that the variables which constitute set IA, or a very similar set, dominate the initial cluster analysis solution.

6.3.2. Identifying further sets of variables

The next step was to discover what happened if set IA was removed and the analysis repeated with the remaining 23 variables. A cluster analysis solution was found using this set of variables (namely the set S-IA), and the resultant cluster labels are shown in the scattergram in Figure 6.13 where the NLM co-ordinates have been calculated using the same set of variables. This gave a totally different set of clusters to the analyses using set S and set IA. Inspection of the histograms for the variables in the set S-IA showed good separation of the clusters on variables V16, V17 and V18, and these were chosen as the initial members of a second set of variables. Turning now again to the principal components solution (Figure 6.4), these variables all have large amplitudes on rotated component 3, therefore set II was chosen as {V14, V16, V17, V18}. A further set, set IIA, was defined,

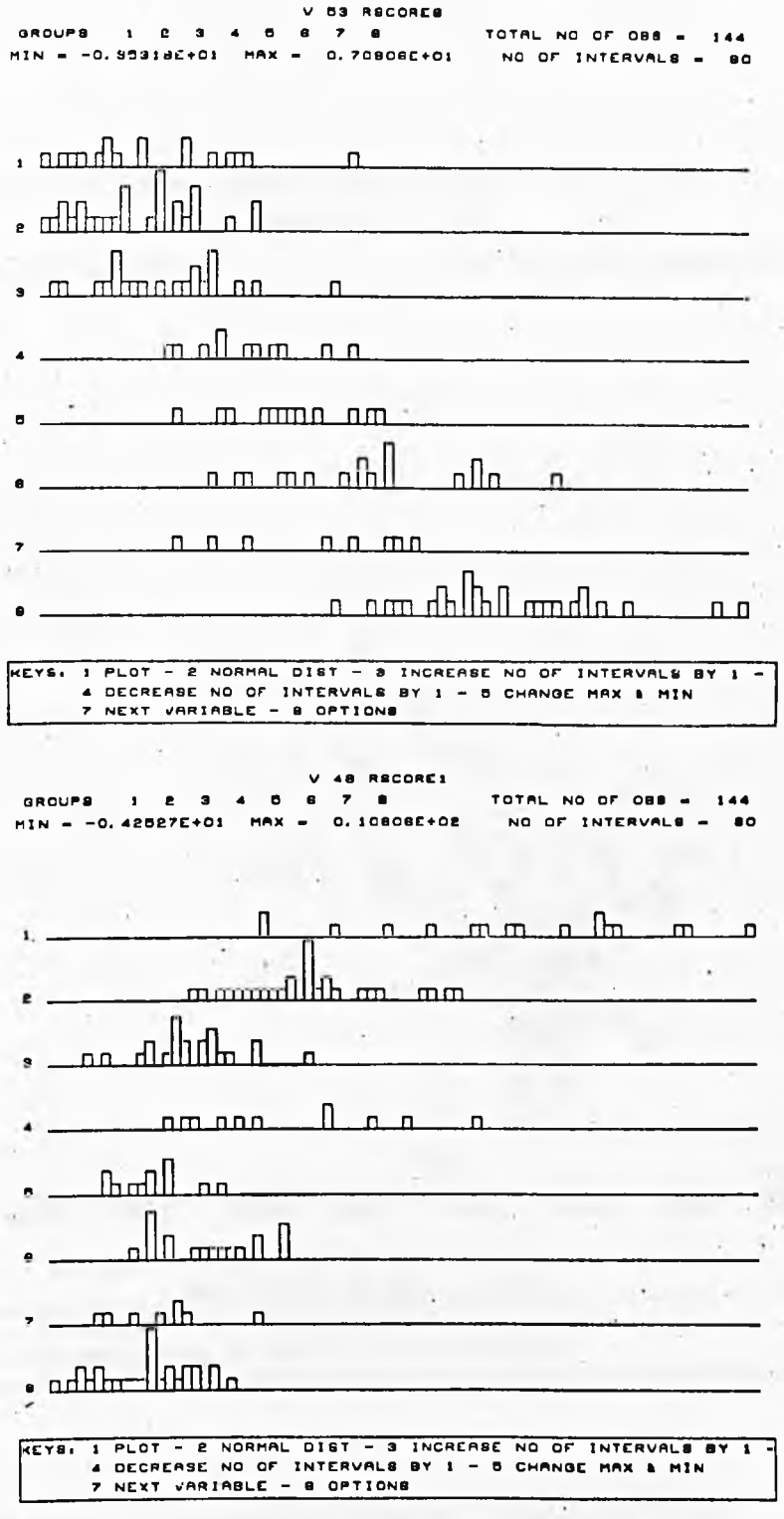


Figure 6.12. Census data: histograms for rotated principal components scores on components 6 and 1

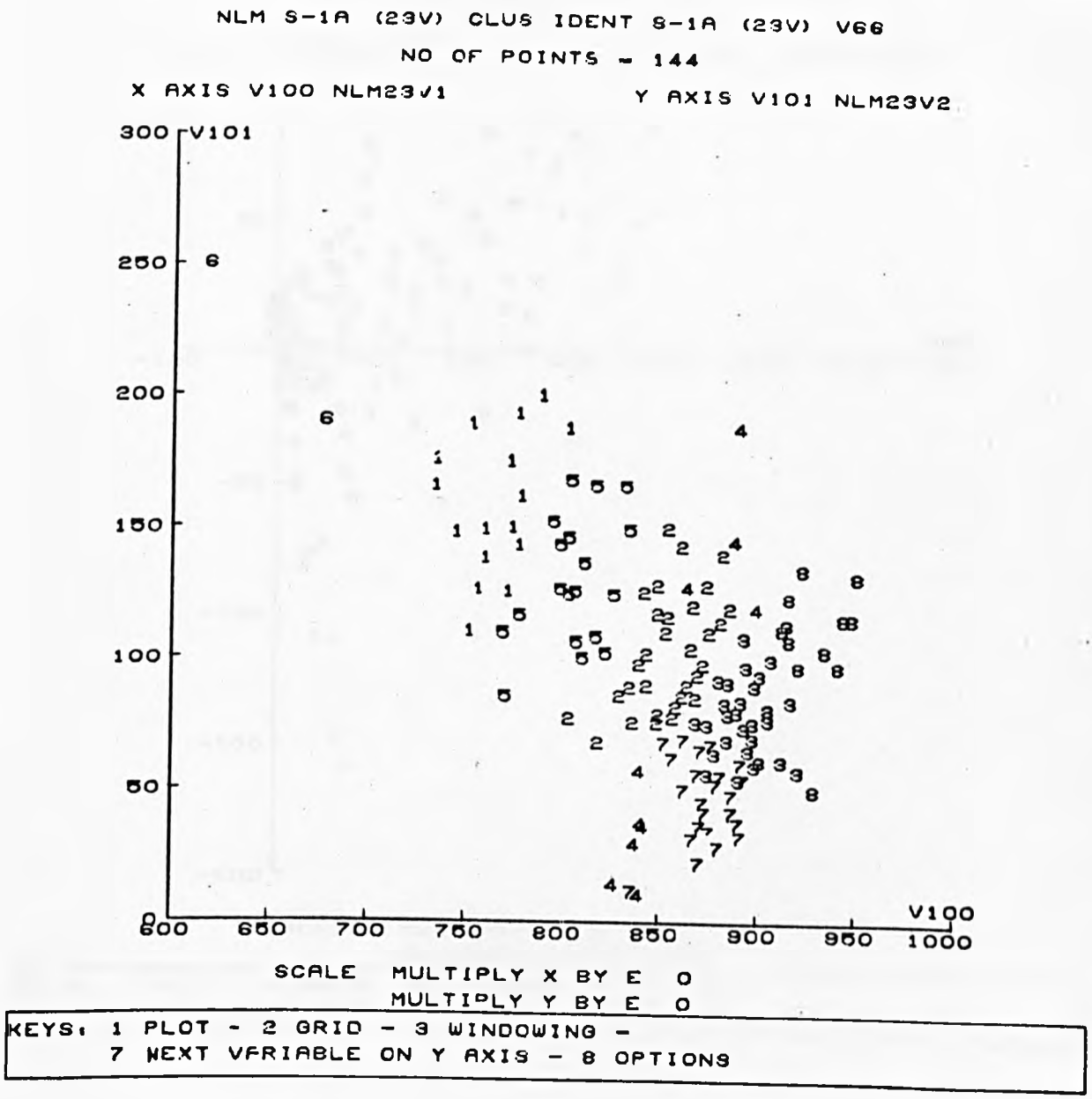


Figure 6.13. Census data: cluster labels set S-IA, NLM set S-IA

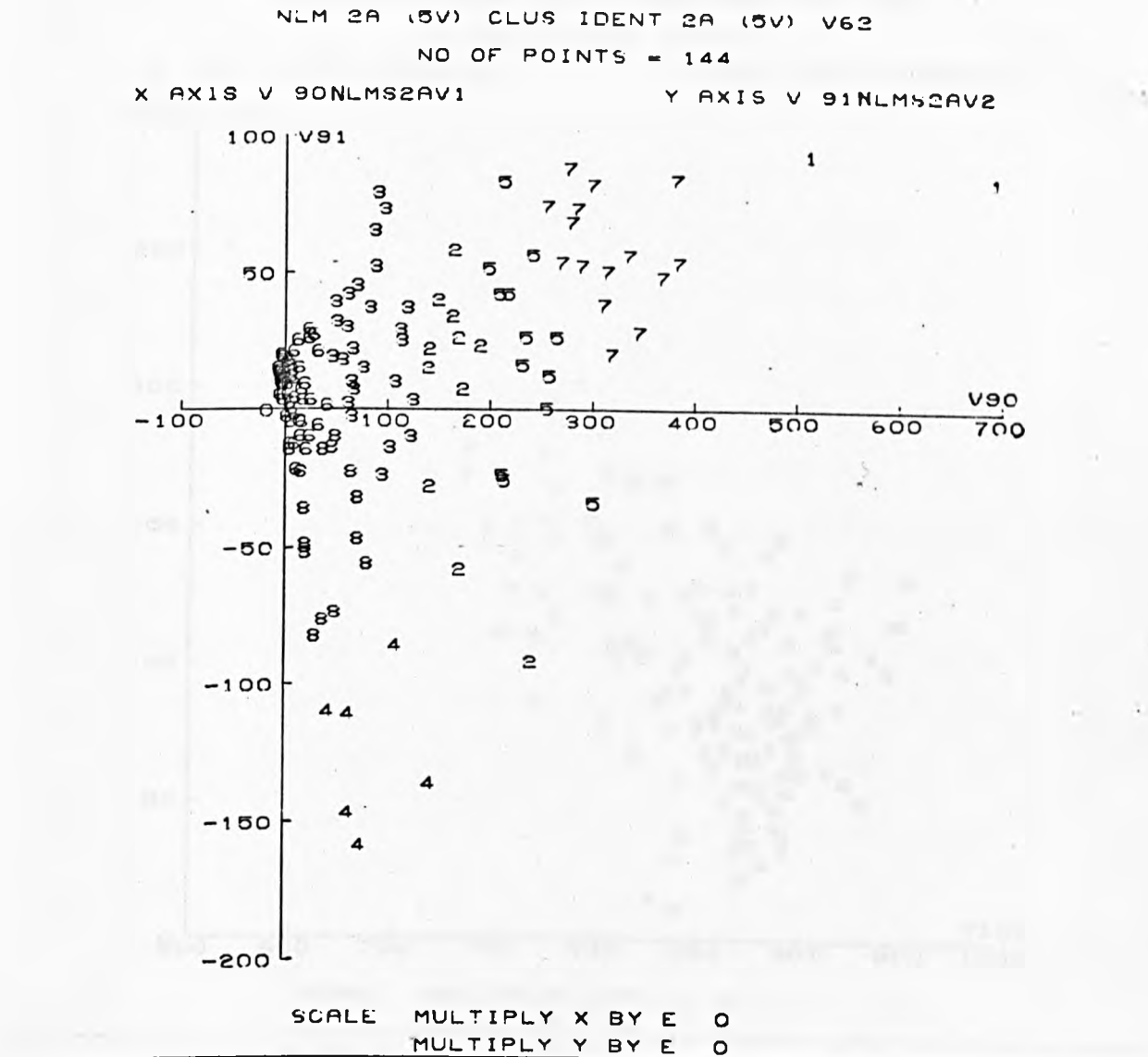


Figure 6.14. Census data: cluster labels set IIA, NLM set IIA

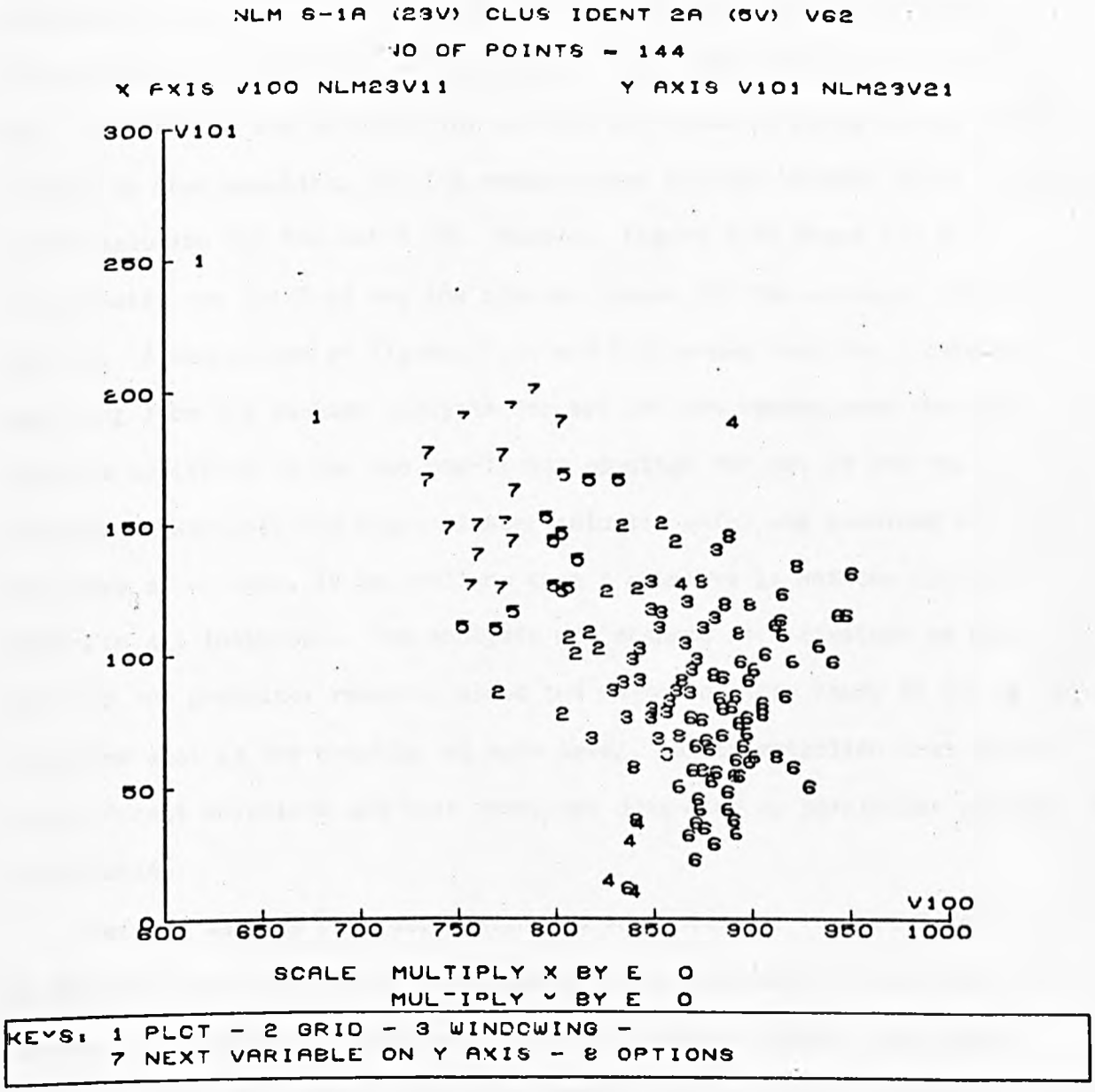


Figure 6.15. Census data: cluster labels set IIA, NLM set S-IA

{V14, V15, V16, V17, V18}, that is, all those variables with the largest amplitudes which have not already been removed as members of set IA. The NLM co-ordinates calculated with the five variables of set IIA and the associated cluster labels are shown in the scattergram in Figure 6.14. The interactive relabelling of the clusters was complicated by the fact that cluster 6 in the solution for set IIA was twice as large as any other cluster in that solution, and its members were divided between three clusters in the solution for the set S-IA. However, Figure 6.15 shows the NLM co-ordinates for set S-IA and the cluster labels for the solution using set IIA. A comparison of Figures 6.14 and 6.15 shows that the clusters resulting from the cluster analysis for set IIA are mapped onto the same relative positions by the two non-linear mappings for set IA and for set IIA. Although it was only the eight cluster solution which was examined in this and every other case, it may well be that 8 clusters is not the correct number in all instances. The analysis was stopped at 8 clusters on each occasion for practical reasons, since the object of this study is not to determine what is the solution at each level, but to establish that there are different solutions and that these are dominated by particular subsets of variables.

