School of Mathematics and Statistics

VARIABLE SELECTION IN PRINCIPAL COMPONENT ANALYSIS: USING MEASURES OF MULTIVARIATE ASSOCIATION

by

Moses Mefika Sithole

This thesis is presented as part of the
requirements for the award of the degree
of Master of Science in Mathematics.

CURTIN UNIVERSITY OF TECHNOLOGY

February, 1992

CERTIFICATION

I hereby certify that the work presented in this thesis is my own work and that all references are duly acknowledged. This work has not been submitted previously, in whole or in part, in respect of any other academic award at this University or elsewhere.

Moses Mefika Sithole

February, 1992.

ABSTRACT

This thesis is concerned with the problem of selection of important variables in *Principal Component Analysis* (PCA) in such a way that the selected subsets of variables retain, as much as possible, the overall multivariate structure of the complete data. Throughout the thesis, the criteria used in order to meet this requirement are collectively referred to as measures of *Multivariate Association* (MVA). Most of the currently available selection methods may lead to inappropriate subsets, while Krzanowski's (1987) M^2-*Procrustes* criterion successfully identifies *structure-bearing* variables particularly when groups are present in the data. Our major objective, however, is to utilize the idea of *multivariate association* to select subsets of the original variables which preserve *any* (*unknown*) multivariate structure that may be present in the data.

The first part of the thesis is devoted to a study of the choice of the *number of components* (say, *k*) to be used in the variable selection process. Various methods that exist in the literature for choosing *k* are described, and comparative studies on these methods are reviewed. Currently available methods based exclusively on the eigenvalues of the covariance or correlation matrices, and those based on *cross-validation* are unsatisfactory. Hence, we propose a new technique for choosing *k* based on the *bootstrap methodology*. A full comparative study of this new technique and the *cross-validatory* choice of *k* proposed by Eastment and Krzanowski (1982) is then carried out using data simulated from Monte Carlo experiments.

The remainder of the thesis focuses on variable selection in PCA using measures of MVA. Various existing selection methods are described, and comparative studies on these methods available in the literature are reviewed. New methods for selecting variables, based on measures of MVA are then proposed and compared among themselves as well as with the M^2-*procrustes* criterion. This comparison is based on Monte Carlo simulation, and the behaviour of the selection methods is assessed in terms of the performance of the selected variables.

In summary, the Monte Carlo results suggest that the proposed *bootstrap* technique for choosing *k* generally performs better than the *cross-validatory* technique of Eastment and Krzanowski (1982). Similarly, the Monte Carlo comparison of the variable selection methods shows that the proposed methods are comparable with or better than Krzanowski's (1987) M^2-*procrustes* criterion. These conclusions are mainly based on data simulated by means of Monte Carlo experiments. However, these techniques for choosing *k* and the various variable selection techniques are also evaluated on some real data sets. Some comments on alternative approaches and suggestions for possible extensions conclude the thesis.

ACKNOWLEDGEMENTS

I would like to thank all the members of the School of Mathematics and Statistics at Curtin University of Technology for their assistance throughout the duration of this project. In particular, I am most grateful to Dr. S. Ganeshanandam for supervising this project and for his invaluable help, encouragement and guidance throughout.

This thesis is a direct result of original research work which I did while I held an Australian International Development Assistance Bureau (AIDAB) award and I wish to thank the Bureau for the award.

During the course of ‘writing up’ this thesis, Dr. S. Ganeshanandam had to leave Curtin University for New Zealand to take up an appointment within the Department of Statistics at Massey University. I would like to thank the other members of my thesis committee Associate Professor L. Caccetta and Dr. N. R. Achuthan for providing me with the necessary help, such as the arrangements for ‘after hours’ access to the personal computers in the Statistics Laboratory and access to the Curtin electronic mail system to communicate with Dr. Ganeshanandam during this time.

On a more personal note, a special ‘thank-you’ to Soo Yuen who typed in part of the data and for her constant support and encouragement during the final stages of the production of this thesis.

CONTENTS Page

LIST OF TABLES Page

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Multivariate data analysis

In nearly all fields of scientific enquiry, ranging from biology to psychology, more often the researcher finds himself or herself faced with research objectives prescribing that several attributes (*variables*) be measured on each of a set of individuals or objects. In all such cases, the resulting data is *multivariate*. Hence, methods of analysing multivariate data constitute an increasingly essential area of statistics.

The origins of the currently available multivariate techniques are to be found in the work of such pioneers as Pearson, Hotelling, Fisher and Mahalanobis. Due to the heavy computational demands that these methods impose, their development remained mostly abstract for years and rendered their application to only small data sets. The last two decades, however, have seen an explosion in computer development and virtually unlimited computing power is now a reality, hence these methods have gained a wide-spread popularity. Reference of detailed descriptions of the more common multivariate methods can be made to any standard multivariate text such as Anderson (1958), Tatsuoka (1971), Bolch and Huang (1974), Mardia, Kent and Bibby (1979), Chatfield and Collins (1986), Dillon and Goldstein (1984), Seber (1984) and Krzanowski (1988).

Choice of the most appropriate method to be used in a particular study mainly depends on the nature of the problem under investigation, the envisaged objectives for the analysis, and the type of data to be analysed. In general, multivariate methods are primarily concerned with studying interrelations among variables, or with looking for possible *group* differences in terms of the variables and drawing inferences concerning the populations from which the sample groups are obtained. In *Principal Component Analysis* and *Factor Analysis* for example, there is no *prior* categorization of either individuals or variables, but both methods are variable-oriented in such a way that they detect new dimensions. On the other hand, *Cluster Analysis* for example, seeks to find groups among individuals. Some techniques, however, such as *Canonical Variate Analysis* and *Discriminant Analysis* involve looking at group structures among individuals or variables and relationships between variables simultaneously.