Set IIA was now removed, leaving 18 variables, or the set S-IA-IIA, to discover what happened if the process of removing sets of variables was repeated. A further solution was found which was different from those obtained with set S or set S-IA. The labels for the cluster analysis solution for set S-IA-IIA are shown in Figure 6.16 with NLM co-ordinates calculated with this same set of variables. Histograms for this solution showed the clusters reasonably well separated on variable 23, and also on variables 2 and 3. Because of the separation on V2 and V3, rotated component 2, in Figure 6.4, was chosen rather than rotated component 5 where V2 and V3 have only small

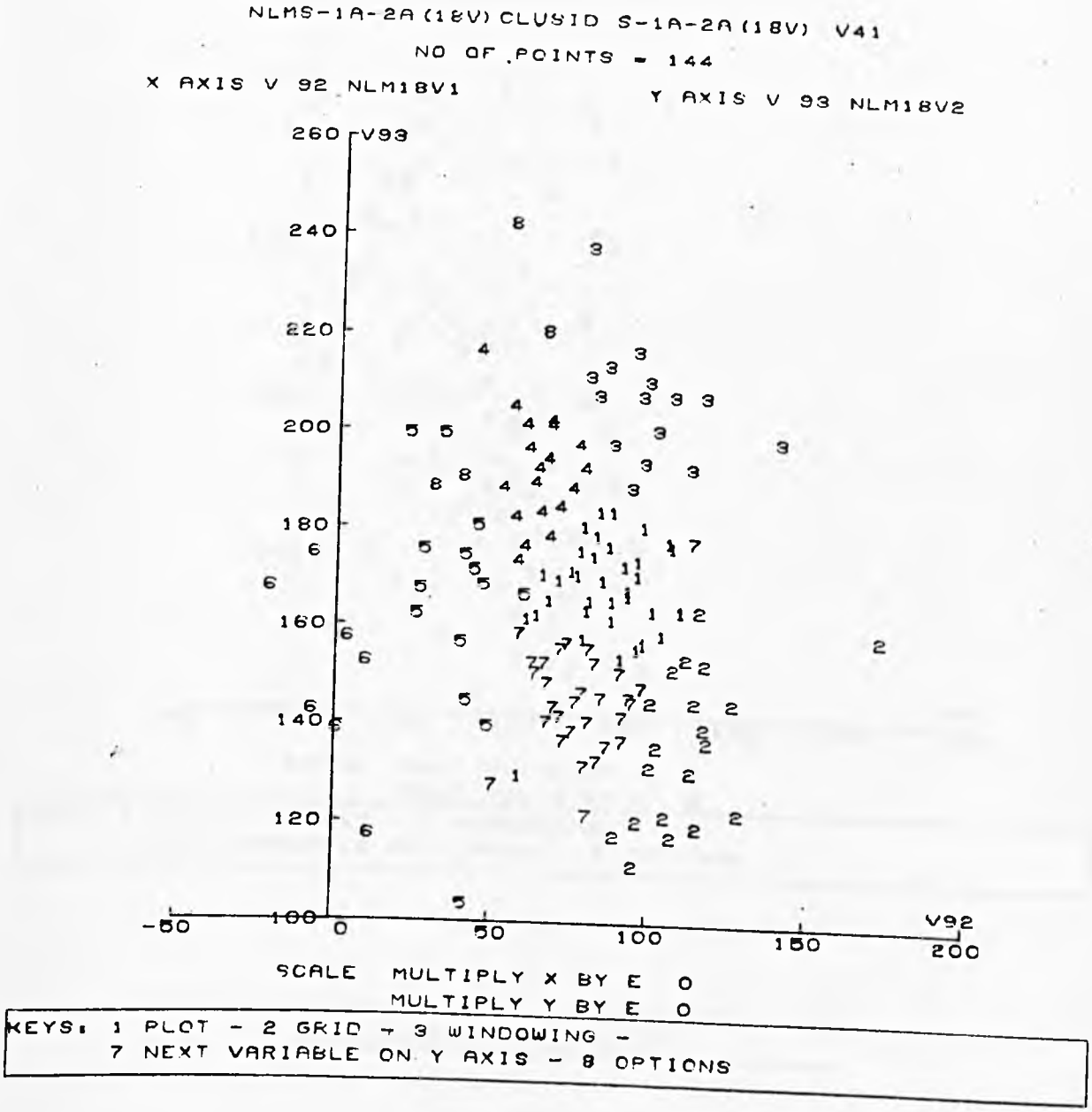


Figure 6.16. Census data: cluster labels set S-IA-IIA, NLM set S-IA-IIA

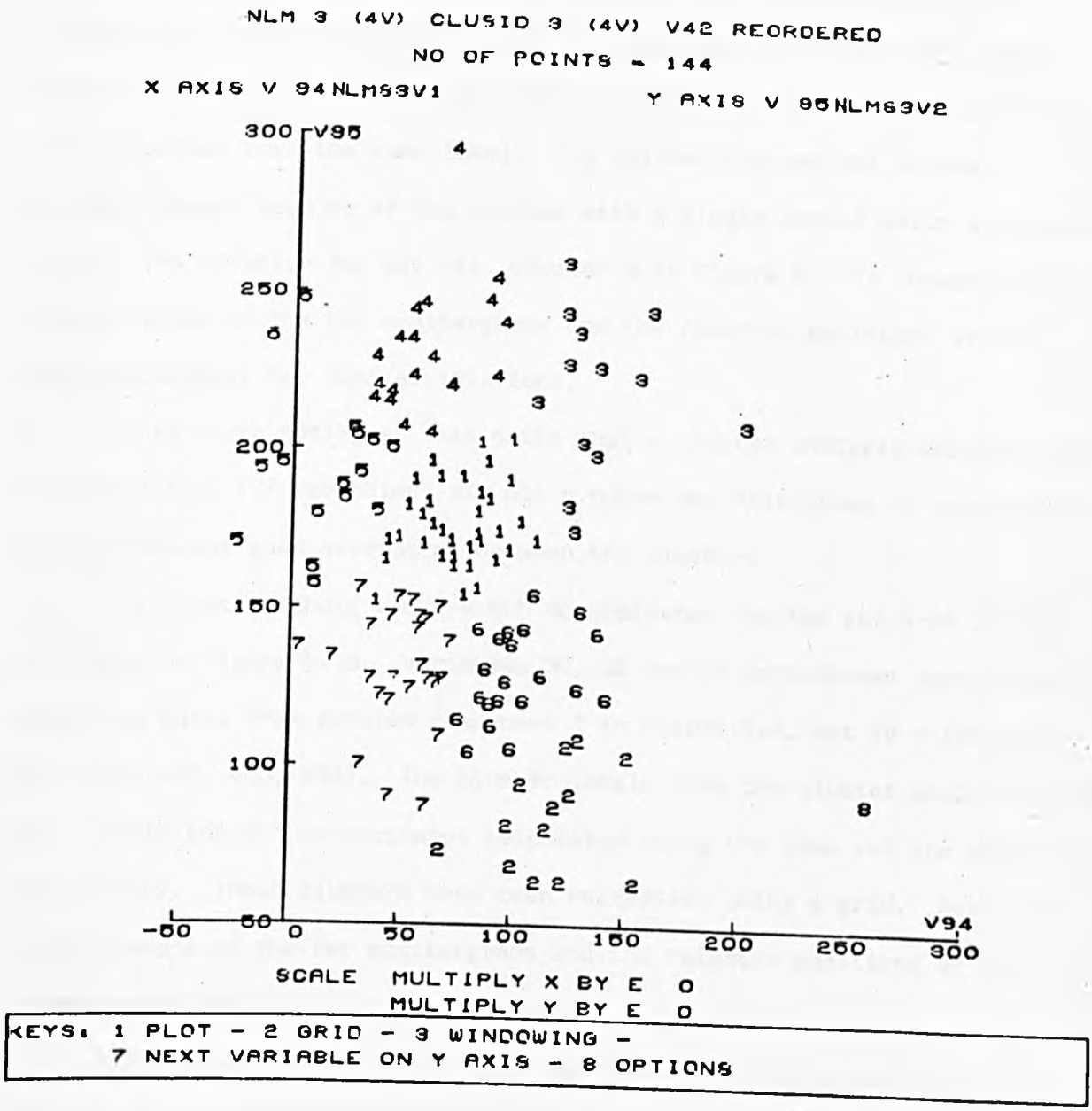


Figure 6.17. Census data: cluster labels set III (relabelled), NLM set III

amplitudes. Therefore set III = {V2, V3, V4, V23} was defined and a cluster analysis using these compared with the cluster analysis using set S-IA-IIA. The cluster labels for the cluster analysis solution using set III and the NLM co-ordinates for set III are shown in Figure 6.17. These clusters have been relabelled so that the clusters which are similar in both solutions have the same label. The relabelling was not entirely straightforward because of the cluster with a single member which occurred only in the solution for set III (cluster 8 in Figure 6.17). However, the overall shape of the two scattergrams and the relative positions of the clusters suggest two similar solutions.

It is worth noting at this point that a cluster analysis solution for the set S-I-II (27 variables) did not produce any histograms or scattergrams where there was good separation between the clusters.

The cluster labels and the NLM co-ordinates for the set S-IA-IIA-III are shown in Figure 6.18. Variables 30, 32 and 33 were chosen using histograms and hence from rotated component 1 in Figure 6.4, set IV = {V6, V11, V22, V30, V32, V33, V34}. The cluster labels from the cluster analysis using set IV with the NLM co-ordinates calculated using the same set are shown in Figure 6.19. These clusters have been relabelled using a grid. Again the overall shape of the two scattergrams and the relative positions of the clusters indicate two similar solutions.

A final set, set V = {V10, V23} was found (rotated component 4). There now only remained five variables; V9, V26, V27, V28 and V29. From inspection of the rotated components in Figure 6.4, it was expected that the remaining variables might come out in three sets; {V9}, {V26, V27} and {V28, V29}. However this did not happen and the next set was {V9, V27, V29}.

This analysis suggests that a given cluster analysis solution is dominated by a subset of the variables used for that analysis, and that the

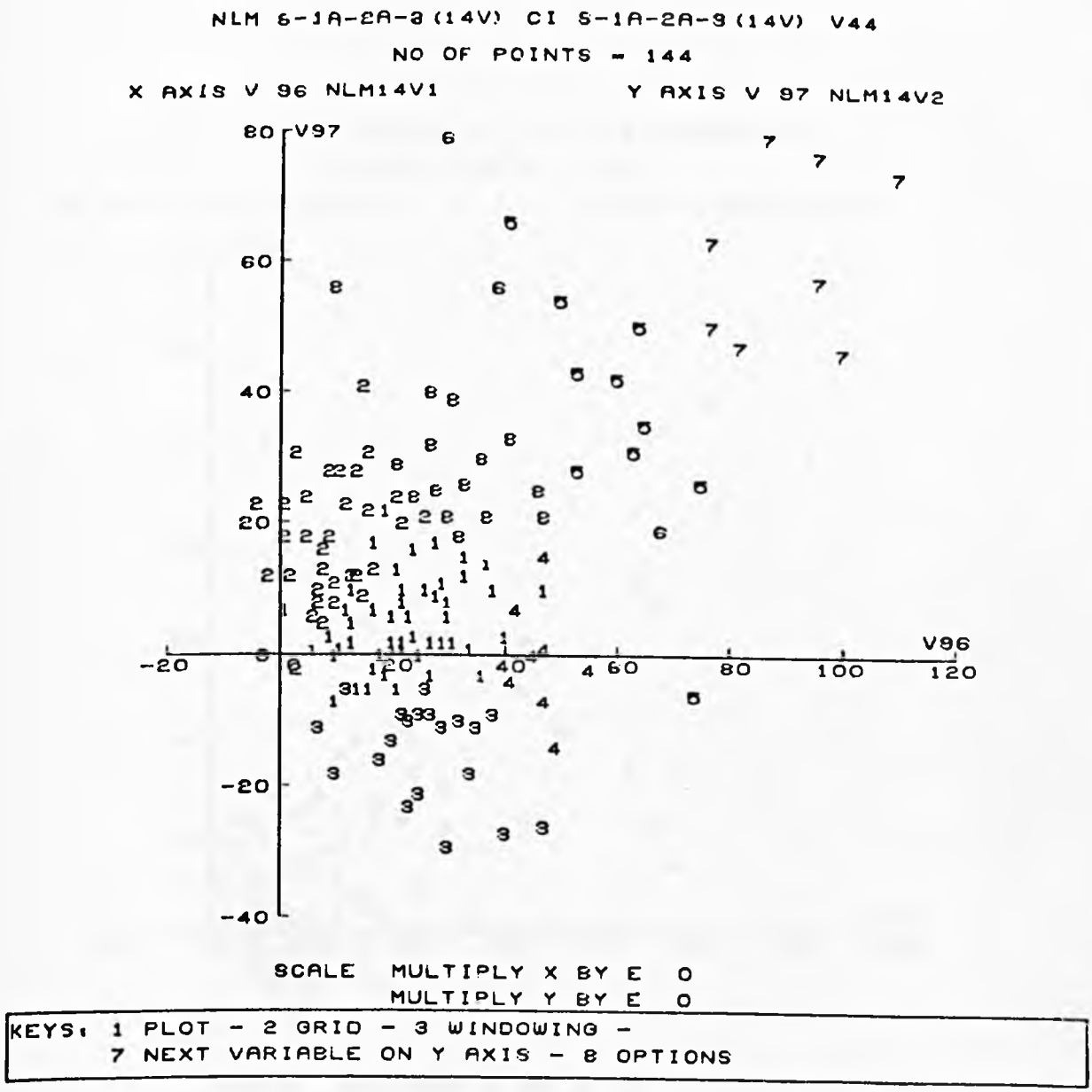


Figure 6.18. Census data: cluster labels set S-IA-IIA-III, NLM set S-IA-IIA-III

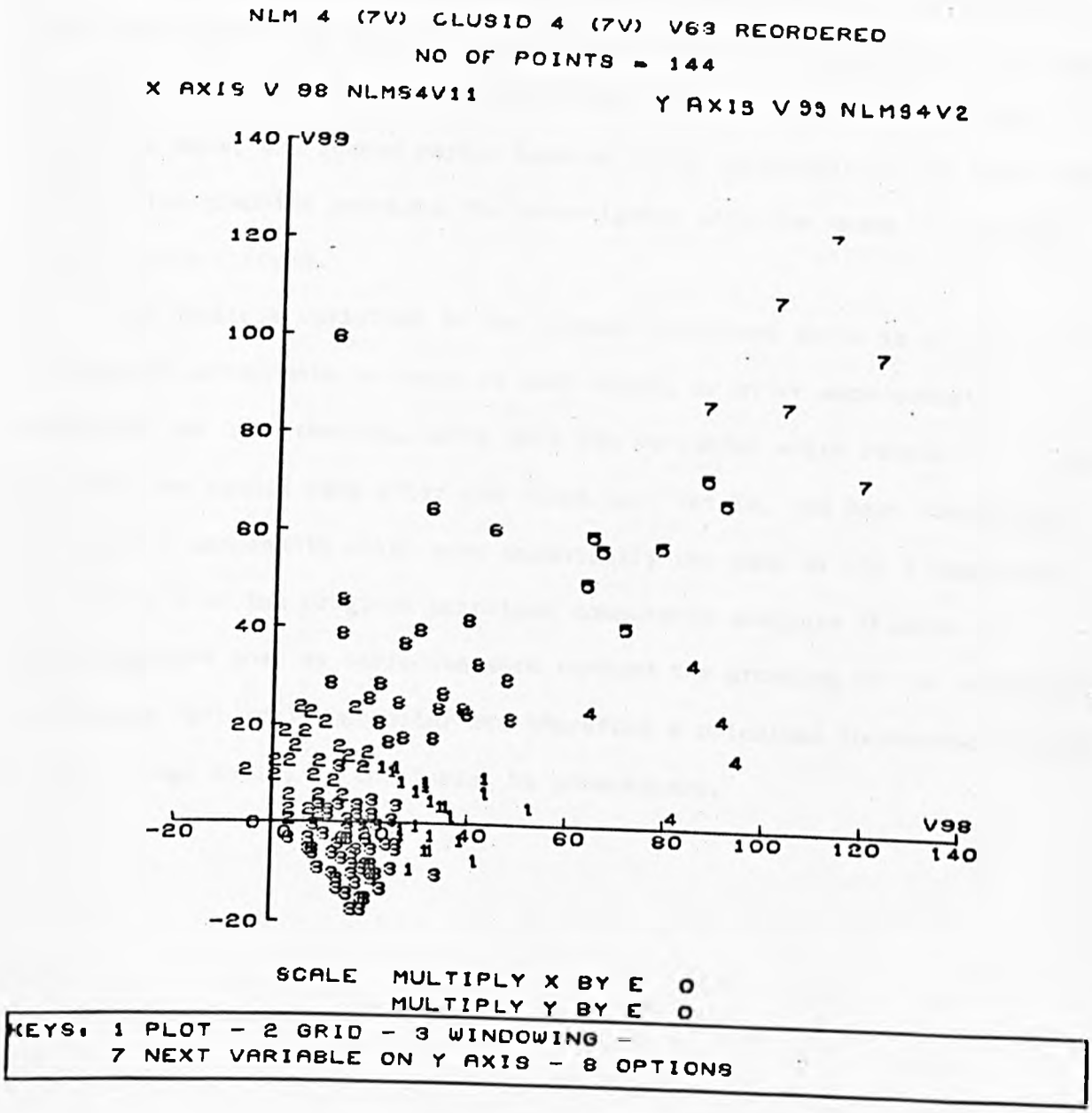


Figure 6.19. Census data: cluster labels set IV (relabelled), NLM set IV

solution is approximately determined by this subset. It also suggests, that for a given set of data it would seem necessary to identify the variables which are contributing to a cluster analysis solution, to remove these and to find alternative clusterings at lower levels in order to discover what happens as sets of variables are successively removed. Although it is impossible to lay down hard and fast rules as to how the analysis should be done, and indeed partly because it is impossible to lay down rules, interactive graphics provides the investigator with the means of carrying the analysis through.

One possible variation in the process described above is to carry out a principal components analysis at each stage, or after each subset of variables has been removed, using only the variables which remain. This was done for the census data after the first set, set IA, had been removed and this gave 7 components which were essentially the same as the 7 components 1-5, 7 and 8 of the original principal components analysis (Figure 6.4). This suggested that as variables were removed the grouping of the variables which were left would not alter and therefore a principal components analysis at each stage would, in this case, be unnecessary.

6.4. The program for discriminant analysis in practice

As a final step in the study of the census data the results of the initial cluster analysis solution were input to the program for discriminant analysis.

The cluster labels for the initial cluster analysis solution were used to define eight groups for input to the procedures for discriminant analysis. All 37 variables were used and the eight groups had 22, 8, 12, 28, 17, 28, 18 and 11 members respectively. The data was standardised so that, over all the data, each variable had zero mean and unit variance. All the observations used to determine the discriminant functions were correctly assigned to their original groups and therefore all the off-diagonal elements of the contingency table were zero.

The means of the groups displayed in terms of the first two canonical variates are shown in Figure 6.20. The two eigenvalues associated with these two vectors are 41.6388 and 4.9883 accounting for 92.93% of the total dispersion. In Figure 6.21 the data points have been superimposed on the diagram (as described in section 4.5.3). The populations 6, 2, 7, 3 and 8 are well separated on the first axis, but it requires the second axis to see clearly the separation between 1, 4 and 5. The histograms for each of these variates are shown in Figure 6.22.

Figure 6.21 may be compared with Figure 6.5 where the same clusters are shown with the same labels but in terms of non-linear mapping co-ordinates. It is interesting to note that the clusters have the same relative positions and while some clusters are more compact and circular in shape in Figure 6.21, e.g. clusters 6 and 4, others, e.g. 7 and 1 have a similar shape in both diagrams. It must be recognised that, whereas the non-linear mapping is a mapping of all the data points onto two dimensions

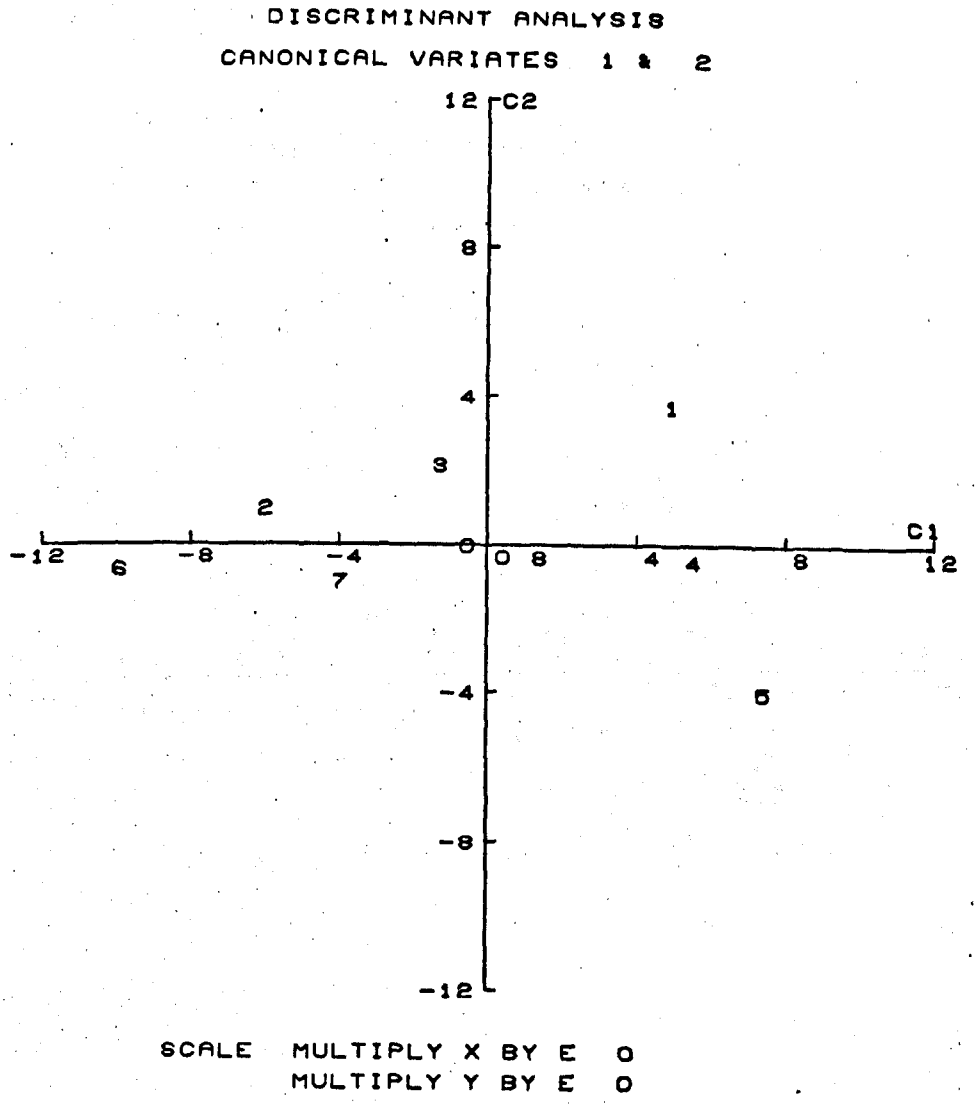


Figure 6.20. Census data: means of clusters (axes - 1st two canonical variates)

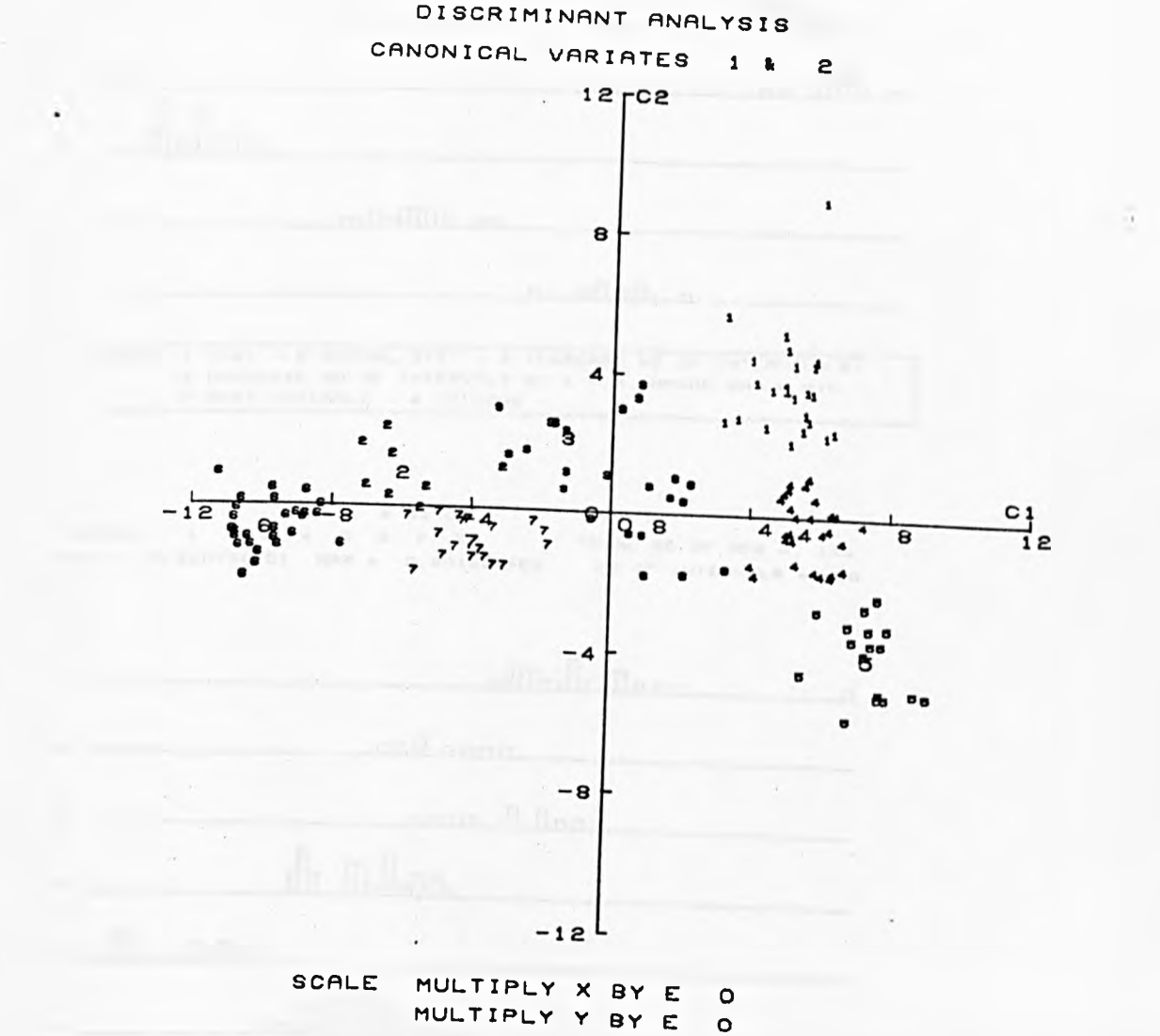
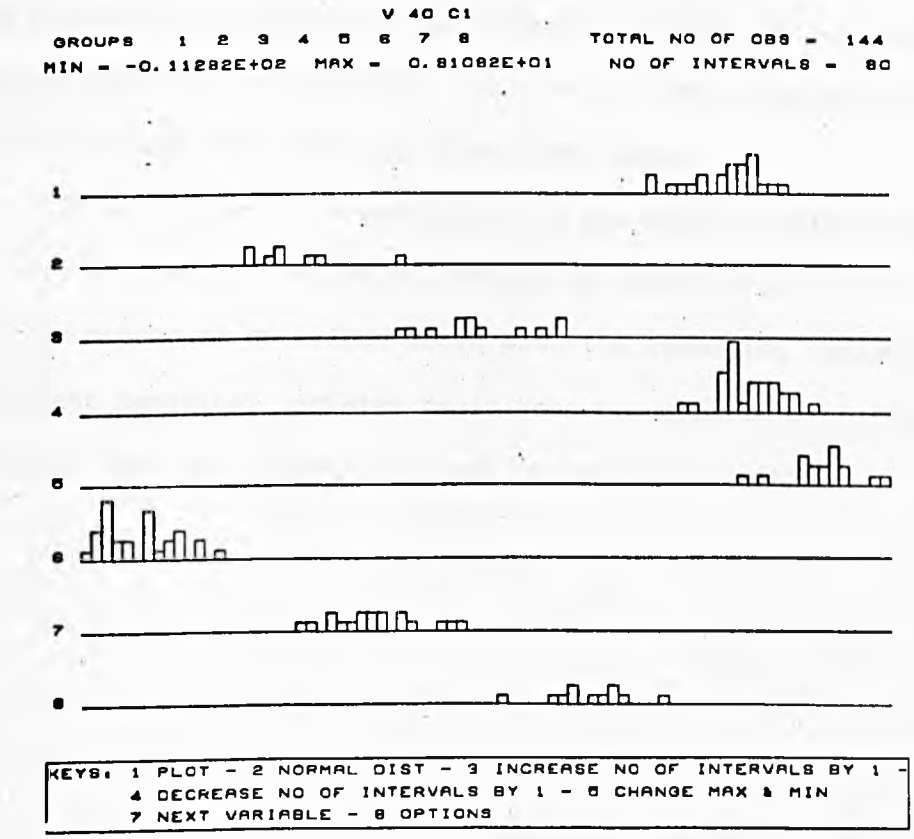


Figure 6.21. Census data: means of clusters and data points (axes - 1st two canonical variates)



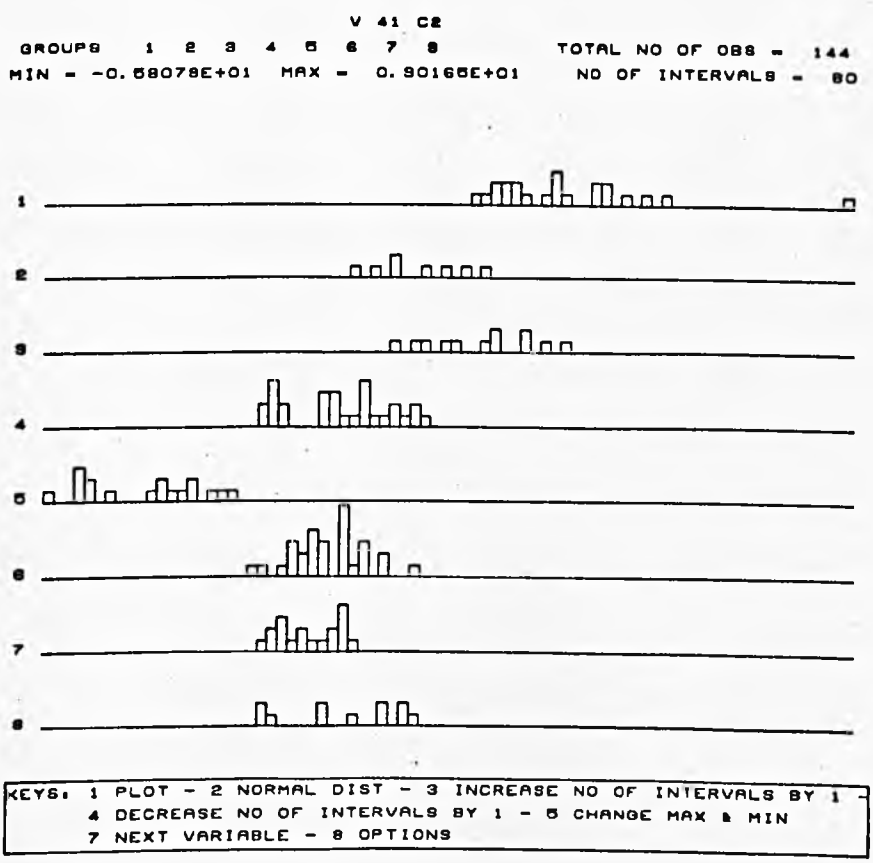


Figure 6.22. Census data: histograms for 1st two canonical variates

and the NLM scattergram gives a complete picture of this mapping, the scattergram with two canonical variates as axes, represents (if $n > 2$ and $k > 3$) a projection onto a 2-dimensional space.

In principle it is possible, and would be of interest to remove each of the five sets of variables defined in section 6.3 in turn and to carry out the procedure described above with the remaining variables. The resultant canonical variates could then be compared with the appropriate NLM scattergrams, however this was not proceeded with.

7. TO WHAT EXTENT IS THIS SYSTEM EFFECTIVE?

Some criteria for an effective system

The originality of this thesis is not in the statistical analyses nor in the use of interactive graphics, but in the combination of these to form an effective system for interactive statistical data analysis. To be effective such a system has to be capable of solving a wide range of problems of varying sophistication, within an acceptable time; it must also be simple to use. In order to solve a range of problems the system must be flexible. If there are no prescribed rules for an analytical procedure, it is necessary to be able to adapt the line of analysis as results are displayed. Numerical and graphical output suggest changes of strategy which can be rapidly implemented in a comprehensive interactive system. Flexibility must be provided by the availability of a variety of relevant procedures, by the provision of different forms of graphical output and a range of possible user interaction with these displays, in addition to the provision of numerical results. The system should also be capable of expansion, without necessarily increasing its complexity, to take account of new ideas generated by use of the programs.

The requirement that a system is flexible often competes with the requirement of simplicity; the more flexible and general the system, the more complex the information which has to be provided to obtain the required facilities. In designing an interactive system (as in designing any system) a suitable balance between these requirements must be achieved.

There are two areas where simplicity of use is important; firstly, the way in which programs function, and the syntax of user-supplied commands, should be straightforward and consistent. Secondly, the means of conveying information to the program (pressing keys, switches, using

the light pen etc.) should make few demands on the user, whose primary concern should be to solve the problem under investigation rather than dealing with the idiosyncracies of a convoluted system.

Some ways in which this system is effective

Various sets of data have been used to illustrate and highlight different features of this interactive system. The final set, the census data, shows in several different respects the degree to which this system is effective: effective, that is, in bringing together the available forms of analysis required to obtain a satisfactory and complete solution.

The range of available multivariate procedures with suitable graphical output was sufficient to gain some insight into the structure of this data set. For instance it has been shown how sets of variables can be identified, sets of variables which appear to influence one another and which appear to have little influence on variables outside that set. Also it was possible to identify outliers and to find for which sets of variables these particular observations had extreme values. The distribution of data along each axis of an n-dimensional space is informative, as are the distributions along new derived axes.

It was not possible, nor practical, to decide at the outset exactly which statistical procedures would be implemented and which forms of associated graphical output provided. However, as the need for new procedures became apparent, partly as a result of the early stages of the census study, the addition of these new procedures and the provision of alternative forms of display was reasonably straightforward. The complementary nature of many of the procedures used, in that the output from one procedure is used as input to another, has meant that the smooth transference from one procedure to the next and the efficient filing of results for reinput has been of some importance operationally.

Many of the results displayed have helped users to gain some intuitive understanding of the statistical techniques, for example, the orthogonal rotation of components with respect to the variables, and the nature of discriminant functions and their relationship to the group means.

How instructive and elucidatory are the particular displays provided? The simple pictures, the histograms and scattergrams, have proved informative for the study of structures within complex data sets, provided that these pictures are examined systematically and in relation to other parts of the analysis. The facility for displaying related histograms simultaneously on the screen for comparison is particularly valuable in this context.

Some aspects in which the system may be made more effective

The range of procedures and the nature and use which can be made of the graphical output provided, are, from the analytical point of view, the main advantages of the system. Experience in using it has demonstrated that more emphasis should be placed on ease of use. Certain aspects of the design were constrained by the 4130 graphics system. A different approach would be made with a more up-to-date computing system, notably in the structure of programs and in the use of files. The provision of more fluent user interaction with the displays is an area for possible refinement. More effective use could be made of the light pen to indicate choice of statistical procedure, choice of options, selection of variables and transformations. Facilities could be provided to extend and increase the flexibility of the presentation of histograms. A range of bivariate and univariate statistics could be provided and graphically identified at a user request.

More emphasis might be placed on making intermediate results more accessible for review. It is important in some analytical procedures to be able to look back over pictures just displayed to assess and compare them with that currently displayed. The improved understanding of what has caused the particular sequence of pictures, which this facility provides, means that processes are more likely to run to completion and to do so economically.

The system described here has been implemented with a refreshed display. A similar adequate system could, for most purposes, be implemented using a cheaper device, given that appropriate forms of graphical interaction are available. The system does not require high speed refreshed graphics per se, nor does it necessarily require the quality of resolution obtained with refreshed displays. More than one screen per user, and an efficient means of recalling and interacting with results previously obtained, would be advantageous for comparison of results.

The nature of this system is experimental and for practical reasons only a limited number of statistical techniques were included. There is scope for introducing additional statistical techniques and providing alternative respresentations of the results and of the data. However, the system as it stands, and the techniques and representations provided, have demonstrated how interactive computer graphics can be made a useful tool for multivariate statistical research.

BLANK PAGE IN ORIGINAL

APPENDIX 1

Rules of syntax for option commands

Each of the syntactic elements which may appear in a list of items in an option command is defined below. Also defined for use in option commands are lists of identifiers, arithmetic expressions, assignment statements and logical expressions.

1. Integers, numbers and identifiers

<null string>::= ␣|<null string>

<digit>::= Ø|1|2|3|4|5|6|7|8|9

<letter>::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

<unsigned integer>::= <digit>|<unsigned integer><digit>

<list of unsigned integers>::= <unsigned integer>|<list of unsigned integers>,
<unsigned integer>

<integer>::= <unsigned integer>|-<unsigned integer>

<list of integers>::= <integer>|<list of integers>,<integer>

<decimal fraction>::= .<unsigned integer>

<unsigned real>::= <unsigned integer>|<decimal fraction>|
<unsigned integer><decimal fraction>

<number>::= <unsigned real>|-<unsigned real>

Examples:	Ø	.45
	123	1.45
	-45	-3.72

<identifier>::= <letter>|<identifier><letter>|<identifier><digit>

Only the first 8 characters are significant.

Examples:	AGE
	AGE1

<separator>::= **|*|/|+|-|(|)|:|=|,|;

<item>::=<separator>|<separator><null string>|<identifier><null string>|
<number>|<number><null string>

<list of items>::= <item>|<list of items><item>

2. Variable identifiers

<user variable name>::= <identifier>

<list of user variable names>::= <user variable name>|
<list of user variable names>,
<user variable name>

<system variable name>::= V<unsigned integer>

Examples: V1
V1Ø
V1ØØ

<variable identifier>::= <user variable name>|<system variable name>

<sublist of variable identifiers>::= <variable identifier>|
<variable identifier>
␣TO␣<variable identifier>

<list of variable identifiers>::= <sublist of variable identifiers>|
<list of variable identifiers>,
<sublist of variable identifiers>

Examples: AGE, ANS1 TO ANS5
V1, V11 TO V15

3. Record identifiers

<record identifier>::= R<unsigned integer>

<sublist of record identifiers>::= <record identifier>|<record identifier>
␣TO␣<record identifier>

<list of record identifiers>::= <sublist of record identifiers>|
<list of record identifiers>,
<sublist of record identifiers>

Example: R1 TO R5Ø, R1Ø1 TO R15Ø

4. Arithmetic expressions and assignment statements

<adding operator>::= +|-

<multiplying operator>::= *|/

<function name>::= ABS|AINT|ALOG1Ø|ATAN|COS|EXP|SIN|SQRT

<function reference>::= <function name>(<arithmetic expression>)

<primary>::= <unsigned integer>|<variable identifier>|<function reference>|
(<arithmetic expression>)

<factor>::= <primary>|<factor>**<primary>

<term>::= <factor>|<term><multiplying operator><factor>

<arithmetic expression>::= <term>|-<term>|<arithmetic expression>
<adding operator><term>

<assignment statement>::= <variable identifier>=<arithmetic expression>

Notes on the evaluation of arithmetic expressions:

1) The following rules of precedence hold

1st	**	exponentiation
2nd	* /	multiplication and division
3rd	+ -	addition and subtraction

2) Operators of equal precedence are evaluated from left to right.

3) Arithmetic expressions enclosed in brackets are evaluated first.

4) All calculations are in floating point arithmetic.

5) The function references produce the same results as the (ANSI) Fortran intrinsic functions and basic external functions of the same name.

Examples:
```
V1 = AINT(V1)
V5 = (V3 + V2)**2
AGEM = AGE1 + 12 * AGE2
R = (A * B)/C
```

5. Logical expressions

<alphanumeric characters>::= +|-|/|*|,|.|(|)|␣|:|<letter>|<digit>

<alphanumeric string>::= <alphanumeric character>|<alphanumeric string>
<alphanumeric character>

Only the first 8 alphanumeric characters are significant.

<alphanumeric constant>::= :<alphanumeric string>:

<logical operator>::= AND|OR

<relational operator>::= GT|LT|EQ|LE|GE|NE

<relational operand>::= <arithmetic expression>|<alphanumeric constant>

<relational expression>::= <relational operand>␣<relational operator>␣
<relational operand>

A relational expression has the value true or false.

<logical expression>::= <relational expression>|<logical expression>␣
<logical operator>␣<relational expression>

A logical expression has the value true or false.

Examples: V1 EQ :I:

V3 EQ 2*V1 OR V3 EQ 2*V1

APPENDIX 2

User's Manual

The User's Manual is reproduced here in its entirety to give details of how the individual programs may be used. Parts of this manual necessarily repeat some aspects of the system which have already been described in the main text of the thesis.