1.2 Principal component analysis

Perception of the sample and of its internal structure in multivariate data cannot be achieved by simply looking either at the data matrix or at a set of summary statistics derived from it. A p-dimensional scatterplot (i.e., one involving p variables) seems appropriate for such an exploratory multivariate study. However, even with such a plot, for $p > 3$, perception of the data would still not be without difficulty. In such cases, a *mathematically* appealing statistical procedure is employed to reduce the dimensionality of the data; most simply by a well chosen projection, which does not disturb the overall features of the data. Thus, such a *low*-dimensional projection is convenient for sample inspection and is expected to reveal interesting patterns with respect to structure. A commonly used projection technique for analysis directed towards exploratory or descriptive modeling is *Principal Component Analysis* (PCA), whose detailed coverage can be found in most multivariate texts and published papers which include Gower (1966), Mardia *et al*. (1979), Chatfield and Collins (1986), Campbell and Atchley (1981), Jolliffe (1986), Krzanowski (1988), and Dunteman (1989). Gower (1966) presents distance properties of principal components, while Campbell and Atchley (1981) and Krzanowski (1988) presents a detailed geometric insight into PCA as well as the algebraic derivation.

At this point, before embarking on the description of the technique, it would seem quite important to mention that PCA, unlike its counterpart, *Factor Analysis* (FA), is a mathematical technique which does not require an underlying statistical model to explain the 'error' structure. The technique is generally treated as a purely descriptive tool and no distributional assumptions are made about the original variables. However, more meaningful descriptions may be attached to the components when *multivariate normality* can be assumed on the data. Mandel (1972) and Eastment and Krzanowski (1982) have argued that retaining the first k principal components in an analysis, implicitly assumes a model for the data.

PCA, in general, attempts to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much of the sample information (variation) in the data as possible. This is achievable by orthogonally transforming the axes representing the 'original' variables into a 'new' set of axes called *principal components* (PCs). These PCs are uncorrelated and are ordered so that the first *few* retain most of the variation from *all* the original variables. From an algebraic point of view, the PCs are linear combinations of the original variables such that the above constraints are satisfied. An essential notion in *Multivariate Analysis* is that of a *linear combination* of variables; it is fundamental to both *Canonical Variate Analysis* and PCA. Let **X** be the $(n{\times}p)$ data matrix obtained by observing n objects on p

variables, 'mean-centered' and appropriately scaled. Further, let $\underline{X} = (X_1\ X_2\ \ldots\ X_p)^T$ be the (px1) vector of the variables in **X** and $\mathbf{v}_j = (v_{1j}\ v_{2j}\ \ldots\ v_{pj})^T$ be a (px1) vector of coefficients. Then the linear combinations are defined by

$$PC_j = \sum_{i=1}^{p} v_{ij}\ X_i = \mathbf{v}_j^T\ \underline{X}, \tag{1.1}$$

where, PC_j (j = 1, 2, ..., p) are the new variables (or dimensions) obtained from the original variables. Once the PC_j's are obtained, the values of these new variables for each observation (object) in the data, can be found by simply substituting the corresponding values of the $\underline{X}$'s in **X** into (1.1). These transformed values are called *Principal Component scores* (PC scores). For a fixed j, PCA seeks such a linear combination, as in equation (1.1) so that the sample variance of the resulting PC scores is the j-th maximum; subject to the PC_j's being uncorrelated (or orthogonal) to each other (j = 1, 2, ..., p). The normalization constraint $\mathbf{v}_j^T\ \mathbf{v}_j = 1$, on the components of $\mathbf{v}_j$ is usually adopted. Thus, the variance of PC_j becomes $\mathbf{v}_j^T\ \mathbf{C}\ \mathbf{v}_j$, where **C** denotes the sample *covariance* matrix of the variables in **X**. Maximization of the variance of the PC_j's subject to the above constraints leads to the *eigen_equation*

$$(\mathbf{C} - l_j\ \mathbf{I})\ \mathbf{v}_j = 0, \tag{1.2}$$

where l_j is the *eigenvalue* and $\mathbf{v}_j$ is the corresponding *eigenvector* and the vector of coefficients in equation (1.1)). Note that if the variables have first been standardized to have unit variance, **C** is replaced by the correlation matrix (usually denoted by **R**). It is also common practice to scale the coefficients in each PC such that their sum of squares is equal to the corresponding eigenvalue. Thus, if $\mathbf{v}_j^*$ is the vector of coefficients corresponding to the j-th PC, scaled as just described (i.e., $\mathbf{v}_j^{*T}\mathbf{v}_j^* = l_j$), then $\mathbf{v}_j^* = \sqrt{l_j}\,\mathbf{v}_j$. These coefficients are called *component loadings*. If the original variables (i.e., the X's) have first been standardized to have unit variance *prior* to PCA (i.e., the correlation matrix **R** has been used in equation (1.2) instead of the covariance matrix **C**), then these coefficients $\mathbf{v}_j^*$ measure the correlations between the PCs and the original (standardized) variables. Hence, they are usually referred to as *component correlations*.

Let $\mathbf{V} = (\mathbf{v}_1\ \mathbf{v}_2\ \dots\ \mathbf{v}_p)$ denote the matrix of eigenvectors, and let the diagonal matrix $\mathrm{L} = (l_1\ l_2\ \dots\ l_p)$ denote the matrix of corresponding eigenvalues. Then the *eigen_equation* becomes

$$\mathbf{C} = \mathbf{VLV}^T, \tag{1.3}$$

and the eigenvectors satisfy $\mathbf{V}^T\mathbf{V} = \mathbf{VV}^T = \mathbf{I}_p$, as the PCs are orthogonal to each other. Note that, since each successive PC accounts for a maximum amount of the variation, subject to being uncorrelated with the previous PCs, and since it can be assumed that **C** or **R** is positive semidefinite and that the eigenvalues are distinct, $l_1 > l_2 > \dots > l_p \geq 0$. It is also important to note that, $\mathbf{v}_1$ and l_1 correspond to the first PC (i.e., PC_1), $\mathbf{v}_2$ and l_2 correspond to the second PC (PC_2) and so on, and that the eigenvalues are the respective variances of the different PCs. An important result, which follows by taking the trace of both sides of equation (1.3), is that the sum of the variances of the original variables is equal to the sum of the variances of the PCs (i.e., the eigenvalues).