AN INTERACTIVE GRAPHICS SYSTEM FOR MULTIVARIATE STATISTICS

USER'S MANUAL

GRAPHICAL STATISTICS

1. INTRODUCTION

A set of individual programs written in Fortran, for the ICL 4130 to perform statistical analyses interactively and to present the results on the Graphical Display. Both alphanumeric data and diagrams can be displayed. When using these programs the user can communicate with the machine by typing commands on the console, by pressing the keys beneath the screen and by using the light pen to indicate points on the screen.

1.1. SUMMARY OF PROGRAMS CURRENTLY AVAILABLE

1.1.1. ADMINISTRATIVE PROGRAMS

RDDATA reads in the data from an initial input file, e.g. cards, disc file, etc. This data will consist of a set of readings (referred to as records or observations) on a set of variables. Variables may be given names, and observations may be identified by their position in the initial input file. RDDATA is run at the start of each session.

FILREC is for the transformation and selection of data, it can also be used to add variables to the data file.

1.1.2. DISPLAY PROGRAM

DISVAR displays data either as histograms or 2-dimensional scattergrams. Data for DISVAR may be the original input data, or output derived from other programs described below, e.g. factor analysis or principal components.

1.1.3. CORRELATION MATRIX PROGRAM

CORR calculates a correlation matrix which may be input to a program for factor analysis and principal components.

1.1.4. FACTOR ANALYSIS AND PRINCIPAL COMPONENTS PROGRAM

FACTOR is for factor analysis and principal components and takes as input a correlation matrix.

1.1.5. NON-HIERARCHICAL AND HIERARCHICAL CLUSTERING

The non-hierarchical clustering program, EUCLID, attempts to find the best set of k clusters, such that the sum of squares of deviations of all elements from their cluster centres is a minimum. All the displays in this program are 2-dimensional scattergrams.

Distance coefficients are calculated for hierarchical clustering by a program DISTANCE.

The hierarchical clustering program CLUSTER has four hierarchical clustering methods; nearest neighbour, furthest neighbour, weighted and unweighted mean pair. The results of these methods can be displayed as dendrograms and scattergrams.

1.1.6. DISCRIMINANT ANALYSIS

DISCRIM is a program for discriminant analysis.

2. GENERAL INSTRUCTIONS FOR THE USE OF PROGRAMS

2.1. TO LOAD A PROGRAM

Each program is loaded by submitting the following cards.

```
&JOB;<ccno>;
&OPTIONS;FIORM,FIORD,DSEG;
&LOAD;<progname>,  CCØ6P1;DC;6;FORTRAN;
&RUN;
     <data if any>
&END;
```

Where <progname> is one of the following

```
RDDATA
FILREC
DISVAR
CORR
FACTOR
EUCLID
DISTANCE
CLUSTER
DISCRIM
```

Any &ASSIGN cards that may be required should be placed before the &RUN; card.

2.2. LISTS OF OPTIONS

All programs, except RDDATA, display a list of options when first loaded. Each item in the list of options represents a command. Figure 1 illustrates the first list of options for CLUSTER.

```
****                CLUSTER ANALYSIS                ****

OPTIONS AVAILABLE AT LEVEL 1.  DEFAULT VALUES IN (),
START EACH COMMAND WITH *) WHERE * = A OR B OR C
FOLLOWED BY REQUIRED INFORMATION IF ANY.
              TERMINATE COMMAND WITH ;

   A)  NEAREST NEIGHBOUR
   B)  FURTHEST NEIGHBOUR
   C)  UNWEIGHTED MEAN PAIR, CENTROID
   D)  WEIGHTED MEAN PAIR
   E)  DISPLAY DENDROGRAM
   F)  DISPLAY SCATTERGRAM
   Z)  EXIT
```

Figure 1

Four flashing asterisks in the top right and left hand corners of the screen indicate that the system is waiting for the user to respond, either as in the example above by typing a command on the console, or in other situations by using keys and in some cases the light pen.

2.3. LEVELS OF OPERATION

The programs function at different levels; different lists of options are displayed for different levels of operation. The first list of options to be displayed is always level 1. Transferring from level I to level I+1 can mean that there are different routes through the program, and a choice of route has been made at level I, or that items can be defined at an earlier level and remain constant at later levels and do not have to be frequently redefined, or that level I+1 is the next stage in the analysis.

2.4. COMMANDS, KEYS AND LIGHT PEN

2.4.1. COMMANDS

Commands are given in the form

<letter>)<list of items>;

where
<letter> is one of the characters which appear at the start of each line on the display. The list of items is optional and whether or not it is given depends on the nature of the command. Definitions of what may appear in the list of items are given for each option. Default values are given in brackets at the end of each command on the screen. They indicate what values each command will take if no action is taken, i.e. if the relevant command is not given. When a list of options is displayed and a command is to be typed, a digit, which corresponds to the level of options currently displayed, is output. Commands must start

<letter>

If they do not, the value of I is output on a new line on the console typewriter and the command must be retyped. The character '%' may be used at any point to enable a command to be restarted.
If a command requires more than one line of the console typewriter, when the end of the line is reached, the user must output a carriage return and line feed and the command may be continued on a new line. CR LF may be used any time.

2.4.2. KEYS

There are 8 switches or keys at the bottom of the screen. This system has been written so that these will operate only when diagrams are displayed, not when lists of options are displayed. Diagrams in this context may be tables of numerical data or line drawings. Instructions on how to use the keys appear under each diagram.

Key 1 is used universally for taking hard copy; on the line printer for numerical data and on the digital plotter for line drawings.

Key 8 is used universally for redisplaying lists of options.

2.4.3. THE LIGHT PEN

The light pen may, in a few circumstances, be used to define points on the screen. It has been programmed to be used with the tracking cross which can be displayed by pressing a key as indicated when a diagram is on the screen.

2.5. EXIT FROM PROGRAMS

A program can be terminated by giving the command

Z);

whenever a list of options is displayed. Other programs can then be loaded.

2.6. DEFINITIONS OF ITEMS THAT APPEAR IN <list of items> IN COMMAND

Definitions that are relevant to one program are deferred till that program is described in detail.

Definitions that will be required for most programs are given within this section.

The symbol '::=' means 'can be defined as' and the symbol '|' means 'or'.

2.6.1. INTEGERS, NUMBERS AND IDENTIFIERS

<null string>::=␣|<null string>␣ (string of blank characters)

<digit>::= 0|1|2|3|4|5|6|7|8|9

<letter>::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

<unsigned integer>::= <digit>|<unsigned integer><digit>

<list of unsigned integers>::= <unsigned integer>|<list of unsigned integers>,
<unsigned integer>

<integer>::= <unsigned integer>|-<unsigned integer>

<list of integers>::= <integer>|<list of integers>,<integer>

<decimal fraction>::= . <unsigned integer>

<unsigned real>::= <unsigned integer>|<decimal fraction>|
<unsigned integer><decimal fraction>

<number>::= <unsigned real>|-<unsigned real>

Examples: 0
1.23
.345
123

<identifier>::= <letter>|<identifier><letter>|<identifier><digit>

Only the first eight characters of an identifier are significant.

Examples: AGE
NAME
V10

<separator>::= **|*|/|+|-|(|)|:|=|;|,

<item>::= <separator>|<separator><null string>|<identifier><null string>|
<number>|<number><null string>

<list of items>::= <item>|<list of items><item>

2.6.2. VARIABLE IDENTIFIERS

Variable identifiers can be given in two forms which are interchangeable:

1) names supplied by the user; these are optional,
2) names automatically supplied by the system.

<user variable name>::= <identifier>

<list of user variable names>::= <user variable name>|<list of user variable names>,<user variable name>

Example: AGE, NAME, INCOME

<system variable name> ::= V<unsigned integer>

The system supplies the unsigned integers sequentially.

Examples: V1
V10
V100

<variable identifier>::= <user variable name>|<system variable name>

In most programs there is an option to choose a set of variables for analysis and a list of variable identifiers must be given.

<sublist of variable identifiers>::= <variable identifier>|<variable identifier> ␣TO␣<variable identifier>

<list of variable identifiers>::= <sublist of variable identifiers>|<list of variable identifiers>,<sublist of variable identifiers>

Example: AGE, ANS1 TO ANS5, INCOME or V1, V11 TO V15, V100

When the 1st variable in the set has user variable name AGE,
the 11th variable in the set has user variable name ANS1,
the 15th variable in the set has user variable name ANS10,
and the 100th variable in the set has user variable name INCOME.

The two lists in this example will be synonymous and the set of variables defined will be the 7 variables 1, 11, 12, 13, 14, 15, 100.

2.6.3. DISPLAY OF VARIABLE IDENTIFIERS

In those programs where a set of variables can be chosen for analysis, a command

<letter>) <list of variable identifiers>;

must be given.

When

<letter>)

is typed the variable identifier and their type ('N' for numeric, 'A' for alphanumeric) will appear on the screen. A maximum of 20 identifiers are displayed at once, further identifiers may be displayed by pressing key 2.

As the list of variable identifiers is typed and variables are included in the set asterisks will appear beside the identifiers on the screen. When the ; terminating the command is typed the list of options for the current level of operation will be redisplayed.

If the default value is to operate either the command should not be given at all, or if the identifiers are displayed as the result of typing

<letter>)

then no list should be given and ; must be typed to redisplay the list of options.

2.6.4. RECORD IDENTIFIERS

Records or observations may be identified by their position in the initial input file.

In some programs there is an option to supply a list of record identifiers to indicate that only those records in the list should be used in the analysis.

<record identifier>::= R<integer>

<sublist of record identifiers>::= <record identifier>|<record identifier>
␣TO␣<record identifier>

<list of record identifiers>::= <sublist of record identifiers>|
<list of record identifiers>,
<sublist of record identifiers>

Example: R1 TO R20, R51 TO R70

The 40 records, 1st - 20th inclusive and the 51st - 70th inclusive, only will be used in the analysis.

3. RDDATA - A PROGRAM TO READ IN THE DATA AND SET UP THE SYSTEM

This program sets up the data file and is unlike all other programs in this system in that nothing appears on the display.

3.1. INPUT FILES

Data may be input from any type of file, e.g. cards, disc file, etc. and RDDATA will create a master input file (m.i.f.) on magnetic tape with this data.

At the start of a session the m.i.f. is the current input file (c.i.f.) for all programs.

If at any time new variables are added or transformations are specified a new file is created on another magnetic tape and this becomes the c.i.f.

The m.i.f. is preserved as the original data file for use in later sessions. The new c.i.f. may also be used at a later session. RDDATA must be run at the start of a session either to create a m.i.f. or to establish and check a c.i.f. from an earlier session.

The program is loaded using the appropriate deck of cards and the message

CHANNEL NO FOR DATA

is output on the console.

If a m.i.f. is to be created, there must be a magnetic tape on handler 0 and the user replies to the message by typing

37(CR)(LF)

which assumes that the data is on cards. If the data is on another peripheral, channel 37 should be reassigned with the appropriate &ASSIGN card. If an existing file, a c.i.f., created during an earlier session, is available the reply to the message is

40(CR)(LF)

and the relevant magnetic tape should be loaded on handler 0. The program will then type the message

CHANNEL NO FOR COMMANDS

and the reply must be

1 (CR)(LF)

to indicate that the commands are coming from the console.

3.2. FORMAT FOR INITIAL INPUT FILE

The first part of this file contains specifications and the second part the data.

A specification is defined as a keyword of a maximum of 8 characters which start in column 1, and the relevant information is in columns 9 - 80.

3.2.1. TITLE CARD. Optional

cols. 1 - 5 TITLE

cols. 9 - 80 Any alphanumeric title

3.2.2. NO. OF VARIABLES CARD. Compulsory

cols. 1 - 4 NVAR

cols. 9 - 80 <integer>
<integer> is the no. of variables in the initial input file (max. 512)

3.2.3. NO. OF OBSERVATIONS CARD. Compulsory

cols. 1 - 4 NOBS

cols. 9 - 80 <integer>
<integer> is the no. of observations in the initial input file, or the number of records to be read; one record may require more than one card.

3.2.4. FORMAT CARD(S). Optional

Default: If no format card is supplied a format of (16F5.0) is assumed. This means that there may be any number of cards in the format (16F5.0) for each record.

If a variable is read under E or F format it has type 'N'.
If a variable is read under A format it has type 'A'.

cols. 1 - 6 FORMAT

cols. 9 - 80 Format for data in the usual Fortran form starting with (etc. Items in this format statement must be either real, i.e. E or F, or alphanumeric, i.e. A. If more than one card is required for the format specification any following cards should have blank in cols. 1 - 8 and the specification may be continued starting in column 9.

3.2.5. VARIABLE NAMES CARD(S). Optional

Default: If no variable names cards are supplied, variables may only be referred to by their system variable name V1, V2, V3, etc.

cols. 1 - 6 VNAMES

9 - 80 <list of user variable names>;
Continuation cards may be used, with blank in cols. 1 - 8 and the list of user variable names continuing in col. 9. Names must not be split over two cards.

The first variable on the initial input file will get the first name in the list and so on; this will include variables that are read in under A format.

3.2.6. MISSING DATA CODES CARD(S). Optional

Default: If no missing data codes cards are supplied, no missing data codes will be recorded.

The following definitions are required. (m.d.c. means missing data code)

<m.d.c.>::= <number>

<m.d.c. spec>::= <user variable name>,<number>

<list of m.d.c. specs>::= <m.d.c. spec>|<list of m.d.c. specs>/<m.d.c. spec>

Example: HEIGHT,999/REPLY,99

The format for the missing data code cards is as follows:

cols. 1 - 7 MDCODES

cols. 9 - 80 <list of m.d.c. specs>;
Continuation cards may be used as for VNAMES card, but neither names nor codes may be split over two cards.

9.2.7. DATA CARD. Compulsory

cols. 1 - 4 DATA

This card introduces the data and must be the last card in the specification section.

The specifications may appear in any order. Note that only three cards are compulsory, therefore the simplest specification that may be supplied is

NOBS <integer>

NVAR <integer>

DATA

followed by the data itself.

Exit from RDDATA is automatic, and an 'A' should appear on the console to indicate that it has run correctly. If any errors do occur, error messages are output to the line printer indicating which specification cards have caused the errors.

4. FILREC - A PROGRAM FOR TRANSFORMATIONS AND SELECTION OF DATA FOR INPUT TO OTHER PROGRAMS

4.1. OPTIONS LEVEL 1

A) TYPE ASSIGNMENT STATEMENTS FOR TRANSFORMATIONS

B) TYPE CONDITIONAL ASSIGNMENT STATEMENTS

C) TYPE SELECTIVE IF STATEMENTS TO SELECT OBSERVATIONS FOR NEW FILE

D) TYPE LIST OF RECORD IDENTIFIERS OF RECORD TO BE INCLUDED IN NEW FILE, MAX 150

E) TYPE LIST OF IDENTIFIERS OF NEW VARIABLES TO BE ADDED TO FILE

F) DISPLAY VARIABLE NAMES FOR SELECTION FOR FILING

X) START FILING AND WHEN FILING IS FINISHED EXIT

Interpretation of options level 1

A) <assignment statement>;

<assignment statement>::= <variable identifier>=<arithmetic expression>

The assignment statement may be for the definition of a new variable, or for the redefinition of an existing one, therefore the variable identifier on the LHS of the assignment statement may be an existing identifier or a new one.

<arithmetic expression> is defined as follows:

<adding operator>::= +|-

<multiplying operator>::= *|/

<function name>::= ABS|AINT|ALOG|ALOG10|ATAN|COS|EXP|SIN|SQRT

<function reference>::= <function name>(<arithmetic expression>)

<primary>::= <unsigned integer>|<variable identifier>|<function reference>|(<arithmetic expression>)

<factor>::= <primary>|<factor>**<primary>

<term>::= <factor>|<term><multiplying operator><factor>

<arithmetic expression>::= <term>|-<term>|<arithmetic expression><adding operator><term>

Floating point arithmetic is used for all calculations.

Examples: A)V1 = AINT(V1);

A)V5 = V3 - V2;

A)AGEM = AGE1 + 12 * AGE2;

Any number of these assignment statements or commands starting B) or C) may be given until the message

NO ROOM FOR FURTHER COMMANDS A) B) or C), PROCEED TO FILING

is output on console.

If an expression is too complex the message

STATEMENT TOO COMPLEX

is output on console. The current command will be ignored and further commands may be typed.

Neither these assignment statements nor conditional assignment statements B) or selective statements C) will be executed until command X); is given when they will be executed for every record on the c.i.f.

B) IF <logical expression>,<assignment statement>;

<logical expression> is defined as follows:

<alphanumeric character>::= +|-|/|*|,|.|(|)|:|<letter>|<digit>

<alphanumeric string>::= <alphanumeric character>|<alphanumeric string>
<alphanumeric character>
(maximum of 8 alphanumeric characters)

<alphanumeric constant>::= :<alphanumeric string>:

<logical operator>::= AND|OR

<relational operator>::= GT|LT|EQ|LE|GE|NE

<relational operand>::= <arithmetic expression>|<alphanumeric constant>

<relational expression>::= <relational operand>␣<relational operator>␣
<relational operand>

A relational expression has value true or false.

<logical expression>::= <relational expression>|<logical expression>␣
<logical operator>␣<relational expression>

A logical expression has value true or false. Therefore when command X); is given, for each record on the c.i.f., if the logical expression is true the assignment statement which follows the comma is executed. If the logical expression is false the assignment statement is not executed.

Examples: B)IF V1 EQ 3, V4=5;

B)IF V1 EQ 3 OR V10 EQ :H:,V11=5;

Error messages are as for A)

C) IF <logical expression>;

When command X); is given the logical expression is evaluated for each record on the c.i.f.

If the logical expression is true the record is included on the new c.i.f. If it is not true the record is not included.

Examples: C)IF V1 EQ :I:;

C)IF V3 EQ 4 OR V3 EQ 6;

Error messages are as for A)

D) <list of record identifier>;

Only those records (max. 150) specified in the list may be output to new c.i.f.

If commands A) or B) or C) have been given, then these commands will only operate on records whose identifiers appear in this list.

E) <list of user variable names>;

This list must contain user variable names which have not already been specified. These names are for new variables which are to be added to the new c.i.f., and which are in a file or channel 33 (paper tape reader, unless channel 33 is reassigned).

Format for file: for each record on channel 33

cols. 1 - 5	integer part of record identifier	I5
cols. 6 - 10	data for 1st variable in list	100F10.0
cols. 16 - 20	data for 2nd variable in list	
etc.		

F) <list of variable identifiers>:

Default: All variables on c.i.f., all new variables defined by assignment statements and all variables defined with command E) will be output to new c.i.f.

Only those variables specified in the list will be output to the new c.i.f.

X);

If c.i.f. is on handler 0, new c.i.f. on handler 3.
If c.i.f. is on handler 3, new c.i.f. on handler 4 and vice-versa.

Program is terminated when filing is completed, new c.i.f. will be used for subsequent programs.

5. DISVAR - A PROGRAM TO DISPLAY THE DATA AS HISTOGRAMS AND 2-DIMENSIONAL SCATTERGRAMS

5.1. OPTIONS LEVEL 1

A) DISPLAY VAR NAMES TO SPECIFY THOSE WHICH MAY BE USED FOR AXES - (ALL)

B) TYPE LIST OF RECORD IDENTIFIERS, MAX. 150 - (ALL)

C) DEFINE GROUPS WITH LOGICAL IF STATEMENTS

X) TRANSFER TO LEVEL 2

Z) EXIT

In this program between level 1 and level 2 data is transferred from the c.i.f. to a workfile on disc for quick access and to enable reasonably fast display of diagrams.

Interpretation of options level 1

A) <list of variable identifiers>;

Default: All variables which have type 'N' will be transferred to the workfile.

Only those variables specified in the list of variable identifiers will be copied to the workfile.

B) <list of record identifiers>;

Default: All records on the c.i.f. will be transferred to the workfile when the program transfers to level 2.

Only those records (max. 150) specified in the list of record identifiers will be transferred to the workfile.