Performing PCA using equation (1.2) (i.e, by initially finding the eigenvalues of the sample covariance or correlation matrix and then finding the corresponding eigenvectors) is already simple and computationally fast. However, ease of computation can be further enhanced by utilizing the connection between PCA and the *singular value decomposition* (SVD) of the 'mean-centered' data matrix **X** which takes the form (see, e.g., Good (1969)):

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}, \tag{1.4}$$

where $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathrm{I}_p$, $\mathbf{V}\mathbf{V}^{\mathrm{T}} = \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathrm{I}_p$ and S is diagonal with diagonal elements $s_1, s_2, \ldots, s_p$. Here, $s_1 \geq s_2 \geq \ldots \geq s_p$ are the non-negative square-roots of the eigenvalues of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ or $\mathbf{X}\mathbf{X}^{\mathrm{T}}$, the columns of **U** are the p orthonormalized eigenvectors of $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ and the rows of $\mathbf{V}^{\mathrm{T}}$ are the orthonormalized eigenvectors of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$. Now, since

$$\begin{aligned}\mathbf{X}^{\mathrm{T}}\mathbf{X} &= (\mathbf{V}\mathbf{S}\mathbf{U}^{\mathrm{T}})(\mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}})\\ &= \mathbf{V}\mathbf{S}^2\mathbf{V}^{\mathrm{T}},\end{aligned} \tag{1.5}$$

the eigenvalues of the sample covariance matrix $(n-1)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}$, which is equivalent to the correlation matrix if the columns of **X** have first been standardized, are the diagonal elements of $(n-1)^{-1}\mathbf{S}^2$.

Suppose that the PC's PC_j ($j = 1, 2, \ldots, p$) are written as in equation (1.1) (i.e., as linear combinations of the original variables X_i ($i = 1, 2, \ldots, p$)). Then for a fixed j ($j = 1, 2, \ldots, p$), the coefficients v_{ij} are given by the elements of the j-th column of **V**; i.e., v_{ij} is the (i,j)-th element of **V**. The *principal component scores* are given by **US**. It can be verified easily that if the (i,j)-th element of **U** is u_{ij} then the (i,j)-th element x_{ij} of the data matrix X, is given by

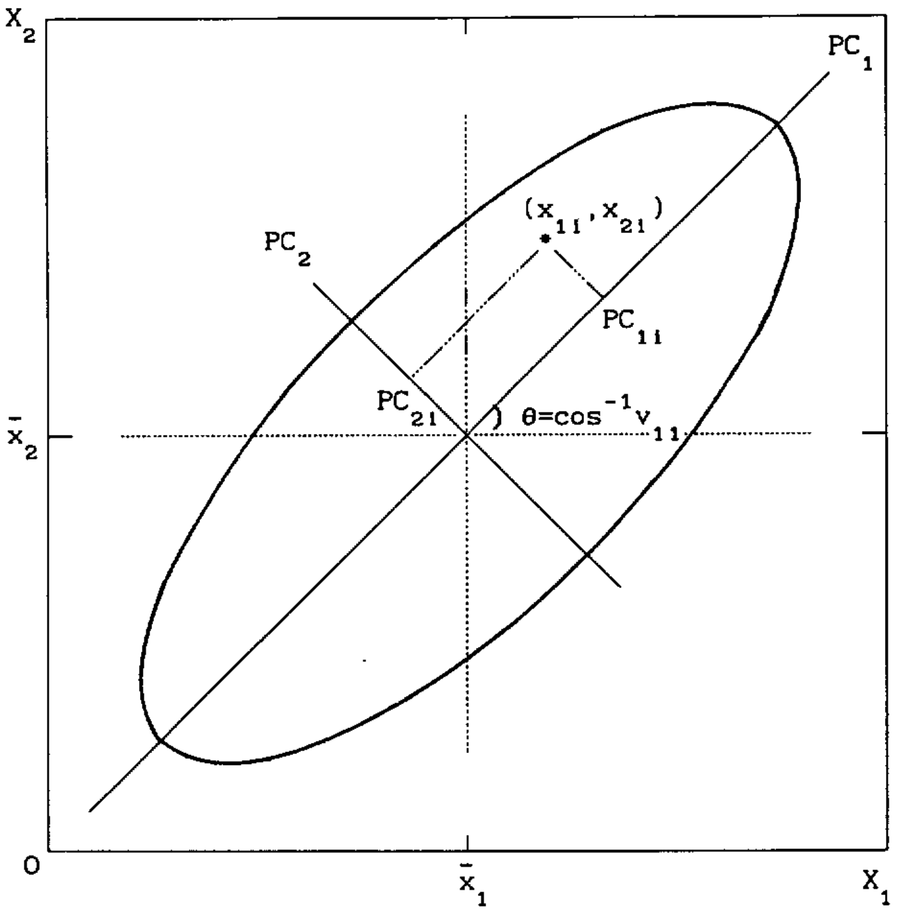
$$x_{ij} = \sum_{t=1}^{p} u_{it}\, s_t\, v_{tj}. \qquad (1.6)$$

Further descriptions of this relationship between PCA and the singular value decomposition may be found in Good (1969) or Gabriel (1978).

A detailed account of PCA with respect to geometry arises from the plot in Figure 1.1. This plot represents PCA for a case where only two variables (X_1 and X_2) are used to obtain the data. Here, it is assumed that we have a sample of *n* observations plotted on the axes OX_1 and OX_2 of the original variable coordinate system determined by the variables X_1 and X_2, respectively. It is further assumed that, the idealized elliptical cluster of points contains 95% of the total number of observations in the sample. PCA can be considered as the following two-step procedure: *translation* of the axes OX_1 and OX_2 such that the point O is moved to $(\bar{x}_1, \bar{x}_2)$, the point determined by the arithmetic means of the original variables X_1 and X_2; followed by a *rotation* of these axes to the new orthogonal axes PC_1 and PC_2, respectively, called *principal axes*, such that these new axes coincide with directions of maximum variation of the original observations.

Suppose that the data points are projected onto the axis PC_1. The point PC_{1i} ($i = 1, 2, \ldots, n$) corresponds to the projection of the point (x_{1i}, x_{2i}) onto the axis defined by the direction PC_1. The projection procedure is performed under the constraint that the variation of the points projected onto PC_1, exceeds the variation of the same points when projected onto any other line passing through the point $(\bar{x}_1, \bar{x}_2)$. The projected values corresponding to this direction of maximum variation are the *principal component scores*.