C) IF <logical expression>;

Default: If no commands of this type are given, no groups will be defined.

The logical expression in this command is as defined in 4.1.

Each command of this type defines a group, the first such command defining group 1, etc.

For each record on the workfile, if the logical expression for group J is true, then that record belongs to group J. A maximum of 10 commands of this type may be given.

X);

Data will be copied from the c.i.f. to a workfile, when this is complete the list of options for level 2 will be displayed.

Z);

Exit.

5.2. OPTIONS LEVEL 2

A) TYPE IN VARIABLE IDENTIFIERS FOR AXES
1 ONLY FOR HISTOGRAM, 2 FOR SCATTERGRAM

B) TYPE IN GROUP NOS OF GROUPS TO BE DISPLAYED - (ALL GROUPS)

C) LARGE CHARACTERS FOR DATA POINTS - (SMALL CHARACTERS)

D) TYPE TITLE FOR DISPLAY, MAX. 40 CHS - (NO TITLE)

E) TYPE LIST OF RECORD IDENTIFIERS TO BE DISPLAYED AND INDIVIDUALLY IDENTIFIED, MAX. 50 - (ALL)

F) IGNORE ANY GROUP DEFINITIONS, DISPLAY ALL DATA -
(DATA DISPLAYED AS GROUPS IF DEFINED AT LEVEL 1)

X) DISPLAY DIAGRAM

Y) RETURN TO LEVEL 1

Z) EXIT

5,2.1. INTERPRETATION OF OPTIONS LEVEL 2 FOR HISTOGRAMS

A) <variable identifier>;

<variable identifier> gives the variable for the histogram.

B) <list of unsigned integers>;

Default: If groups have not been defined all the records on the workfile will be included in the histogram. If groups have been defined all the groups will be displayed, each one as a separate histogram. The group definitions will be evaluated in the order the commands C) were given at level 1 for each record on the workfile. As soon as a record is found to belong to a group it will be included in the data for the relevant histogram.

The <list of unsigned integers> defines which groups are to be displayed as histograms. The histograms will be displayed one above the other, up to a maximum of 10.

C);

Not relevant for histograms.

D) any alphanumeric title ;

Default: No title.

Title, (max. 40 characters) displayed above histogram.

E);

Not relevant for histograms.

F);

Default: If group definitions have not been given at level 1 this command is irrelevant.
If group definitions have been given at level 1 one histogram will be displayed for each group.

F); Cont.

Any group definitions given at level 1 will be ignored, and one histogram will be displayed.

X);

Display histograms.

Y);

Return to level 1.

Z);

Exit.

5.2.2. DISPLAY OF HISTOGRAM

The following are displayed above the histograms.

1. Title, if any.
2. Variable identifier.
3. Groups for which histograms are displayed if any commands C) were given at level 1.
4. No. of records used for all histograms.
5. No. of intervals for each histogram, initially 10 times the number of groups.
6. Minimum and maximum values of data for all histograms.
7. If only one histogram is displayed, mean and standard deviation of the data used for that histogram.

If there is only one histogram, the height of the histogram is such that a normal distribution curve can be superimposed on the picture. For more than one histogram the height is determined by the interval with the largest number of entries.

Keys 1: PLOT

2: NORMAL DIST.
For one histogram, first time key is pressed normal distribution curve will be displayed superimposed on histogram. Pressing key 2 subsequently will alternately delete and redisplay the curve.

For more than one histogram, one curve will be displayed for each histogram superimposed on one another.

Initially the height of the curves is determined by the height of the curve (not displayed) which represents all the data. The histograms are not displayed simultaneously, and the keys function as follows when the curves are displayed.

Keys 1: PLOT

2: RESTORE HISTOGRAM

Keys 6: CHANGE SCALE
First time key 6 is pressed, curves will be shown with the maximum height available on the screen.

Pressing key 6 subsequently will alternately display the curves with their initial and maximum heights.

8: OPTIONS
Options level 2.

Keys for histogram continued.

Keys 3: INCREASE NO OF INTERVALS BY 1
No. of intervals for histograms will be increased by 1 (max. 100 inte[illegible]).

4: DECREASE NO OF INTERVALS BY 1
No. of intervals for histograms will be decreased by 1.

5: CHANGE MAX & MIN
The message

MAX =

will be output on the console, reply <number> terminated by a space.

To the message

MIN =

reply <number> terminated by a space.

New histograms will be displayed in which only records with values < new max. and ≥ new min. for the variable being displayed have been included.

7: NEXT VARIABLE
The next variable on the workfile will be displayed as histograms. All commands given at level 2, except A), will remain effective. If the last variable in the workfile is currently being displayed the next variable will be the first.

8: OPTIONS
Options level 2.

5.2.3. INTERPRETATION OF OPTIONS LEVEL 2 FOR SCATTERGRAMS

A) <variable identifier 1>,<variable identifier 2>;

<variable identifier 1> will be the X-axis

<variable identifier 2> will be the Y-axis.

B) <list of unsigned integers>;

Default: If groups have not been defined all the records on the workfile will be displayed, each point appearing as an 0.

B) If groups have been defined all groups will be displayed; members of group 1 will appear as small 1's etc. The group definitions will be evaluated in the order the commands C) were given at level 1 for each record on the workfile. As soon as a record is found to belong to a group it will be displayed as a member of that group.

The list of unsigned integers defines the groups to be displayed.

C);

Default: The data points are displayed as small characters.

The data points are displayed as large characters.

D) <any alphanumeric title>;

Title, (max. 40 characters) displayed above scattergram.

E) <list of record identifiers>;

Default: Data points will be displayed as 0's or digits representing groups.

Only those records (max. 50) specified in the list of record identifiers will be displayed. Each record will be displayed as the integer part of its record identifier.

If B) is used with this command, B) will be ignored.

F);

Default: If group definitions have not been given at level 1 this command is irrelevant.

If group definitions have been given at level 1 groups will be displayed as defined for command B) at level 2.

Any group definitions given at level 1 will be ignored, and data points will all be displayed as small 0's.

X);

Display scattergram.

Y);

Return to level 1.

Z);

Exit.

5.2.4. DISPLAY OF SCATTERGRAM

The following are displayed above the scattergram.

1. Title, if any.

2. No. of points in scattergram

3. Variable identifier for X and Y-axis.

Keys 1: PLOT

2: GRID
First time key is pressed a grid is displayed which replaces the scattergram. Pressing key 2 subsequently will alternately delete and redisplay the grid. In each square of the grid the number of points which appeared within the square is displayed.

3: WINDOWING
To magnify a specified portion of the scattergram.
Press key 3 (1st): tracking cross displayed.
Move cross with light pen to minimum position on X-axis.

Press key 3 (2nd): record min X.
Move cross with light pen to maximum position on X-axis.

Press key 3 (3rd): record max X.
Move cross with light pen to minimum position on Y-axis.

Press key 3 (4th): record min Y.
Move cross with light pen to maximum position on Y-axis.

Press key 3 (5th): record max Y.

Press key 3 (6th): new scattergram of specified portion displayed.

7: NEXT VARIABLE ON Y-AXIS
A new scattergram displayed with the next variable on the workfile after <variable identifier 2> as the Y-axis.

When Y-axis represents last variable on workfile (that is not <variable identifier 1>)it will restart at the first variable.

8: OPTIONS
Options level 1.

6. CORR - A PROGRAM FOR MEANS, STANDARD DEVIATIONS AND CORRELATIONS

6.1. OPTIONS LEVEL 1

A) DISPLAY VARIABLE NAMES FOR SELCTION, MAX. 60 - (MIN (ALL, FIRST 60))

B) TYPE LIST OF RECORD IDENTIFIERS, MAX. 150 - (ALL)

C) WEIGHT DATA WITH LAST VARIABLE IN LIST - (NO WEIGHTING)

D) CHECK MISSING DATA CODES - (NO CHECK ON MISSING DATA)

X) START PROCEDURE AND TRANSFER TO LEVEL 2

Z) EXIT

Interpretation of options level 1

A) <list of variable identifiers>;

Default: Either all the variables, or the first 60 which have type 'N' on the c.i.f., whichever is the minimum will be used for the analysis.

The variables (max. 60) included in the list of variable identifiers will be used.

No. of variables for analysis = NVARC.

B) <list of record identifiers>;

Default: All the records on the c.i.f. will be used.

Only those records (max. 150) included in the list of record identifiers will be used.

C);

Default: Data will not be weighted.

Data will be weighted by the last variable in the list of variable identifiers. For each observation each variable will be multiplied by the specified variable. Therefore, for each observation the multiplier is constant, but it will vary from observation to observation.

D);

Default: Missing data codes will not be checked.

Missing data codes will be checked for; if such values do occur they will be omitted from the calculations, without omitting all the values for the observation.

X);

Calculations will start and when complete the list of options for level 2 will be displayed.

Z);

Exit.

6.2. OPTIONS LEVEL 2

A) DISPLAY MEANS AND STANDARD DEVIATIONS

B) DISPLAY CORRELATION MATRIX

C) FILE CORRELATION MATRIX AND MEANS AND ST. DEVS.

D) DISPLAY PAIRED STATISTICS

Y) RETURN TO LEVEL 1

Z) EXIT

Interpretation of options level 2

A);

To display numerical values of means and standard deviations. A maximum of 20 variables are displayed at once, the first 20 being displayed first.

Keys 1: LP OUTPUT

2: NEXT PAGE
The means and standard deviations for the next set (or page) of 20 (max) variables is displayed. If the current page is the last page the next page will be the first.

3: PREVIOUS PAGE
The means and standard deviations for the previous set (or page) of 20 variables is displayed. If the current page is the first page, the previous page will be the last.

8: *OPTIONS*
Options level 2

B);

To display the numerical values of the correlation matrix. The correlation matrix is displayed in sections or pages of a maximum of 8 columns and 16 lines at a time. The matrix is divided up as shown in Figure 2.

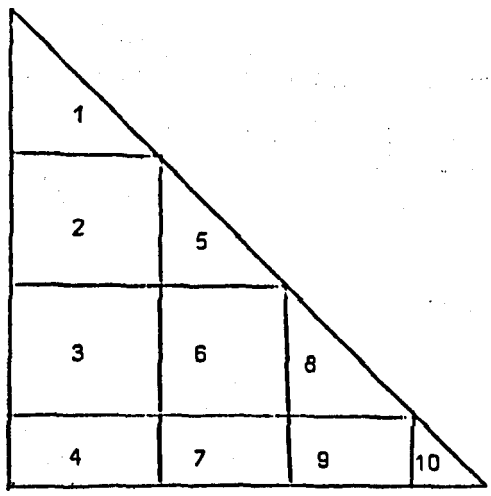


FIGURE 2

Initially page 1 is displayed, other pages may be displayed in the order indicated by pressing keys 2 and 3.

Keys 1: LP OUTPUT
The complete correlation matrix is output on the line printer. Each matrix can only be output once.

2: NEXT PAGE
The next page as indicated in Figure 2 is displayed. If the current page is the last, the next page is the first.

3: PREVIOUS PAGE
The previous page as indicated in Figure 2 is displayed. If the current page is the first, the next page is the last.

8: OPTIONS
Options level 2.

C);

To file correlation matrix, means and standard deviations at the end of the c.i.f. for input to FACTOR.

D);

To display paired statistics. This is only relevant when the missing data code option has been implemented. For each pair of variables (X,Y), (((X,Y), Y=1, X-1), X=2,NVARC), it will give the number of occasions missing data codes did not occur for that pair, and the relevant means and standard deviations. Statistics for a maximum of 20 pairs are displayed at once, the first 20 pairs in the sequence defined above, being displayed first.

Keys 1: LP OUTPUT

2: NEXT PAGE
Statistics for the next set of 20 pairs of variables in the sequence will be displayed. If the current set is the last set, the next set will be the first.

3: PREVIOUS PAGE
Statistics for the previous set of 20 pairs of variables in the sequence will be displayed. If the current set is the first set, the previous set will be the last.

Y);

To return to level 1.

Z);

Exit.

FACTOR

ENTRY

(1)

FACTOR ANALYSIS & PRINCIPAL COMPONENTS

OPTIONS AVAILABLE AT LEVEL 1

A) FACTOR ANALYSIS

B) PRINCIPAL COMPONENTS

Z) EXIT

EXIT

(2)

FACTOR ANALYSIS

OPTIONS AVAILABLE AT LEVEL 2

DEFAULT VALUES IN ()

A) INPUT FACTOR MATRIX - (CORRELATION MATRIX FROM CORR USED AS STARTING MATRIX)

B) INPUT USERS OWN CORRELATION MATRIX - (CORRELATION MATRIX OUTPUT FROM CORR IS USED)

C) SAVE FACTOR MATRIX - (MATRIX NOT SAVED)

D) DISPLAY VARIABLE NAMES FOR SELECTION - (ALL)

E) TYPE IN NO. OF FACTORS - (ALL)

F) KAISERS CRITERION USED TO DETERMINE NO. OF FACTORS - (ALL)

G) TYPE MINIMUM PERCENT OF VARIANCE TO BE EXPLAINED BY FACTOR STRUCTURE - (TEST WILL NOT BE PERFORMED)

H) COMMUNALITY ESTIMATES TAKEN FROM ORIGINAL EIGENVECTORS - (SQUARED MULTIPLE CORRELATIONS, RSQD, ARE USED AS COMMUNALITY ESTIMATES)

I) TYPE IN COMMUNALITY ESTIMATES - (RSQD)

J) MAXIMUM NO. OF ITERATIONS - (50)

K) CONVERGENCE CRITERION TO DETERMINE WHEN COMMUNALITIES HAVE CONVERGED - (.005)

L) COMMUNALITIES NOT PERMITTED TO FALL BELOW RSQD - (NO MINIMUM RESTRICTION)

X) START

Y) RETURN TO LEVEL 1

Z) EXIT

(3)

FACTOR ANALYSIS

CONVERGENCE OF COMMUNALITIES

KEYS: 1 PLOT - 2 CHANGE CONV. CRITERION - 3 CHANGE MAX. NO. ITERATIONS - 8 OPTIONS LEVEL 3

(4)

FACTOR ANALYSIS

OPTIONS AVAILABLE AT LEVEL 3

DEFAULT VALUES IN ()

A) DISPLAY EIGENVALUES

B) DISPLAY FACTOR WEIGHTS

C) DISPLAY VARIABLES IN FACTOR SPACE, TYPE IN 2 FACTOR NOS. FOR AXES

D) DISPLAY FACTORS WITH WEIGHTS AS AMPLITUDES, TYPE IN FACTOR NOS., MAX. 4

E) DISPLAY CUMULATIVE SUMS OF SQUARES OF FACTOR WEIGHTS

F) VARIMAX ROTATION

G) SCORES

H) GRAPHICAL ORTHOGONAL ROTATIONS, TYPE IN FACTOR NOS. - (1 & 2)

I) DISPLAY COMMUNALITIES

Y) RETURN TO LEVEL 2

Z) EXIT

(9)

PRINCIPAL COMPONENTS

OPTIONS AVAILABLE AT LEVEL 2

DEFAULT VALUES IN ()

A) INPUT COMPONENTS MATRIX

B) INPUT USERS OWN CORRELATION MATRIX - (CORRELATION MATRIX OUTPUT FROM CORR IS USED)

C) SAVE COMPONENTS MATRIX - (MATRIX NOT SAVED)

D) DISPLAY VARIABLE NAMES FOR SELECTION - (ALL)

X) START

Y) RETURN TO LEVEL 1

Z) EXIT

(10)

PRINCIPAL COMPONENTS

OPTIONS AVAILABLE AT LEVEL 3

DEFAULT VALUES IN ()

A) DISPLAY EIGENVALUES

B) DISPLAY COMPONENT WEIGHTS

C) DISPLAY VARIABLES IN EIGENVECTOR SPACE, TYPE IN 2 EIGENVECTOR NOS. FOR AXES

D) DISPLAY EIGENVECTORS WITH WEIGHTS AS AMPLITUDES, TYPE IN EIGENVECTOR NOS. MAX 4

E) DISPLAY CUMULATIVE SUMS OF SQUARES OF EIGENVECTOR WEIGHTS

F) VARIMAX ROTATION

G) SCORES

H) GRAPHICAL ORTHOGONAL ROTATIONS, TYPE IN EIGENVECTOR NOS. - (1 & 2)

Y) RETURN TO LEVEL 2

Z) EXIT

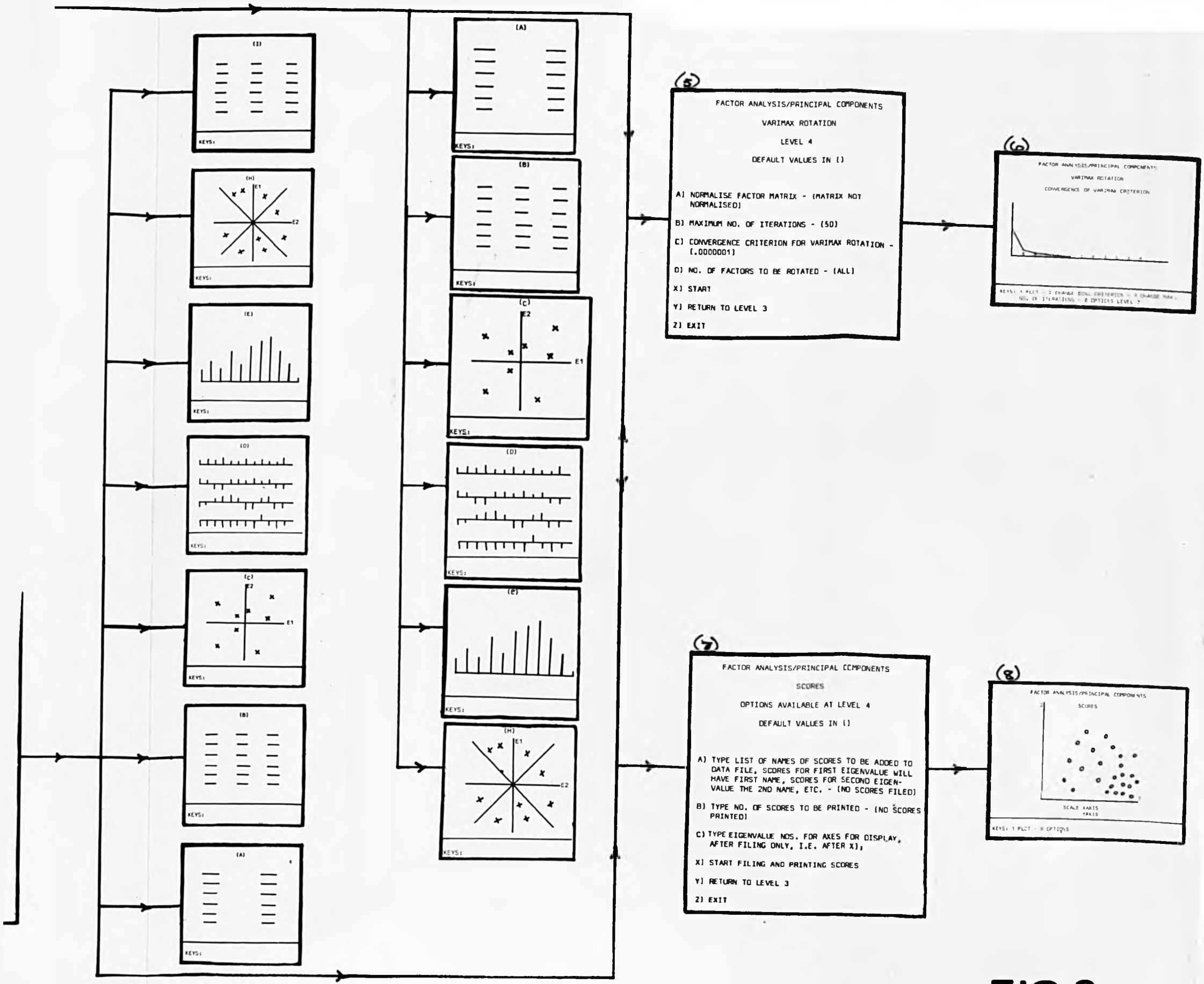

(I)
KEYS:
(H)
E1
E2
KEYS:
(E)
KEYS:
(D)
KEYS:
(C)
E2
E1
KEYS:
(B)
KEYS:
(A)
KEYS:
(A)
KEYS:
(B)
KEYS:
(C)
E2
E1
KEYS:
(D)
KEYS:
(E)
KEYS:
(H)
E1
E2
KEYS:
(5)
FACTOR ANALYSIS/PRINCIPAL COMPONENTS
VARIMAX ROTATION
LEVEL 4
DEFAULT VALUES IN ()
A) NORMALISE FACTOR MATRIX - (MATRIX NOT NORMALISED)
B) MAXIMUM NO. OF ITERATIONS - (50)
C) CONVERGENCE CRITERION FOR VARIMAX ROTATION - (.0000001)
D) NO. OF FACTORS TO BE ROTATED - (ALL)
X) START
Y) RETURN TO LEVEL 3
Z) EXIT
(6)
FACTOR ANALYSIS/PRINCIPAL COMPONENTS
VARIMAX ROTATION
CONVERGENCE OF VARIMAX CRITERION
(7)
FACTOR ANALYSIS/PRINCIPAL COMPONENTS
SCORES
OPTIONS AVAILABLE AT LEVEL 4
DEFAULT VALUES IN ()
A) TYPE LIST OF NAMES OF SCORES TO BE ADDED TO DATA FILE, SCORES FOR FIRST EIGENVALUE WILL HAVE FIRST NAME, SCORES FOR SECOND EIGENVALUE THE 2ND NAME, ETC. - (NO SCORES FILED)
B) TYPE NO. OF SCORES TO BE PRINTED - (NO SCORES PRINTED)
C) TYPE EIGENVALUE NOS. FOR AXES FOR DISPLAY, AFTER FILING ONLY, I.E. AFTER X);
X) START FILING AND PRINTING SCORES
Y) RETURN TO LEVEL 3
Z) EXIT
(8)
FACTOR ANALYSIS/PRINCIPAL COMPONENTS
SCORES
SCALE XAXIS
YAXIS

FIG.3

7. FACTOR - A PROGRAM FOR FACTOR ANALYSIS AND PRINCIPAL COMPONENTS

This program will be described with reference to the flowchart shown in Figure 3, and the boxes will be referred to as (1), (2), etc. The program operates at four levels

Level 1:- (1)
Level 2:- (2/3), (9)
Level 3:- (4), (10)
Level 4:- (5/6), (7/8)

OPTIONS LEVEL 1 (1)

A);

Factor analysis

B);

Principal components

Z);

Exit.