Figure 1.1 *Idealized representation of scatter plot for two variables, showing the arithmetic mean for each variable ($\bar{x}_1$ and $\bar{x}_2$), 95% concentration ellipse and principal axes* PC_1 *and* PC_2.



The first *principal axis* or *principal component* (PC_1) is often referred to as the *least squares line* since the sum of squares of the perpendicular deviations of the original points from this line is *minimal*. However, it is important to note that, this line differs from *regression lines* which minimize the sum of squares of either *horizontal* or *vertical* displacements. The cosine of the angle θ, denoted by v_{11} is the coefficient of X_1 in the linear combination representation of PC_1 (see equation (1.1)). Such coefficients are often referred to as the *direction cosines*. The first *eigenvalue* of the covariance matrix is simply the usual sample variance of the data points projected onto PC_1. Successive principal axes are determined similarly with the property that they are orthogonal to the previous axes and that they maximize the variation of the projected points subject to these constraints. For the two variables case, only one more direction can be determined and this second axis is represented by PC_2 in Figure 1.1.

It is important to notice that the PCA of a set of data depends critically upon the scales used to measure the variables. For example, when the measured variables are not comparable in magnitude of variance as well as in terms of their units of measurement, those variables with large variances will dominate the *first few* PCs of the covariance matrix whatever the correlation structure. One possible solution to this scaling problem is to ensure that *all* the variables are of the same type (e.g., *all* heights or *all* weights). An alternative procedure most commonly employed, is to standardize *all* the variables so that they have unit variance *prior* to performing the analysis. This is equivalent to finding the PCs of the correlation matrix rather than the covariance matrix. Although the computational procedure is the same, it is important to realize that the components derived from the correlation matrix are different from those that arise from using the covariance matrix. Furthermore, the former set of components does not give any indication about the nature of the latter set, or vice versa.

A major argument for using correlation matrices rather than covariance matrices to obtain the PCs is that the results of analyses for different sets of random variables are more directly comparable than for analyses based on covariance matrices. This is because, as noted before, the PCs based on covariance matrices are sensitive to the units or scales used to measure the variables.

Also, sizes of variances of PCs from different analyses have the same implications for correlation matrices, but not for covariance matrices. Furthermore, patterns of coefficients in the PCs can be readily compared for different correlation matrices to decide whether or not two correlation matrices are giving similar PCs. Such informal comparisons are often less straightforward for covariance matrices.

Nevertheless, the use of covariance matrices to perform PCA does have a general advantage over the use of correlation matrices, and one other advantage which occurs in a special case. The general advantage is that *statistical inference* regarding population PCs based on sample PCs is easier for covariance matrices. However, in practice, PCA is commonly used as a *descriptive* rather than an *inferential* tool. Hence, this advantage becomes less crucial. The second advantage of using covariance matrices occurs when *all* the variables are measured in the same units. In this case, it can be argued that standardizing the variables is equivalent to making an arbitrary choice of the measurement units. This argument about arbitrariness can be applied more generally to the use of correlation matrices, but when the variables are measurements of different types, the use of covariance matrices leads to a more arbitrary choice of the units of measurement. Hence, in this case, the use of correlation matrices should be preferred.

It is important to note that the *scaling problem* does not occur in *Correlation* and *Regression Analysis*. *Correlation coefficients* and the *regression coefficients* between the *response* variable and the *predictor* variables are not dependent on the scales or units of measurement of the variables. *Regression equations* are equivalent whatever scales are used because the *regression line* is chosen so that the sum of squared distances of the data points (or observations) from this line *parallel* to one of the co-ordinate axes is *minimal*. On the other hand, we have already noted that in PCA, the *first* PC is chosen so as to minimize the sum of squares of the *perpendicular* distances of the data points from this line.

The arguments above suggest that a careful decision must be made before a PCA is attempted, as to whether or not *standardization* of the variables is desirable. All the points raised in these arguments should be considered before making such a decision.

PCA is considered a useful tool only when the first few PCs account for most of the variation, so that a few *2-dimensional* scatterplots of principal component scores, may be used to summarize the multivariate data. If this is the case, it can be argued that the 'essential dimensionality' of the data is less than *p*. In other words, if some of the original variables are expressible as linear combinations of the others (because they are highly correlated), then these variables may be effectively 'conveying the same message'. In such cases it is hoped that the first few PCs will be intuitively meaningful and will be useful in subsequent analyses where we can operate with a smaller number of variables.

Perhaps at this point, having used some notation in the description of principal components, it would be appropriate to conclude this section by explaining briefly, the type of notation used throughout the thesis. Bold face upper case letters represent matrices, while underlined upper case letters are used to denote vectors of variables and underlined lower case letters represent vectors of observations. Vectors of coefficients for the linear combinations of variables are denoted by bold face lower case letters. To show the sizes of these matrices and vectors, two subscripts separated by the sign x are used. The first subscript is the number of rows of the matrix or vector, while the second subscript is the number of columns. Individual variables are denoted by plain upper case letters, while their sample means are given by plain lower case letters with bars above them. Double-subscripted plain lower case letters are used to describe the elements of a matrix or a vector. The first subscript represents the row number, while the second subscript is the column number. The transpose of a matrix is given by the appropriate letter representing that matrix with the superscript T. Further notation, perhaps somewhat less general, will be explained as it is introduced in the thesis.

1.3 Variable Selection

1.3.1 General variable selection techniques

The need for a sensible initial choice of variables for inclusion in a scientific investigation is largely self-evident. However, knowledge of the system under investigation may be limited and this may lead to omission of important variables. Needless to say, omission of variables which play a crucial role in the system will lead to serious and in most cases irredeemable consequences. In such cases, the tendency is to start with a very large and exhaustive number p of such *descriptors*. This way, the investigator is guided against ignoring essential variables. One practical example is the *security returns* problem (see Narayanaswamy and Raghavarao (1991)), where it is desired to select the model which best explains the pricing of securities. When initiating such an experiment, the appropriate descriptors which best explain the daily returns variability may be unknown. Hence, it would seem quite reasonable to include, initially, a comprehensive number of variables and to proceed to a variable selection phase to seek the 'best' q $(<p)$ descriptors. In this particular case, the data consist of daily returns on 1200 securities and hence each competing model is tested on 1200 descriptors. Apart from the considerable computational problems some of the included variables may inevitably be redundant, with their presence increasing the level of 'noise' in the data. Consequently, we are faced with the problem of choosing a subset of the available variables, which hopefully will in some sense be almost as informative as the entire set of variables.

Clearly, for future experiments, a model which retains only q variables is more appealing than one with *all* p variables. Inclusion of all the variables is not only wasteful of time and resources but it also makes the model more difficult to understand, and generally, such a model yields less interpretable results. Furthermore, the 'over-fitted' model is sensitive to sampling error, and over-inclusion of variables may impose other harmful effects. For instance, the larger the variable-size, the greater the danger that interesting relationships, effects or patterns will go unnoticed. This is because the power of any statistical tests which might be applied is a strictly decreasing function of the number of variables. Moreover, highly correlated variables add very little to the explanatory power of the data and can create difficulties in the numerical estimation of parameters, giving rise to estimates which are vulnerable to stability. With the 'smaller' model, for a given expenditure of time and effort, a larger number of observations can be included per sample. Large samples are advantageous in the sense that, firstly, an increase in sample size is usually accompanied by a decrease in the error due to sampling and secondly, for sufficiently large samples, the opportunity exists to divide the sample randomly into subgroups prior to the analysis and to proceed to a *cross-validatory* check of how well the model fits. For these reasons, interest in models involving subsets of the original variables is often expressed and can scarcely be over-emphasized.

The identification of essential variables in multivariate analysis arises in a variety of contexts. Areas which have received an extensive amount of investigation are *Multiple Regression* and *Discriminant Analysis*. Here, the optimality criteria on which selection can be based exist most naturally, namely, the residual mean square and the error rate, respectively. Authors who have addressed the regression area include Draper and Smith (1966), Beale *et al.* (1967), Furnival and Wilson (1974), Hocking (1976), McKay (1977, 1979), Krishnaiah (1982) and Brook and Arnold (1985). Minimizing the residual mean square in a regression analysis involving p variables is equivalent to maximizing the multiple correlation of the criterion variable with the remaining p-1 predictors. Beale *et al.* (1967) suggested a method for retaining variables which play a significant role in the system by maximizing the minimum multiple correlation between the selected variables and the discarded variables. This is referred to by Beale *et al.* as *Interdependence Analysis*, of which PCA is a special case. Among the references for variable selection in Discriminant Analysis are McKay (1976), McLachlan (1979, 1980), Murray (1977), Constanza and Afifi (1979), van Ness (1979), McKay and Campbell (1982a, 1982b), Daudin (1986), Ganeshanandam (1987), Ganeshanandam and Krzanowski (1989) and Snapinn and Knoke (1989).

1.3.2 Variable selection in principal component analysis

The usefulness of PCA is often assessed by how well it detects the effective dimensionality of a set of data. This follows from the fact that one of the major objectives of the analysis is to reduce the dimensionality of the data without disturbing the overall features of the sample, i.e., retaining as much of the sample variation (information) as possible. Another related objective of PCA appears to be the identification of new meaningful underlying variables, the PCs. Rotational techniques that transform the linear combinations produced by PCA into more interpretable linear combinations, are available in most multivariate statistical computing software packages. The popularity of these techniques is a realization of the fact that the original linear combinations are often difficult to interpret. Thus, while PCA solves a well-defined mathematical problem, it frequently fails to provide the statistical consumer with useful results. Another major deficiency of PCA in this respect is that, although the dimensionality of the problem may be reduced, each component is a linear combination of *all* of the p original variables. Thus, even though the first few PCs may be considered, interpreting the results using *all* p original variables seems inevitable. Hence, inclusion of *all* the original variables in the analysis is one of the major causes of lack of interpretability of the linear combinations. In the presence of 'noise' variables, interesting patterns or relationships which, otherwise, would be revealed by the PCs, will be less prominent if not completely hidden.

Suppose a PCA meets one of its primary objectives, namely dimensionality reduction and suggests that k PCs are sufficient to model the 'signal' in the data, the remaining p-k PCs being a reflection of the 'noise'. This suggests that the minimum number of variables necessary for recovery of the data structure is k. For example, in the case of the *Alate* data (see Krzanowski (1987) or Jeffers (1967)), 19 variables were measured on each of 40 winged aphids (alate adelges) that had been caught in a light trap. The cross-validatory technique for determining the number of components to be considered in PCA, proposed by Eastment and Krzanowski (1982), chooses the first 4 components suggesting that the remaining 15 dimensions are a reflection of the 'noise' in the data. This, in turn, suggests that a minimum of 4 of the original variables can reproduce the structure of the data with sufficient accuracy. An investigator involved in an analysis of multivariate data with a total of p variables often suspects, and even hopes that a subset of these variables may adequately explain the data. It may well be that the main objective of the investigation is simply to identify the factors of importance in some process or phenomenon. Furthermore, patterns in *2-dimensional* scatterplots of *principal component scores* may be revealed by using a subset of only the best k of the original variables. This would help us understand the data better and enable us to operate with only a few variables in subsequent analyses.

The simplicity of dealing with variables rather than with their linear combinations would seem to justify the increasing need in exploratory multivariate studies, to incorporate variable selection in dimensionality reduction techniques such as PCA. Hence, this leads us to a third objective of PCA, namely the elimination of those variables which contribute relatively little extra information.

Although the literature covers the subject of PCA considerably broadly, the variable subset selection problem has received very little attention, yet it arises frequently in practice as just enlightened. The currently available techniques for selecting variables in PCA can be categorized into three classes; namely, those based on eigenvectors, i.e., the coefficients between the PCs and the original variables, those based on various criteria for optimizing PCs (for example, the property that for a given number of PCs the sum of the variances of the PCs is a maximum; subject to the sum of squares of the coefficients between each PC and the original variables being unity), and those based on *measures of Multivariate Association* (MVA). The criterion for selecting variables has to be linked as closely as possible to the practical aim of PCA. In view of the primary objective of the analysis, the importance of a variable subset should be assessed in terms of whether or not it retains most of the variation present in the set with all p variables. The criteria based on *eigen_analysis*, proposed by Jolliffe (1972, 1973) choose those variables which are highly associated with the first few PCs and reject those which are highly associated with the last few PCs. A subset of the original variables which optimizes one of the criteria,