7.1. FACTOR ANALYSIS

7.1.1. INTERPRETATION OF OPTIONS LEVEL 2 FOR FACTOR ANALYSIS (2)

A);

Default: A correlation matrix is input for factor analysis.

To input a matrix saved during an earlier run with C);, if the matrix is on tape, mount tape on handler 6, or if the matrix is in a V-file use &ASSIGN card for channel 106.

B);

Default: Correlation matrix taken from c.i.f.

To input user's own correlation matrix from channel 37 (the card reader, unless channel 37 is reassigned).

Format of correlation matrix.

Header card.

Cols. 1 - 5	No. of variables in matrix (max. 60)	I5

Matrix in lower triangular form, diagonal present.

1st row		
Cols. 1 - 8	r_{11} (= 1.0)	F8.0
2nd row		
Cols. 1 - 8	r_{12}	2F8.0
Cols. 9 - 16	r_{22} (= 1.0)	
etc.		

If there are more than 10 variables, the 11th and subsequent rows will require more than one card. Each row must start on a new card.

C);

Default: Factor matrix not saved.

To save an unrotated factor matrix, if the matrix is to be on tape, mount tape on handler 6, or if the matrix is to be in a V-file, use &ASSIGN card for channel 106, for a V-file which must have previously been established.

D) <list of variable identifiers>;

Default: All the variables in the correlation matrix are used for factor analysis.

To display names of variables in the correlation matrix and to specify a set of variables for analysis. No. of variables for analysis = NVARF

E) <integer>;

Default: All factors are used.

To specify the number of factors NFACT

F);

Default: All factors are used.

Kaiser's criterion used to determine number of factors, i.e. NFACT = no. of eigenvectors with eigenvalues greater than 1.

G) <number>;

Default: All factors are used.

To specify that the factors must account for a given percentage of the variance; <number> is the percentage.

H);

Default: Squared multiple correlations, RSQD, are used as communality estimates.

Initial communality estimate, $(h_i^2)_0$, are taken from eigenvectors resulting from the diagonalisation of the correlation matrix. $(h_1^2)_0$ will be the sum of the squares of the elements of the first row summed over NFACT columns, etc.

I) <list of NFACT numbers>;

Default: RSQD used as communality estimates.

Initial communality estimates supplied as a list of NFACT numbers.

J) <integer>;

Default: Maximum number of times the factor matrix is diagonalised and the communalities checked for convergence is 50.

<integer> will be the new maximum.

K) <number>;

Default: Convergence criterion for each communality for two successive iterations is .005.

<number> will be the convergence criterion.

L);

Default: For communalities, upper bound = 1, lower bound = 0.

Upper bound = 1, lower bound = RSQD.

X);

To start procedure.

Y);

To return to level 1.

Z);

Exit.

7.1.2. CONVERGENCE OF COMMUNALITIES (3)

Graph. The current convergence criterion and maximum number of iterations are displayed.

X-axis: each interval represents 1 iteration.

Y-axis: a function of the sum of the differences in the communalities between iterations.

The Y co-ordinate of the rth point is

$$d_r = \sum_{i=1}^{NVARF} \left[(h_i^2)_{r-1} - (h_i^2)_r \right]^2 \Big/ NVARF \quad r = 0,1 \ldots$$

Keys. Keys will be interrogated after each iteration, therefore they may be placed in the up position at any time.

1: PLOT

2: CHANGE CONV CRITERION
To change convergence criterion.

To the message

CONV =

reply <number> terminated by a space.

New criterion is displayed on the screen.

3: CHANGE MAX NO. OF ITERATIONS
To change maximum number of iterations.

To the message

NMAX =

reply <integer> terminated by a space.

New maximum displayed on screen.

8: OPTIONS LEVEL 3
If this key is pressed before communalities have converged or maximum number of iterations is reached, options level 3 (4) is displayed.

Keys must all be switched off when the next iteration is started.

If the communalities converge within the maximum number of iterations

COMMUNALITIES CONVERGED IN <integer> ITERATIONS

is output on console.

If they do not, message

COMMUNALITIES DID NOT CONVERGE IN <integer> ITERATIONS

is output.

Then key 8 for List of Options level 3.

7.1.3. INTERPRETATION OF OPTIONS LEVEL 3 FOR FACTOR ANALYSIS
(4) with display boxes

A):

To display numerical values of final set of eigenvalues and percentages.

Keys 1: LP OUTPUT

8: OPTIONS
Options level 3

B):

To display numerical values of factor weights. The factors normalised to their eigenvalue are displayed sequentially.

Keys 1: LP OUTPUT
The complete factor matrix is output on the line printer normalised to the eigenvalues. Each matrix can only be output once.

2: NEXT VECTOR
Display next factor. If the current factor is the last, the next factor is the first.

Keys 3: PREVIOUS VECTOR
Display previous factor. If the current factor is the first, the previous factor is the last.

4: NORMALISE TO UNITY
The current and subsequent factors will be displayed normalised to unity until key 5 is pressed.

5: NORMALISE TO EIGENVALUE
The current and subsequent factors will be displayed normalised to the eigenvalues until key 4 is pressed.

8: OPTIONS
Options level 3.

C) <integer 1>,<integer 2>;

To display the variables as points in a 2-dimensional scattergram with factor <integer 1> as the X-axis and factor <integer 2> as the Y-axis. Initially the maximum and minimum values on both axes are 1.0 and -1.0.

Keys 1: PLOT

2: CHANGE SCALE ON BOTH AXES
If the max. and min. are currently 1.0 and -1.0 the new max. and min. on both axes will be determined by the maximum absolute value for all variables on both axes.

A max. and min. of 1.0 and -1.0 can be restored by pressing key 2 again.

8: OPTIONS
Options level 3.

D) <integer 1>,<integer 2>,<integer 3>,<integer 4>;

To display a maximum of four factors with the factor weights as amplitudes. A sketch shown in the flow chart D) beneath (4), each horizontal line represents a factor. Vertical lines represent amplitudes, +ve above the line, -ve below. Size displayed depends on maximum absolute value for each factor, this maximum is displayed on the +ve side of each factor.

A cursor ; a feint vertical line, which enables the user to distinguish individual variables, can be displayed and moved across the screen by pressing keys 3, 4 and 5.

Keys 1: PLOT

2: DISPLAY/DELETE CURSOR
The cursor is displayed, if it is not already there.
If it is displayed it is deleted.
Initially it points to the first variable.
The position in the correlation matrix, the number and name of the variable the cursor is currently pointing to are displayed in the top right hand corner of the screen.

3: ADVANCE CURSOR
The cursor is advanced to the next variable. After the last variable it will restart at the first.

Keys 4: MOVE CURSOR BACK
The cursor is moved one step back.
After the first variable it will go back to the last.

5: MOVE CURSOR TO START
The cursor is moved to the first variable.

8: OPTIONS
Options level 3.

E);

To display cumulative sums of squares of factor weights. Initially there is a horizontal line across the bottom of the screen.

Keys 1: PLOT

2: ADD VARIABLES
The first time this key is pressed vertical lines appear (one for each variable). Each line represents the square of the weight of that variable on the first factor. The next time key 2 is pressed the square of the weight of each variable on the second factor is added, and each vertical line then represents the sum of the squares of weights of each variable on the first two factors. When NFACT factors have been included the vertical lines represent the communalities.

If key 2 is pressed again, the procedure will be restarted.

8: OPTIONS
Options level 3.

F);

For Varimax rotation see 7.1.4-5.

G);

For factor scores see 7.1.6.

H) <integer 1>,<integer 2>;

Default: If <integer 1> and <integer 2> are not supplied, axes are initially factor 1 and factor 2.

For interactive graphical orthogonal rotations. Variables are displayed as points in a 2-dimensional scattergram with factor <integer 1> as the X-axis and factor <integer 2> as the Y-axis. Axes may be rotated using keys 4, 5. Current angle of rotation, measured in anti-clockwise direction, is displayed in top right hand corner.

When rotation is satisfactory, new scattergram may be displayed with a different pair of axes.

Keys 1: PLOT

2: NEXT PAIR AXES
New scattergrams are displayed with axes changed as follows: factor for Y-axis is increased by one until NFACT is reached, then factor for X-axis is increased by one and Y-axis is one greater than X-axis.

Keys 3: PREVIOUS PAIR AXES
Scattergram is displayed with previous pair of axes in this cycle.

4: ROTATE ANTICLOCKWISE
Rotate axes anti-clockwise, one degree at a time.

5: ROTATE CLOCKWISE
Rotate axes clockwise, one degree at a time.

6: REFLECT X
Absolute values of all factor weights for factor <integer 1> are displayed.
-ve values cannot be restored.

7: REFLECT Y
Absolute values of all factor weights for factor <integer 2> are displayed.
-ve values cannot be restored.

8: OPTIONS
Options level 3.

I);

To display numerical values of communality estimates.

Keys 1: LP output

8: Options level 3.

7.1.4. INTERPRETATION OF OPTIONS LEVEL 4 FOR VARIMAX ROTATION OF FACTORS (5)

Type F); at level 3, then (5) will be displayed.

A);

Default: Factor matrix is not normalised for rotation.

Factor matrix is to be normalised.

B) <integer>;

Default: A maximum of 50 iterations.

<integer> will be the new maximum.

C) <number>;

Default: Convergence criterion for Varimax criterion for two successive iterations is .0000001.

<number> will be the new convergence criterion.

D) <integer>;

Default: NFACT factors will be rotated.

<integer> factors will be rotated.

X);

To start rotation.

Y);

To return to level 2.

Z);

Exit.

7.1.5. CONVERGENCE OF VARIMAX CRITERION (6)

Graph. The current convergence criterion and maximum number of iterations are displayed.

X-axis: each interval represents 1 iteration.

Y-axis: the difference between the varimax criterion between iterations.

If V_o = initial Varimax criterion

and V_i = Varimax criterion after ith iteration,

then Y co-ordinate of ith point = $V_i - V_{i-1}$

Keys. Keys will be interrogated after each iteration, therefore they may be placed in the up position at any time.

1: PLOT

2: CHANGE CONV CRITERION
To the message

CONV =

reply <number> terminated by a space.

New criterion is displayed on the screen.

3: CHANGE MAX NO OF ITERATIONS
To the message

NMAX =

reply <integer> terminated by a space.

New maximum is displayed on the screen.

8: OPTIONS
If this key is pressed before Varimax criterion has converged or maximum number of iterations is reached, options level 3, (4), is displayed.

Keys must all be switched off when the next iteration is started.

If the Varimax criterion converges within the maximum number of iterations

VARIMAX CRITERION CONVERGED IN <integer> ITERATIONS

is output on console.

If it does not, the message

VARIMAX CRITERION DID NOT CONVERGE IN <integer> ITERATIONS

is output.

Then key 8 for list of options level 3, (4). See 7.1.3, all of which will now refer to the rotated factor matrix.

7.1.6. INTERPRETATION OF OPTIONS LEVEL 4 FOR FACTOR SCORES (7)

Type G); at level 3, then (7) will be displayed.

Scores are written to new c.i.f. (optional), line printer (optional) and a disc file for quick access for display. Therefore command C) for display must only be used after X); when all filing and printing is finished.

A) <list of user variable names>;

Default: No scores will be filed on new c.i.f.

New c.i.f. to be created with scores as additional variables. List contains new user variable names for scores that are to be filed. If there are NSCORE names in the list, the first NSCORE scores will be filed.

If any observations on the c.i.f. were not included in the correlation matrix computed by program CORR, variables which represent the scores will be given a value of 999. for these observations.

If c.i.f. on handler 0, new c.i.f. on handler 3.
If c.i.f. on handler 3, new c.i.f. on handler 4 and vice-versa.

New c.i.f. will not be used until command Z); is given and the next &RUN; card is read for any program.

B) <integer>;

Default: No scores are printed.

<integer> scores are printed on line printer for each observation used for the correlation matrix.

C) <integer 1>,<integer 2>;

Command may only be used after X); i.e. after filing and printing.

Factors scores displayed as points in a 2-dimensional scattergram with factor <integer 1> as X-axis and factor <integer 2> as Y-axis (see (8)). A maximum of the first 200 observations are displayed. If there are more than 200 scores the message

ONLY FIRST 200 SUBJECTS ARE DISPLAYED

is displayed. To see the scores for all subjects, scores must be filed using A) and displayed using the program DISVAR.

Keys 1: PLOT

8: OPTIONS
Options level 4.

X);

To start filing and printing.

Y);

Return to level 3.

Z);

Exit.

7.2. PRINCIPAL COMPONENTS

7.2.1. INTERPRETATION OF OPTIONS LEVEL 2 FOR PRINCIPAL COMPONENTS (9)

A);

Default: A correlation matrix is input for principal components analysis.

To input a matrix saved during an earlier run with C);,
if the matrix is on tape, mount tape on handler 6
or if the matrix is in a V-file use &ASSIGN card for channel 106.

B);

Default: Correlation matrix taken from c.i.f.

To input user's own correlation matrix from channel 37. (The card reader, unless channel 37 is reassigned). For format of correlation matrix see 7.1.1 B).

C);

Default: Components matrix not saved.

To save an unrotated components matrix,
if the matrix is to be on tape, mount tape on handler 6
or if the matrix is to be in a V-file, use &ASSIGN card for channel 106, for a V-file which must have previously been established.

D) <list of variable identifiers>;

Default: All the variables in the correlation matrix are used for principal components.

To display names of variables in the correlation matrix and to specify a subset of NVARF(= NCOMP) variables for analysis.

X);

To start procedure.

Y);

To return to level 1.

Z);

Exit.

7.2.2. INTERPRETATION OF OPTIONS LEVEL 3 FOR PRINCIPAL COMPONENTS (10) with display boxes

(10) will be displayed when diagonalisation of correlation matrix is complete.

A);

To display numerical values of eigenvalues and then percentages of total variance.

Keys 1: LP OUTPUT

8: OPTIONS
Options level 3.

B);

To display numerical values of eigenvectors or component weights. The components normalised to their eigenvalue are displayed sequentially.

Keys 1: LP OUTPUT
The complete components matrix is output on the line printer normalised to the eigenvalues. Each matrix can only be output once.

2: NEXT VECTOR
Display next vector. If the current vector is the last, the next vector is the first.

3: PREVIOUS VECTOR
Display previous vector. If the current vector is the first, the previous vector is the last.

4: NORMALISE TO UNITY
The current and subsequent vectors will be displayed normalised to unity until key 5 is pressed.

5: NORMALISE TO EIGENVALUE
The current and subsequent vectors will be displayed normalised to the eigenvalues until key 4 is pressed.

8: OPTIONS
Options level 3.

C) <integer 1>,<integer 2>;

To display the variables as points in a 2-dimensional scattergram with component <integer 1> as the X-axis and component <integer 2> as the Y-axis. Initially the maximum and minimum values on both axes are 1.0 and -1.0.

Keys 1: PLOT

2: CHANGE SCALE ON BOTH AXES
If the max. and min. are currently 1.0 and -1.0 the new max. and min. on both axes will be determined by the maximum absolute value for all variables on both axes.

A max. and min. of 1.0 and -1.0 can be restored by pressing key 2 again.

8: OPTIONS
Options level 3.

D) <integer 1>,<integer 2>,<integer 3>,<integer 4>;

To display a maximum of four components with their weights as amplitudes. A sketch is shown in the flow chart D) beneath (10), each horizontal line represents a component. Vertical lines represent amplitudes, +ve above the line, -ve below. Size displayed depends on maximum absolute value for each component, this maximum is displayed on the +ve side of each component.

A cursor, a feint vertical line, which enables the user to distinguish individual variables can be displayed and moved across the screen by pressing keys 3, 4 and 5.

Keys 1: PLOT

2: DISPLAY/DELETE CURSOR
The cursor is displayed, if it is not already displayed, on the screen.

If it is displayed it is deleted.
Initially it points to the first variable.
The position in the correlation matrix, the number and name of the variable the cursor is currently pointing to are displayed in the top right hand corner of the screen.

3: ADVANCE CURSOR
The cursor is advanced to the next variable. After the last variable it will restart at the first.

4: MOVE CURSOR BACK
The cursor is moved one step back. After the first variable it will go back to the last.

5: MOVE CURSOR TO START
The cursor is moved to the first variable.

8: OPTIONS
Options level 3.

E);

To display cumulative sums of squares of component weights.

Initially there is a horizontal line across the bottom of the screen.

Keys 1: PLOT

2: ADD VARIABLES
The first time this key is pressed, vertical lines appear (one for each variable). Each line represents the square of the weight of that variable on the first component. The next time key 2 is pressed the square of the weight of each variable on the second component is added, and each vertical line then represents the sum of the squares of the weights of each variable on the first two components.

If, after all components have been added, key 2 is pressed again, the procedure will be restarted.

8: OPTIONS
Options level 3.

F);

For Varimax rotation see 7.2.3-4.

G);

For component scores see 7.2.5.

H) <integer 1>,<integer 2>;

Default: If <integer 1> and <integer 2> are not supplied, axes are initially component 1 and component 2.