$$
\begin{array}{lll}
\text{(a)} & \text{Minimize } \prod_{j=1}^{k^*} \theta_j; & \\
\text{(b)} & \text{Minimize } \sum_{j=1}^{k^*} \theta_j; & \\
\text{(c)} & \text{Minimize } \sum_{j=1}^{k^*} \theta_j^2; & (1.7) \\
\text{(d)} & \text{Maximize } \sum_{j=1}^{k^-} \rho_j^2; &
\end{array}
$$

is termed a set of *principal variables* by McCabe (1984). Here, θ_j (j = 1, 2, ...,k^*) are the eigenvalues of the conditional covariance (or correlation) matrix of k^* deleted variables; given the values of k selected variables, ρ_j (j = 1, 2, ...,k^- = min (k, k^*)) are the *canonical correlations* between the set of k^* deleted variables and the set of k selected variables. These criteria actually satisfy the property of maximum variation of the first k PCs. While the techniques just described optimize various optimality criteria, they lack the ability to choose those variables which will 'reproduce', as closely as possible, the general features of the complete data. A technique appropriate for this would optimize the *multivariate association* between PCs produced by the subset of variables and the PCs arising from the entire set of variables. So far, in the literature, the M^2-*Procrustes* technique for variable selection in PCA proposed by Krzanowski (1987) is the only technique that comes close to meeting this requirement. This technique minimizes the discrepancy between corresponding points in the subset configuration and the configuration of the entire set of variables. Hence, in our investigation, emphasis is put on empirical techniques for variable subset selection in PCA, which utilize the idea of *multivariate association.* A technique which uses PCA indirectly, based on *Principal Component Regression* is available in the literature and can be effected by performing PC regression and considering the vector $\hat{\underset{\sim}{\beta}}$ to be the proposed estimator for the regression coefficients. Then we can test whether or not the elements of $\hat{\underset{\sim}{\beta}}$ are significantly different from zero. Those variables whose coefficients are found to be not significantly different from zero, can then be deleted from the model. However, it would seem that this selection technique does not quite choose variables in the 'true'

PCA sense as it requires a response variable in the model; yet in PCA, variables arise on an equal 'footing'.

At this stage, it is very important to note that, the number of variables to be selected (q) and the number of PCs to be used (k) need to be determined *prior* to subset selection. Several methods based on eigenvalues for fixing k have enjoyed popularity for a considerable period of time and recently, Wold (1976, 1978) and Eastment and Krzanowski (1982) proposed methods based on the *cross-validation* technique. For the reason mentioned earlier that, determining the value of k can be viewed as equivalent to finding the number of dimensions required to model the 'signal' in the data with the remaining p-k dimensions being a reflection of the 'noise', it would seem reasonable to set q equal to k.

The optimal approach in selecting variables is to consider *all* possible subsets of the available p variables, and choose those that optimize the selection criterion. In practice, however, this procedure may not be computationally feasible, hence, some form of sequential procedure needs to be considered. In other words, given r ($r \leq q \leq p$) variables in the selected subset, compute the criterion of interest associated with each variable omitted from the subset in turn and *delete* the variable whose omission gives rise to the optimal value of the criterion. This procedure, owing to its nature, is termed *backward elimination*. Another procedure, *forward selection*, operates through *addition* of variables to the existing set of r variables. The criterion of interest is computed for each of the p-r remaining variables added to the set in turn. The variable which gives rise to the optimal value of the criterion is then added to the existing set of r variables. A third sequential procedure combines the above two selection techniques, so that, at each stage the backward elimination is followed by a forward selection. This modified procedure is called *stepwise selection.* In our investigation, selection is effected using the backward elimination procedure. However, we feel that it is necessary to compare the results with those which arise from using the stepwise selection procedure.

A detailed description of the various methods for selecting variables in PCA together with an explicit review of the comparative studies on these selection procedures that exist in the literature are given in Chapter 5 of this thesis.

1.4 Objectives of the current research project

The main aim of the current study is concerned with the selection of important variables of a given *multivariate* set of data for PCA. However, as a lead-in to this, it was decided to address the question of the *number of components* to be used in the selection process. Not only should the components retained in principal component analysis contain as much of the sample features as possible with exclusion of most of the 'noise' in the data, but also the selected subsets of the original variables should satisfy these constraints. Hence, it would seem reasonable to use such components in the selection process and to retain as many of the original variables as the number of these components. Currently existing techniques for determining the number of components to retain in a PCA prove to be unsatisfactory. The techniques based on eigenvalues are *subjective* while the cross-validatory techniques may be *inappropriate*, particularly for small sample cases. For this reason, it was decided to consider a study in which a technique based on the *bootstrap methodology* is introduced and compared with Eastment and

Krzanowski's (1982) cross-validatory technique.

Chapters 2, 3 and 4 of this thesis concentrate entirely on the *choice of the number of components* in PCA. A detailed description of the various existing techniques for determining the number of components and a review of some comparative studies that are available in the literature are presented in Chapter 2. Chapter 3 describes the proposed bootstrap based technique, followed in Chapter 4 by a comprehensive comparative study of this new technique with Eastment and Krzanowski's cross-validatory technique. The data for this study are simulated from Monte Carlo experiments using guidelines from previously published studies.

The remainder of the thesis focuses on variable selection in PCA. Most of the criteria available in the literature for selecting variables in PCA are concerned with the overall features, either of the subset data or of the complete data. Thus, the criteria are based exclusively on covariance or correlation matrices and their eigenvalues/eigenvectors. More appropriate criteria for preserving *multivariate structure* would involve comparing principal component scores from the subset data and those from the full set data. For a given subset size (say q), retained subsets would then be those which maximize the *association* (closeness) between these two configurations. Krzanowski's (1987) M^2-*Procrustes* criterion utilizes this idea, but the study shows that the structure is preserved particularly when the data are grouped. However, our particular interest is to utilize the idea of *multivariate association* to choose those subsets of the original variables which carry whatever *unknown multivariate structure* that may be present in the data. The major objective of our research project is therefore to provide empirical subset selection methods for PCA based on *Canonical*

Correlations and *Graph-Theoretic* criteria; and to compare their behaviour with that of the M^2-*Procrustes* criterion.