For interactive graphical orthogonal rotations.
Variables are displayed as points in a 2-dimensional scattergram with component <integer 1> as the X-axis and component <integer 2> as the Y-axis.

Axes may be rotated using keys 4, 5.

Current angle of rotation measured in anti-clockwise direction is displayed in top right hand corner.

When rotation is satisfactory new scattergram may be displayed with a different pair of axes.

Keys 1: PLOT

2: NEXT PAIR AXES
New scattergrams are displayed with axes changed as follows: component for Y-axis is increased by one until NCOMP is reached, then component for X-axis is increased by one and Y-axis is one greater than X-axis.

3: PREV PAIR AXES
Scattergram is displayed with previous pair of axes in this cycle.

4: ROTATE ANTICLOCKWISE
Rotate axes anti-clockwise, one degree at a time.

5: ROTATE CLOCKWISE
Rotate axes clockwise, one degree at a time.

Keys 6: REFLECT X
Absolute values of all component weights for component <integer 1> are displayed.

-ve values cannot be restored.

7: REFLECT Y
Absolute values of all component weights for component <integer 2> are displayed.

-ve values cannot be restored.

8: OPTIONS
Options level 3.

7.2.3. INTERPRETATION OF OPTIONS LEVEL 4 FOR VARIMAX ROTATION OF PRINCIPAL COMPONENTS (5)

Type F); at level 3, then (5) will be displayed.

A);

Default: Component matrix is not normalised for rotation.

Component matrix is to be normalised.

B) <integer>;

Default: A maximum of 50 iterations.

<integer> will be the new maximum.

C) <number>;

Default: Convergence criterion for Varimax criterion for two successive iterations is .0000001.

<number> will be the new convergence criterion.

D) <integer>;

Default: All the components will be rotated.

<integer> components will be rotated.

X);

To start rotation.

Y);

To return to level 2.

Z);

Exit.

7.2.4. CONVERGENCE OF VARIMAX CRITERION (6)

Graph. The current criterion and maximum number of iterations are displayed.

X-axis: each interval represents 1 iteration.

Y-axis: the difference between the Varimax criterion between iterations.

If V_0 = initial Varimax criterion

and V_i = Varimax criterion after ith iteration

then Y co-ordinate of the ith point = $V_i - V_{i-1}$

Keys. Keys will be interrogated after each iteration, therefore they may be placed in the up position at any time.

1: PLOT

2: CHANGE CONV CRITERION
To the message

CONV =

reply <number> terminated by a space.

New criterion is displayed on the screen.

3: CHANGE MAX NO OF ITERATIONS
To the message

NMAX =

reply <integer> terminated by a space.

New maximum is displayed on the screen.

8: OPTIONS
If this key is pressed before Varimax criterion has converged or maximum number of iterations is reached, options level 3, (4) is displayed.

Keys must all be switched off when the next iteration is started.

If the Varimax criterion converges within the maximum number of iterations

VARIMAX CRITERION CONVERGED IN <integer> ITERATIONS

is output to console.

If it does not the message

VARIMAX CRITERION DID NOT CONVERGE IN <integer> ITERATIONS

is output.

Then key 8 for list of options level 3, (4). See 7.2.2 all of which will now refer to the rotated components matrix.

7.2.5. INTERPRETATION OF OPTIONS LEVEL 4 FOR COMPONENT SCORES (7)

Type G); at level 3, then (7) will be displayed.

Scores are written to new c.i.f. (optional), line printer (optional), and a disc file for quick access for display. Therefore command C) must be used after X) when all filing and printing is finished.

A) <list of user variable names>;

Default: No scores will be filed on new c.i.f.

New c.i.f. to be created with scores as additional variables. List contains new user variable names for scores that are to be filed. If there are NSCORE names in the list, the first NSCORE scores will be filed.

If any observations on the c.i.f. were not included in the correlation matrix computed by program CORR, variables which represent the scores will be given a value of 999. for these observations.

If c.i.f. on handler 0, new c.i.f. on handler 3.
If c.i.f. on handler 3, new c.i.f. on handler 4 and vice-versa.

New c.i.f. will not be used until command Z); is given and the next &RUN; is read for any program.

B) <integer>;

Default: No scores are printed.

<integer> scores are printed on line printer for each observation used for the correlation matrix.

C) <integer 1>,<integer 2>;

Command may only be used after X); i.e. after filing and printing.

Component scores displayed as points in a 2-dimensional scattergram with component <integer 1> as X-axis and component <integer 2> as Y-axis (see (8)). A maximum of the first 200 observations are displayed. If there are more than 200 scores the message

ONLY FIRST 200 SUBJECTS ARE DISPLAYED

is displayed.

To see the scores for all subjects, scores must be filed using A) and displayed using the program DISVAR.

Keys 1: PLOT

8 OPTIONS
Options level 4.

X);

To start filing and printing.

Y);

Return to level 3.

Z);

Exit.

8. EUCLID - A PROGRAM FOR NON-HIERARCHICAL CLUSTERING OR EUCLIDEAN CLUSTER ANALYSIS

8.1. METHOD

This program takes raw data and attempts to find the best set of k clusters such that the sums of squares of deviations of each observation from its cluster centre is a minimum, where k is defined by the user. Data used in the analysis can be displayed as a 2-dimensional scattergram with any two variables on the c.i.f. as axes.

The method requires an initial set of cluster centres to be defined; these can be chosen randomly or defined interactively with a scattergram and the light pen.

Initially k should be larger (or smaller) than is eventually required, a solution can then be found for this initial value of k. k can then be decreased (or increased) by one, interactively, and the cluster centres for the previous iteration are used as the starting position for the current iteration.

When k is being decreased two clusters are merged. These are chosen such that the increase in the overall sum of squares of deviations is minimised, and the centre of the new cluster is calculated.

When k is being increased the cluster with the largest squared deviation is chosen for division to define two new cluster centres. These are defined as follows: for each variable or dimension one co-ordinate for one new cluster is the mean of the old cluster less one standard deviation and for the other new cluster the mean plus one standard deviation.

8.2. DISPLAY OF SCATTERGRAMS

For this program all data is standardised for scattergrams.

Initially all points which lie within 3 standard deviations will be displayed, any points outside this range will not appear. Data points are enclosed in a box above which appear the number of standard deviations used for each axis and the number of points displayed. This can be changed by pressing key 7 at points indicated in the following sections. Once key 7 is pressed the message

NO OF STDEVS FOR X =

is typed on the console typewriter.

Reply

<number> terminated with a space

for the number of standard deviations for the X-axis.

NO OF STDEVS FOR Y =

is then typed on the console typewriter.

Reply

<number> terminated with a space

for the number of standard deviations for the Y-axis.

8.3. OPTIONS LEVEL 1

A) DISPLAY VARIABLE NAMES FOR SELECTION, MAX 60 - (MIN(ALL, FIRST 60)).

B) TYPE LIST OF RECORD IDENTIFIERS, MAX 150 - (MIN(ALL, FIRST 150)).

C) STANDARDISE DATA - (DATA NOT STANDARDISED).

D) INITIAL NO. OF CLUSTERS, MAX 20 - (20).

E) FINAL NO. OF CLUSTERS, MAX 20 - (1).

F) MINIMUM NO. OF OBS. ALLOWED FOR EACH CLUSTER - (1).

G) LIST OF VARIABLES WHICH MAY BE USED TO DISPLAY CLUSTERS, MAX 20 - (CURRENT AXES ONLY).

H) RANDOM SELECTION OF PTS FOR INITIAL CLUSTERS, TYPE NO. OF INITIAL CALLS TO RANDOM - (FIRST DATA PTS)

I) USER SELECTION OF PTS FOR INITIAL CLUSTERS - (FIRST DATA PTS)

J) CURRENT AXES TO DISPLAY CLUSTERS - (NO DISPLAY)

X) START

Z) EXIT

Interpretation of options level 1.

A) <list of variable identifiers>;

Default: Either all the variables, or the first 60 which have type 'N', on the c.i.f., whichever is the minimum, will be used for the analysis.

The variables (max. 60) included in list of variable identifiers will be used.

No. of variables for analysis = NVARC

B) <list of record identifiers>;

Default: Either all the records or the first 150 on the c.i.f., whichever is the minimum will be used.

The records (max. 150) included in the list of record identifiers will be used.

No. of records (observations) for analysis = NOBSC

C);

Default: Data will not be standardised.

Data will be standardised so that each variable used in the analysis has mean 0 and standard deviation 1.

D) <integer>;

Default: Initial no. of clusters 20.

<integer> will be initial no. of clusters (max. 20).

LFCLUS = initial no. of clusters.

E) <integer>;

Default: Final no. of clusters 1.

<integer> will be the final no. of clusters (max. 20).

NFCLUS = final no. of clusters.

F) <integer>;

Default: Minimum no. of observations allowed for each cluster = 1.

<integer> will be minimum no. of observations allowed for each cluster.

G) <list of variable identifiers>;

(If scattergrams are to be displayed, the axes may be changed at any point in the analysis).

Default: Only variables defined with J) and variables defined with A) i.e. those used in analysis may be used as axes for scattergrams.

A set of additional variables that may be required for axes can be defined with the list of variable identifiers.

H); or H) <integer>; (see note)

Default: The data for the first LFCLUS observations will be used to define initial cluster centres.

Random selection of data points which will define initial cluster centres.

I);

Default: The data for the first LFCLUS observations will be used to define initial cluster centres.

User selection of points to define initial cluster centres. Selection will be made after command X);.

J) <variable identifier 1>,<variable identifier 2>;

Default: If command G) is not given there will be no display, if command G) is given first 2 variables in the list will be used as axes for scattergram.

<variable identifier 1> defines X-axis.
<variable identifier 2> defines Y-axis for scattergram,

these are called the current axes and may be changed during the analysis.

This command or G) must be given if command I); is given.

X);

To start procedure.

Z);

Exit.

NOTE: Second form H) <integer>; is for different random starts; integer calls made to random number generator before random numbers are used for cluster centres. For a single &RUN; card, each random start will be different.

8.4. USER DEFINITION OF INITIAL CLUSTER CENTRES

When command X); is given, scattergram displayed if requested. If command I); has been given, cluster centres must be defined. This is done by defining circles on the scattergram with the light pen. The number of circles currently defined is displayed above the scattergram. Initially all points displayed as small o's.

Keys 1: PLOT

2: DISPLAY/DELETE CROSS
Display tracking cross if not displayed.
Delete tracking cross if displayed.

3: CENTRES & RADII
To define a cluster centre.
Move cross to point defining circle centre,
press key 3, 'x' will be displayed.
Move cross to point defining circle radius,
press key 3 again, 'x' will be deleted and circle displayed.

All points lying within circle will be displayed as cluster identifier, i.e. items lying within first circle will appear as 1's etc.

4: DELETE NEAREST CIRCLE
To delete a circle, move cross close to circle centre and press key 4.
Circle centre closest to current position of cross will be deleted.
Points will be redisplayed as small O's, and number of clusters currently defined readjusted.

7: CHANGE SCALE
To change scale on both axes. See 8.2 for definition.

8: START

The number of clusters defined when key 8 is pressed will override anything supplied for initial number of clusters (LFCLUS) at level 1. The centres of gravity of the sets of points lying within each circle are calculated and the cluster analysis starts.

8.5. SCATTERGRAM FOR ANALYSIS

Keys are used to initiate next iteration, i.e. change in the value of k.

Current no. of cluster and total squared deviation are displayed above scattergram.

Keys 1: PLOT

2: NEXT ITERATION
k = k - 1, if LFCLUS > NFCLUS
k = k + 1, if LFCLUS < NFCLUS

3: PRINT C MEMBERS
Output on line printer:
No. of clusters, total squared deviation and for each cluster:
Squared deviation
Co-ordinates of cluster centre
Cluster members

Keys 4: PREVIOUS ITERATION
As results are computed they are filed.
With this key the previous iteration will be read and displayed.

5: F STATISTIC
Pseudo F-Statistic displayed.

If current no. of clusters = K1, statistic displayed is

$$\frac{R(K1) - R(K2)}{R(K2)} \Big/ \left[\frac{(NOBSC - K1)}{(NOBSC - K2)} \left(\frac{K2}{K1} \right)^{2/NVARC} - 1 \right]$$

if LFCLUS > NFCLUS K2 = K1, ... LFCLUS and

if LFCLUS < NFCLUS K2 = K1 - 2,K1 - 1,K1

with NVARC(K2-K1) and NVARC(NOBSC-K2) degrees of freedom.

7: CHANGE SCALE
To change scale on both axes. See 8.2 for definition.

8: OPTIONS
Options level 2.

8.6. OPTIONS LEVEL 2 TO DISPLAY INDIVIDUAL CLUSTERS

A) DISPLAY INDIVIDUAL CLUSTERS WITH IDENTIFIERS

B) TYPE 2 VARIABLE IDENTIFIERS TO DEFINE NEW AXES

C) RESTART PROCEDURE WITH MOST RECENT STARTING POSITION

D) GIVE VARIABLE NAME FOR CLUSTER IDENTIFIER TO BE FILED, AND COMMENCE FILING

Y) RETURN TO LEVEL 1 TO RESTART WITH NEW STARTING POSITION

Z) EXIT

Interpretation of options level 2.

A):

To display individual clusters, sequentially, as scattergrams with the current axes. Observations are represented by their record identifiers. Cluster no. and squared deviation are displayed above scattergram.

Keys 1: PLOT

2: NEXT ITERATION
Will return to scattergram defined in 8.4 and proceed with next iteration.

3: NEXT CLUSTER
Next cluster displayed, if the last cluster is currently displayed, next cluster is first cluster.

4: PREVIOUS CLUSTER
Previous cluster displayed, if the first cluster is currently displayed, next cluster is the last cluster.

Keys 7: CHANGE SCALE
To change scale on both axes. See 8.2 for definition.

8: OPTIONS
Options level 2.

B) <variable identifier 1>,<variable identifier 2>;

New scattergram displayed with axes

<variable identifier 1> for X-axis
<variable identifier 2> for Y-axis

These two identifiers must be either amongst those used for analysis or included in list following command G) or J) at level 1.

Keys 1: PLOT

2: NEXT ITERATION
Will return to scattergram defined in 8.4 and proceed with next iteration.

7: CHANGE SCALE
To change scale on both axes. See 8.2 for definition.

8: OPTIONS
Options level 2.

C);

Return to scattergram defined in 8.4 and restart procedure with most recent starting position.

D) <user variable name>;

A new c.i.f. is created with one additional variable, the cluster identifier with name given by <user variable name>. This variable will be of type 'N' and will take integer values 1,...k, if k is the current number of clusters. If there are observations on the c.i.f. not included in the analysis, for these observations the value of this variable will be zero.

If c.i.f. on handler 0, new c.i.f. on handler 3.
If c.i.f. on handler 3, new c.i.f. on handler 4 and vice-versa.

New c.i.f. will not be used until command Z); is given and the next &RUN; card is read for any program.

Y);

Return to level 1 and restart procedure from beginning.

Z);

Exit.

9. DISTANCE - A PROGRAM TO COMPUTE A MATRIX OF DISTANCE COEFFICIENTS

9.1. OPTIONS LEVEL 1

A) TYPE LIST OF VARIABLES TO BE INCLUDED IN EACH DIJ, MAX 60 - (MIN(ALL, FIRST 60)).

B) TYPE LIST OF OBS TO BE INCLUDED - (MIN(ALL,FIRST 150)).

C) PRINT DISTANCE MATRIX - (MATRIX NOT PRINTED).

D) STANDARDISE DATA - (DATA NOT STANDARDISED).

E) WRITE DISTANCE MATRIX TO FILE FOR LINK TO UMRCC - (MATRIX NOT FILED).

X) START AND EXIT.

Interpretation of options level 1.

A) <list of variable identifiers>;

Default: Either all the variables, or the first 60 which have type 'N' on the c.i.f., whichever is the minimum, will be used for the analysis.

The variables (max. 60) included in the list of variable identifiers will be used in the analysis.

No. of variables for analysis = NVARC.

B) <list of record identifiers>;

Default: Either all the records on the c.i.f., or the first 150 on the c.i.f., whichever is the minimum, will be used for the analysis.

Only those records (max. 150) specified in the list of record identifiers will be used.

No. of observations for analysis = NOBSC.

C);

Default: distance matrix will not be printed.

Distance matrix will be output to line printer.

D);

Default: Data will not be standardised.

Data will be standardised so that each variable used for distance coefficients has mean 0, and standard deviation 1.

E);

Default: Distance matrix will be filed at end of c.i.f., but it will not be filed elsewhere.

Distance matrix will be punched on paper tape, channel 33, which may be reassigned.

Format of file: Matrix in lower triangular form, diagonal absent, (NOBSC-1) records).

d_{ij} = distance between obsi and obsj.

1st record d_{12} 5E14.6

2nd record d_{13} d_{23}

etc.

X);

Matrix of distance coefficients is calculated and filed at the end of c.i.f., and (optionally) printed and filed elsewhere. Exit from the program is automatic.

10. CLUSTER - A PROGRAM FOR HIERARCHICAL CLUSTER ANALYSIS

10.1. OPTIONS LEVEL 1

A) NEAREST NEIGHBOUR

B) FURTHEST NEIGHBOUR

C) UNWEIGHTED MEAN PAIR, CENTROID

D) WEIGHTED MEAN PAIR

E) DISPLAY DENDROGRAM

F) DISPLAY SCATTERGRAM

Z) EXIT

10.2. OPTIONS LEVEL 2 FOR COMMANDS A); B); C; AND D);

If command A); or B); or C); or D); is given at level 1, list of options described below will be displayed.

A) INPUT USERS OWN MATRIX - (MATRIX AT END OF C.I.F.)

B) MATRIX OF SIMILARITIES - (MATRIX OF DISSIMILARITIES)

C) LIST OF RECORD IDENTIFIERS OF OBSERVATIONS TO BE CLUSTERED - (ALL OBS. IN MATRIX)

D) PRINT HISTORY OF CLUSTERING - (NOT PRINTED)

E) PRINT ORDER OF OBSERVATIONS FOR DENDROGRAM - (NOT PRINTED)

X) START AND RETURN TO LEVEL 1

Z) EXIT

Interpretation of options level 2.

A);

Default: Distance matrix taken from c.i.f.

To input user's own matrix from channel 37 (the card reader unless channel 37 is reassigned).

Format of matrix.

Header card.

Cols. 1 - 5	No. of observations in matrix (max 150)	I5

Matrix in lower triangular form, diagonal present.

1st row		
Cols. 1 - 10	S_{11}	E10.5
2nd row		
Cols. 1 - 10	S_{12}	2E10.5
Cols. 11 - 20	S_{22}	
etc.		

If there are more than 8 observations the 9th and subsequent rows will require more than one card. Each row must start on a new card.

B);

Default: The matrix is one of dissimilarities.

A matrix of similarities is to be input.

C) <list of record identifiers>;

Default: All the observations in the matrix will be used in the cluster analysis.

Only those specified in the list of record identifiers will be used.

D);

Default: History of clustering not printed.

History of clustering output to line printer under the following headings.

CYCLE NO.	IGROUP	NO. OF OBS IN IGROUP	JGROUP	NO. OF OBS IN JGROUP	DISTANCE

For nearest neighbour and furthest neighbour, IGROUP and JGROUP are the clusters being merged at the current cycle, the number of items in IGROUP and JGROUP gives the number of observations already in each group. The distance is the distance between the two clusters.

For weighted and unweighted mean pair, when two clusters are merged the new cluster assumes the name of the IGROUP cluster, therefore the identifier for JGROUP will never reappear. For weighted mean pair the number of observations for JGROUP and IGROUP will always be 1. For unweighted mean pair these columns will contain the number of observations in IGROUP and JGROUP.

E);

Default: Ordered list of observations for dendrogram will not be printed.