A review of literature on variable selection in PCA which includes various selection methods and a few comparative studies of these methods is presented in Chapter 5 of this thesis. Chapter 6 gives a description of the new selection methods proposed in this project. The comparative study in our investigation is based mainly on Monte Carlo simulations, and the behaviours of these methods are compared by means of the performance of the selected variables. This study is described in Chapter 7.

As noted before, the results of a PCA are heavily dependent on the type of *variation* matrix used. It is therefore appropriate to monitor the choice of satisfactory subsets of retained variables according to whether the covariance or the correlation matrix is used for PCA. Jolliffe (1972) considered a ranking scheme appropriate only if the subsets are obtained from PCA using the correlation matrix. Hence, another major aim of the current research project is to obtain a more general ranking scheme to assess the performance of the variable selection methods. This objective is also met in Chapter 7.

Although major conclusions are mainly based on data simulated from normal parent populations by means of Monte Carlo experiments, the techniques discussed above are also applied to some real data sets in Chapter 8. The difficulty encountered in 'real data investigation' is that, unlike with the simulated data, the true 'best' subset of variables is unknown; hence, comparisons among the selection methods can only be achieved through the use of *3-dimensional* and *2-dimensional* scatterplots of principal component scores. This is a subjective approach to the problem at hand.

Overall conclusions of the current research study are presented in Chapter 9. Some general comments on the techniques and suggestions for possible extensions and future developments are also discussed in Chapter 9 to conclude the thesis.

CHAPTER 2
REVIEW OF LITERATURE ON THE CHOICE OF THE NUMBER OF COMPONENTS IN PRINCIPAL COMPONENT ANALYSIS

2.1 Introduction

As seen in Chapter 1, one of the most popular uses of principal component analysis is *dimensionality reduction.* Frequently, just the *first* few of the PCs are sufficient to represent the 'original' data adequately, though in some circumstances, the *last* few, rather than the *first* few are of interest to the researcher. In the present study, however, the traditional idea of attempting to *reduce dimensionality* using PCA is considered, and the possible virtues of the last few PCs are ignored.

If PCA is performed to serve this purpose of dimensionality reduction, it becomes necessary to choose the number of components (say, k) which will 'reproduce' the data with 'sufficient accuracy' and without loosing much sample information. The precise number of components to be retained, however, is often unclear. From the practitioner's point of view, components to retain are those which are a 'true' representation of the 'signal' in the data, the rest being only a reflection of the 'noise'. Several techniques are available in the literature which attempt to select an appropriate subset of the PCs to account for most of the variation in the data. These techniques can be categorized into two classes, namely, those based on *eigenvalues* and those based on *cross-validation.* Descriptions of some existing techniques are given in section 2.2 of this Chapter, while section 2.3 provides a review of some comparative studies of these techniques.

2.2 Some existing techniques for choosing the number of components

2.2.1 Techniques based on eigenvalues

We have already noted in section 1.2 that the eigenvalues can be interpreted as the respective variances of the different components. Suppose $l_1 > l_2 > \ldots > l_p \geq 0$ are the eigenvalues of the covariance or the correlation matrix, corresponding to the principal components PC_1, PC_2, ..., PC_p, respectively. Then, the sum of the variances is given by

$$\sum_{j=1}^{p} \mathrm{Var}(\mathrm{PC}_j) = \sum_{j=1}^{p} l_j. \tag{2.1}$$

It was also noted in section 1.2 that computing the trace of each side of equation (1.3) shows that, in general, the sum of the variances of the original variables is the same as the sum of the variances of the corresponding PCs, i.e.,

$$\begin{aligned} \sum_{j=1}^{p} \mathrm{Var}(X_j) &= \sum_{j=1}^{p} \mathrm{Var}(\mathrm{PC}_j) \\ &= \sum_{j=1}^{p} l_j. \end{aligned} \tag{2.2}$$

Thus, even though the first few PCs may retain most of the variation in the data, the total variation of the system is not changed by the PCA. This important result can also be realized from the *geometrical* point of view. For example, in Figure 1.1, the total variation of the data points in the 'original' space defined by the axes OX_1 and OX_2 is not changed by their projection onto the 'new' space defined by the axes PC_1 and PC_2. Thus, the percentage of the total variation of the system 'accounted for' by the first k PCs is given by

$$P_k = \left[\sum_{j=1}^{k} l_j \bigg/ \sum_{j=1}^{p} l_j \right] \times 100. \qquad (2.3)$$

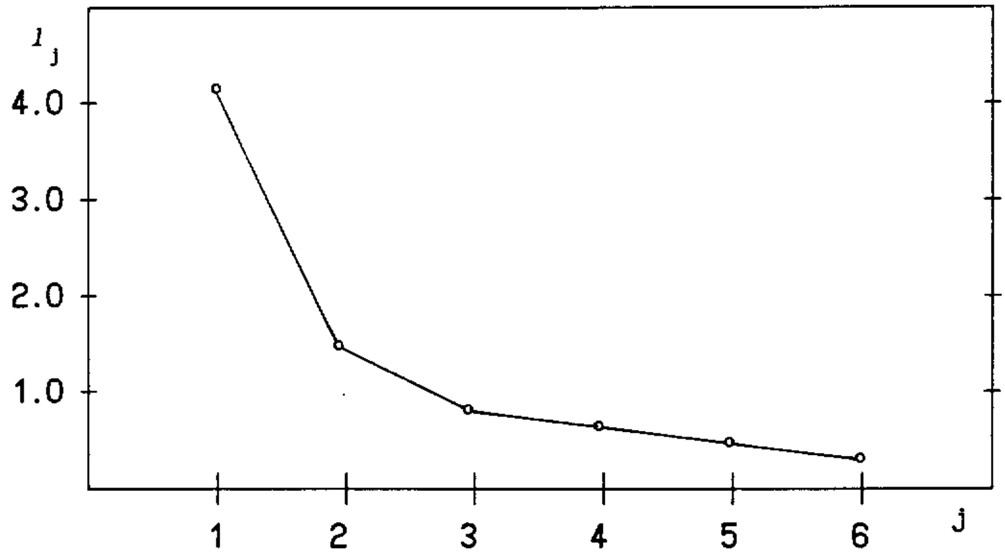
Since the diagonal elements of the correlation matrix are all unity, the sum of these diagonal terms is equal to *p*. Hence, the sum of the variances of the standardized original variables and of the PCs, and the sum of the eigenvalues of the correlation matrix are also equal to *p*. Therefore, the percentage of total variance in the data 'explained' by the first *k* PCs will be given by

$$P_k = \left[\sum_{j=1}^{k} l_j \bigg/ p \right] \times 100. \qquad (2.4)$$

Perhaps this P_k is the most obvious criterion for choosing the appropriate number of components in PCA. A common decision is, to choose the smallest *k* for which $P_k > 75\%$ (say). More strictly, it might be required that $P_k > 80\%$ or even $P_k > 85\%$ before stopping the inclusion of more PCs in the subset of retained PCs. Thus, in this approach, the final decision is *subjective* and somewhat *arbitrary*.