The list of observations in the order they appear in the dendrogram. (There is no room for these when the dendrogram is displayed).

X);

The procedure will be started and when complete the list of options for level 1 will reappear.

10.3. INTERPRETATION OF OPTIONS LEVEL 1 CONT.

E);

Display dendrogram.

Keys 1: PLOT

Keys 3: CLEVEL
Press key 3 (1st): tracking cross displayed.

Press key 3 (2nd): vertical position of cross recorded, dotted horizontal line drawn across dendrogram and CLEVEL recorded for scattergram. If line crosses k vertical lines then k clusters will be defined in scattergram. Clusters will be numbered starting from the left.

8: OPTIONS
Options level 1.

F);

Display scattergram and list of options level 2 will appear.

A) VARIABLES FOR AXES TO DISPLAY DATA - (FIRST 2 ON C.I.F.)

X) START

Y) RETURN TO LEVEL 1

Z) EXIT

A) <variable identifier 1>,<variable identifier 2>;

Default: First two variables on c.i.f. will be used as axes for scattergram.

<variable identifier 1> will be used for X-axis
<variable identifier 2> will be used for Y-axis

X);

Display scattergram.

If dendrogram has not been displayed, and key 3 was not pressed and a CLEVEL recorded, all observations will appear as small O's.

If a CLEVEL has been recorded each observation will be displayed as a member of one of k groups.

All data will be standardised for the display.

Initially all data points which lie within 3 standard deviations will be displayed, any points which lie outside this range will not appear. The number of standard deviations for the display can be altered by using key 7.

Keys 1: PLOT

7: CHANGE SCALE
To message on console typewriter

NO OF STDEVS FOR X =

reply

<number> terminated with a space for number of standard deviations for X-axis.

To message

NO OF STDEVS FOR Y =

reply

<number> terminated by a space for number of standard deviations for Y-axis.

8: OPTIONS
Options for scattergram, level 2.

Y);

Return to level 1.

Z);

Exit.

11. DISCRIM - A PROGRAM FOR DISCRIMINANT ANALYSIS

11.1. OPTIONS LEVEL 1

A) DISPLAY VARIABLE NAMES FOR SELECTION, MAX 60 - (MIN(ALL, FIRST 60))

B) TYPE LIST OF RECORD IDENTIFIERS TO DEFINE ONE GROUP, FIRST COMMAND B) DEFINES FIRST GROUP ETC.

C) TYPE IF STATEMENTS TO DEFINE ONE GROUP, FIRST COMMAND C) DEFINES FIRST GROUP ETC.

D) STANDARDISE DATA - (NOT STANDARDISED)

E) GROUPS WEIGHTED BY PROPORTIONATE FREQUENCY FOR DISCRIMINANT SCORES - (EQUAL WEIGHTS)

X) TRANSFER TO LEVEL 2

Z) EXIT

Interpretation of options level 1.

A) <list of variable identifiers>;

Default: Either all the variables, or the first 60 which have type 'N' on the c.i.f., whichever is the minimum, will be used for the analysis.

The variables (max. 60) included in the list of variable identifiers will be used in the analysis.

No. of variables for analysis = NVARC.

Commands starting B) or C) may be used to define a maximum of 10 sample populations or groups. There must be one command for each group. For the evaluation of any one set of discriminant functions commands starting B) and C) may not be mixed.

B) <list of record identifiers>;

Each command of this type defines a group, the first such command defining the first group, etc. The group will consist of the records whose identifiers appear in the list of record identifiers.

C) IF <logical expression>;

Each command of this type defines a group, the first such command defining the first group, etc. The group will consist of those records for which the logical expression is true.

D);

Default: The data will not be standardised.

The data will be standardised so that, for all the observations included in the analysis each variable has zero mean and unit variance.

E);

Default: The probabilities of occurrence of each group will be assumed to be equal.

The probabilities of occurrence of each group will be weighted according to the relative size of the group.

X);

Transfer to level 2.

Z);

Exit.

11.2. OPTIONS LEVEL 2

A) DISPLAY WEIGHTS & CONSTANT TERM FOR DISCRIMINANT SCORES

B) DISPLAY CONTINGENCY TABLE

C) SCATTERGRAM WITH CANONICAL VARIATES AS AXES, TYPE 2 INTEGERS FOR AXES - (1 & 2)

D) FILE CANONICAL VARIATE SCORES, TYPE LIST OF USER VARIABLE NAMES

E) TYPE VALUES FOR NEW OBSERVATION FOR ALLOCATION TO GROUP, RESULT GIVEN ON CONSOLE

Y) RETURN TO LEVEL 1

Z) EXIT

Interpretation of options level 2.

A):

To display numerical values of weights and constant term for discriminant scores. These are displayed for one group at a time.

Keys 1: LP OUTPUT
The complete set of weights and constant terms is output to the line printer. These can only be output once.

2: NEXT GROUP
Display weights and constant term for the next group. If the current group is the last, the next group is the first.

3: PREVIOUS GROUP
Display weights and constant term for the previous group. If the current group is the first, the previous group is the last.

8: OPTIONS
Options level 2.

B);

A contingency table is displayed showing how the data used to define the discriminant functions is allocated using these same discriminant functions. The columns give the predicted groups, or the groups to which the observations are assigned by means of the discriminant functions, the rows give the actual groups.

Keys 1: LP OUTPUT

8: OPTIONS
Options level 2.

C); or C) <integer 1>,<integer 2>;

Default: If integer 1 and integer 2 are not supplied, initially the X and Y axes are the first and second canonical variates respectively.

To display group means with canonical variates as axes; variate <integer 1> as X-axis, variate <integer 2> as Y-axis. Group means are displayed as a single character; '1' for the first group mean etc.

Keys 1: PLOT

2: DISPLAY CROSS AND SPECIFY GROUPS
To display the observations with the group means.

Press key 2 to display tracking cross.
i) Move cross with light pen close to mean of group whose observations are to be displayed.
ii) Press key 2.

Repeat i) and ii) for each group whose observations are to be displayed.

3: DELETE CROSS
Cross deleted and observations displayed for groups which were specified while tracking cross was displayed.

4: DELETE OBS
All observations currently displayed will be deleted.

6: CHANGE CHARACTER SIZE
Initially group means displayed as medium size characters and observations as small. Key 6 will change means to large and observations to medium size. Pressing key 6 again changes them back to their original size.

7: NEXT CANONICAL VAR ON Y AXIS
New scattergram will be displayed with X-axis remaining the same and variate number for Y-axis increased by one.

If Y-axis is currently the last of the canonical variates and key 7 is pressed, the Y-axis will be the first canonical variate which is not equal to the current variate for X-axis.

8: OPTIONS
Options level 2.

D) <list of user variable names>;

New c.i.f. to be created with canonical variate scores as additional variables. List contains new user variable names for scores that are to be filed. If there are NSCORE names in the list, the first NSCORE scores will be filed.

If c.i.f. on handler 0, new c.i.f. on handler 3.
If c.i.f. on handler 3, new c.i.f. on handler 4 and vice-versa.

New c.i.f. will not be used until command Z); is given and the next &RUN; is read for any program.

E) <list of NVARC numbers>;

List contains NVARC numbers defining new observation to be allocated to a group.

Message output to console

ALLOCATED TO GROUP <integer>

indicating to which group the new observation has been allocated.

Y);

Return to level 1.

Z);

Exit.

REFERENCES AND BIBLIOGRAPHY

Annoni, S., Giulitti, L., Saccone, G., Casè, L. and Ninni, P. (1972) GRIPS: an interactive system for integrated circuit layout. ON LINE 72 Conference Proceedings, Vol. 2, 819-40.

Anderson, E. (1960) A semigraphical method for the analysis of complex problems. Technometrics 2, 387-91.

Andrews, D. F. (1972) Plots of high-dimensional data. Biometrics 28, 125-36.

Armit, A. P. (1971a) Curve and surface design using Multipatch and Multiobject design systems. Computer Aided Design 3, 3-12.

Armit, A. P. (1971b) The interactive languages of Multipatch and Multiobject design systems. Computer Aided Design 3, 10-5.

Ashton, E. H., Healy, M. J. R. and Lipton, S. (1957) The descriptive use of discriminant functions in physical anthropology. Proc. Roy. Soc. B 146, 552-72.

Avery, K. R. and Avery, C. A. (1975) Design and development of an interactive statistical system. (SIPS). Proceedings of Computer Science and Statistics: 8th Annual Symposium on the Interface. Health Sciences Computing Facility, University of California, Los Angeles. 49-55.

Ball, G. H. and Hall, D. J. (1965) ISODATA - a novel method of data analysis and pattern classification. Stanford Research Institute, Menlo Park, Calif.

Ball, G. H. and Hall, D. J. (1970) Some implications of interactive graphic computer systems for data analysis and statistics. Technometrics 12, 17-31.

Beale, E. M. L. (1969) Euclidean cluster analysis. Contributed paper to the 37th session of the International Statistical Institute.

Beaujon, H. J. (1970) An interactive graphical display system for illustrating elementary properties of statistical distributions. (Master's Thesis) Univ. N. Carolina. Chapel Hill N.C.

Blair, F. W., Greismer, J. H. and Jenks, R. D. (1970) An interactive facility for symbolic mathematics RC2766 (No. 12987), IBM, T. J. Watson Research Center, Yorktown Heights, N.Y.

Britt, P. M., Dixon, W. J., Jennrich, R. I. (1969) Time Sharing and interactive statistics. Statistical Computation. Milton and Nelder, Ed. Academic Press. 243-65.

Burstall, R. M., Collins, J. S. and Popplestone, R. J. (1971) Programming in Pop-2. University Press, Edinburgh.

Chamberlain, R. L. (1975) Computer graphics and time series analysis. Proceedings of Computer Science and Statistics: 8th Annual Symposium on the Interface. Health Sciences Computing Facility, University of California, Los Angeles. 20-6.

Chambers, J. M. (1967) Some general aspects of statistical computing. Applied Statistics 16, 124-32.

Chau, A. Y. C., Davies, B. W. and Zacharov, B. (1974) ISLAND - An interactive graphics system for mathematical analysis. 17, 104-12.

Chernoff, H. (1971) The use of faces to represent points in n-dimensional space graphically. Stanford Technical Report.

Cheung, T. (1974) An interactive graphic display for region partitioning by Linear programming. Comm. ACM 17, 513-6.

Ciaffi, F. and Marello R. (1972) An interactive smoothing system for curves and surfaces of car bodies. ONLINE 72 Conference Proceedings, Vol. 1, 693-708.

Colin, A. J. T. (1967) On-line access systems in statistics. Applied Statistics 16, 111-9.

Cooper, B. E. (1967) ASCOP - A statistical computing procedure. Applied Statistics 16, 100-10.

Cooper, B. E. (1970) ASCOP User Manual. The National Computing Centre Limited.

den Hartog, G. and Veenman, P. (1972) DIECAST - Display Interaction Enhancing Computer Aided Shape Technique. ONLINE 72 Conference Proceedings, Vol.2, 425-43.

Dixon, W. J. (1967) Use of displays with packaged statistical programs. Proceedings AFIPS Fall Joint Computer Conference 31. Thompson Books. 481-4.

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

Dixon, W. J. (Ed) (1975) BMD. Biomedical Computer Programs. University of California Press, Berkeley.

Engleman, C. (1965) MATHLAB: a program for on-line machine assistance in symbolic computations. Proc. AFIPS FJCC. 27. Pt. 2. AFIPS Press, Montvale, N.J. 117-26.

Feder, P. I. (1974) Graphical techniques in statistical data analysis. Technometrics 16, 287-99.

Feeser, L. J. (1972) Highway design evaluation using interactive graphics. ONLINE 72 Conference Proceedings Vol. 1, 771-88.

Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179-88.

Fletcher, R. and Reeves, C. M. (1964) Function minimization by conjugate gradients. Computer Journal 7, 149-54.

Frane, J. W. (Ed) (1975) Proceedings of Computer Science and Statistics: 8th Annual Symposium on the Interface. Health Sciences Computing Facility, University of California, Los Angeles.

Frane, J. W. (1976) The BMD and BMDP series of statistical computer programs. Comm. ACM 19, 570-6.

Fukunaga, K. and Koontz, W. L. G. (1970) A criterion and an algorithm for grouping data. IEEE Transactions on Computers C-19, 917-23.

Fukunaga, K. and Olsen, Dr. R. (1971) A two-dimensional display for the classification of multivariate data. IEEE Transactions on Computers C-20, 917-23.

GINO-F (1975) The General Purpose Graphics Manual. CAD Centre, Cambridge.

Glass, R. L. (1969) An elementary discussion of compiler/interpreter writing. Computer Surveys 1, 55-77.

Goodenough, J. B. (1965) A light-pen-controlled program for on-line data analysis. Comm. ACM 8, 130-4.

Gower, J. C. (1966) Some distance properties of latent roots and vector methods used in multivariate analysis. Biometrika 53, 325-38.

Gower, J. C. (1967) A comparison of some methods of cluster analysis. Biometrics 23, 623-37.

Gower, J. C., Simpson, H. R. and Martin, A. H. (1967) A statistical programming language. Applied Statistics 16, 89-99.

Hall, G. H., Ball, D. J., Wolf, D. E. and Eusebio, J. W. (1968) PROMENADE - An improved interactive graphics man/machine system for pattern recognition. Final report, Contact F 30602-67-C-0351.

Harman, H. H. (1960) Modern Factor Analysis. University of Chicago Press, Chicago.

Harris, D. R. (1972) GOLDA, a graphical on-line system for data analysis. (Ph.D Thesis). Ohio State University, Columbus, Ohio.

Hart, W. B. (1972) The application of computer aided design techniques to glassware and mould design. Computer Aided Design 4, 57-66.

Herraman, C. (1968) Sums of squares and products matrix. Algorithm AS 12. Applied Statistics 17, 289-92.

Higman, B. (1967) A Comparative Study of Programming Languages. MacDonald/Elsevier Computer Monographs, 2.

Hill, A. B., Wild, R. and Ridgeway, C. C. (1969) Women at Work. University of Bradford Management Centre.

Hope, K. (1968) Methods of Multivariate Analysis. University of London Press Ltd.

IBM Systems Journal (1968) 7. Interactive Graphics in Data Processing.

ICL 4100 Technical Manual. (1969) Section 1.4.10 for functional specifications and Section 2.16.3 for Fortran interface routines.

Iverson, K. E. (1962) A Programming Language. John Wiley and Sons. New York.

Jardine, N. and Sibson, R. (1968) The construction of hierarchic and non-hierarchic classifications. Computer Journal 11, 177-84.

Jardine, N. and Sibson, R. (1971) Mathematical Taxonomy. Wiley.

Joyce, S. M. (1972) The development of an interactive statistical language. ONLINE 72 Conference Proceedings Vol. 2, 477-96.

Kaiser, H. F. (1958) The varimax criterion for analytic rotation in factor analysis. Psychometrika 23, 187-200.

Kaiser, H. F. (1960) The application of electronic computers to factor analysis. Educational and Psychological Measurement 20, 141-51.

Kendall, M. G. (1966) Discrimination and classification. Multivariate Analysis I. P. R. Krishnaiah, Ed. Academic Press, New York. 165-85.

Kendall, M. G. (1968) A Course in Multivariate Analysis. Charles Griffin, London.

Kruskal, J. B. (1964a) Multidimensional scaling by optimising goodness-of-fit to a nonmetric hypothesis. Psychometrika 29, 1-27.

Kruskal, J. B. (1964b) Nonmetric multidimensional scaling: a numerical method. Psychometrika 29, 115-29.

Lance, G. N. and Williams, W. T. (1967) A general theory of classificatory sorting strategies. I. Hierarchical systems. Computer Journal 9, 373-80.

Ledermann, W. (1937) On the rank of the reduced correlation matrix in multiple-factor analysis. Psychometrika 2, 85-93.

Lee, R. C. T., Slagle, J. R. and Blum, H. (1977) A triangulation method for the sequential mapping of points from N-space to 2-space. IEEE Transactions on Computers C-26, 288-92.

MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symp. on Probability and Statistics 1, 281-97.

Margolin, B. H. (1976) Design and analysis of factorial experiments via interactive computing in APL. Technometrics 18, 135-50.

Maver, T. W. (1972) PACE: an interactive package for building design appraisal. ONLINE 72 Conference Proceedings Vol. 1, 967-86.

McDouall, D. C. (1969) Electronic design using graphics. Computer Graphics. Parslow, Prowse and Green (Ed). Plenum Press. 169-77.

Morris, R. (1968) Scatter storage techniques. CACM 11, 38-44.

MULREG (1973) A general stepwise multiple regression program. (ICL 4130) KOS User Manual.

Muxworthy, D. (1976) Introductory Guide to the Conversational SPSS Program on EMAS. Program Library Unit, Edinburgh Regional Computer Centre.

National Population Census. (1971) Office of Population Censuses and Surveys.

Nelder, J. A. et al. (1975a) GENSTAT Reference Manual. Program Library Unit, Edinburgh Regional Computer Centre.

Nelder, J. A. (1975b) The GENSTAT Language. Program Library Unit, Edinburgh Regional Computer Centre.

Nelder, J. A. (1975c) GLIM Manual. Numerical Algorithms Group, Oxford.

Newman, W. M. and Sproull, R. F. (1973) Principles of Interactive Computer Graphics. McGraw-Hill.

Nie, N. H., Hull, C. H., Steinbrenner, K. and Bent, D. H. (1975) Statistical Package for the Social Sciences. Second edition. McGraw-Hill.

Pickett, R. M. and White, B. W. (1966) Constructing data pictures. Proceedings of the National Symposium of the Society for Information Display, 75-81.

Prior, W. A. J. (1972) The GHOST Graphical Output System User Manual. UKAEA Research Group Program Documentation Note. Report CLM-PDN 8/71.

Rao, C. R. (1952) Advanced Statistical Methods in Biometric Research. John Wiley and Sons, New York.

Roberts, H. V. (1974) Conversational Statistics. Hewlett-Packard.

SPSS. (1974) Document on Interactive SPSS Project. Program Library Unit, Edinburgh Regional Computer Centre.

SSP. (1970) System/360 Scientific Subroutine Package. Version III Programmer's Manual. GH20 0205 4.

Sammon, J. W. (1968) On-line pattern analysis and recognition system. (OLPARS) RADC-TR-68-263.

Sammon, J. W. (1969) A non-linear mapping for data structure analysis. IEEE Transactions on Computers C-18, 401-9.

Schucany, W. R., Minton, P. D. and Stanley, S. H. (1972) A survey of statistical packages. Computing Surveys 4, 65-79.

Sibson, R. (1971) Some observations on a paper by Lance and Williams. Computer Journal 14, 156-7.

Sibson, R. (1973) SLINK: An optimally efficient algorithm for the single-link cluster method. Computer Journal 16, 30-4.

Smith, L. B. (1970a) A survey of interactive graphical computer systems for mathematics. Computing Surveys 2, 261-301.

Smith, L. B. (1970b) The use of interactive graphics to solve numerical problems. Comm. ACM 13, 625-34.

Sokal, R. R. and Sneath, P. H. (1963) Principles of Numerical Taxonomy, W. H. Freeman.

Sparks, D. N. (1973) Euclidean cluster analysis. Algorithm AS 58. Applied Statistics 22, 126-9.

Stamen, J. P. and Wallace, R. M. (1973) JANUS: a data management and analysis system for the behavioral sciences. Proc. ACM 1973 Annual Conf., 273-82.

Thurstone, L. L. (1947) Multiple Factor Analysis. University of Chicago Press. Chicago.

Tukey, J. W. (1962) The future of data analysis. Annals of Mathematical Statistics 33, 1-67.

Walter, P. E. (1969) Computer graphics used for architectural design and costing. Computer Graphics. Parslow, Prowse and Green, Ed. Plenum Press. 125-33.