Another popular approach is to construct a *scree diagram* by plotting l_j against j (j = 1, 2, ..., *p*); an idea first introduced by Cattell (1966). A typical pattern is illustrated in Figure 2.1. Here the first two eigenvalues show a sharp drop, followed by a much more gradual decline. It can be argued that those PCs corresponding to the almost 'flat' portion of this graph represent the 'noise' components of the system. Thus, it would seem logical to choose *k* to be the value of j at which the 'elbow' of the scree diagram occurs, for example, in Figure 2.1 this would probably be at $k = j = 3$.

Figure 2.1 *Example of a scree diagram*



An alternative technique which is also in popular use is to exclude those PCs for which $l_j < \bar{l}$, where $\bar{l}$ is the *arithmetic mean* of the eigenvalues, or those for which $l_j < 1$ if the correlation matrix has been used to perform the PCA. The idea behind this rule is that if the data were roughly *spherical*, i.e., the original variables were almost independent, the PCs would be roughly the same as the original variables and each of the PCs would have variance approximately equal to $\bar{l}$. Thus in any PCA, a component with variance less than $\bar{l}$ contains less information than at least one of the original variables, hence, it is considered unsuitable to be retained.

A similar criterion based on the 'broken stick' model can also be used to determine k. If we imagine a stick with unit length to be broken, at random, into p segments, then it can be shown that the expected length of the k-th longest segment is given by

$$l_k^* = \frac{1}{p} \sum_{j=k}^{p} \frac{1}{j} \; . \qquad (2.5)$$

Thus, the k-th component is retained if $l_j > l_k^*$ (j = 1, 2, ..., p) and is deleted otherwise.

The final eigenvalue-based criterion for deciding on the value of k makes use of Bartlett's test for the null hypothesis, $H_0 : l_{j+1} = l_{j+2} = \ldots = l_p$ versus the general alternative, H_1 : at least two of the last p-j eigenvalues are unequal. H_0 is rejected at significance level α, if

$$n' \left[(p-j) \log_e \tilde{l} - \sum_{k=j+1}^{p} \log_e l_k \right] \geq \chi^2_{\nu;\alpha} \qquad (2.6)$$

where

$$n' = n - \left[(2p + 11) \Big/ 6\right],$$

$$\tilde{l} = \sum_{k=j+1}^{p} l_k \,/(p - j),$$

and

$$\nu = 1/2\ (p - j + 2)\ (p - j - 1).$$

If the null hypothesis is true, then it can be argued that the last p-j PCs do not contribute significantly to the total variation; hence, these PCs can be deleted without serious loss of information. If k is the number of PCs which represent the 'signal' in the data, then the test can be used sequentially to find k.

Nevertheless, it must be pointed out that the methods above are *ad hoc*, with no formal statistical justification. Only the facts that they are intuitively plausible and that they often work in practice, would seem to explain their popularity. One disadvantage of the method based on Bartlett's test is that it is dependent on the assumption of *multivariate normality* of the data matrix **X** which is often unrealistic. Furthermore, in this method, the overall significance level is completely unknown because the number of tests to be performed is *random*, and these tests are not independent of each other. Hence, the method could be added to the list of *ad hoc* rules. In fact, it can be seen as somewhat similar to the choice of PCs according to Cattell's scree graph. Looking for two consecutive eigenvalues which are the same in the sense suggested by Bartlett's test, corresponds to looking for the 'elbow' in Cattell's scree graph. However, techniques based on more statistically motivated principles are available in the literature and these are described in section 2.2.2 below.

2.2.2 Cross-validatory techniques

Cross-validation is a statistical tool which is generally used to obtain unbiased or nearly unbiased non-parametric estimators of prediction error. The standard procedure consists of dividing the data into subgroups which are then deleted from the data, one at a time; and for each competing model, the deleted values are predicted using the remainder of the data. An overall measure of the degree of association between actual and predicted values for each model is then computed, and the model which optimizes this measure is chosen.

Consider the first rule in section 2.2.1, in which the number of PCs is chosen on the basis of the percentage of the total variance 'explained' by the retained PCs. This procedure is, in a sense, equivalent to looking at the *spectral decomposition* of the covariance (or correlation) matrix or looking at the *Singular Value*

Decomposition (SVD) of the 'mean centered' data matrix **X**. In either case, deciding upon the number of terms to be included in the decomposition in order to obtain a good fit, is closely related to looking for the smallest integer k, for which $P_k > P^*$, where P^* is the chosen cut-off point. However, using the rule based on the SVD of **X** rather than one based solely on the eigenvalues of the covariance (or correlation) matrix, has two major advantages. Firstly, the former rule utilizes the entire information available in the data and it leads to a particular choice of k which is unique to the given data matrix **X**; in contrast to the latter which chooses a value of k unique only to the covariance (or correlation) matrix. In other words, there may be several distinct data matrices **X** yielding almost identical covariance (or correlation) matrices; and the choice of k for each of these cases using the latter rule would be the same (as each is associated with the *same* set of eigenvalues), whereas the former rule applied to these data matrices would probably lead to different choices of k. Secondly, implicit in the approach based on the SVD of the data matrix **X** is the idea that the choice of k should indicate the number of PCs to be retained in order to 'reproduce' **X** with sufficient accuracy. To illustrate this point, we shall first establish the relationship between PCA and the SVD of **X**.

Consider equations (1.1) and (1.4) in section 1.2. Equation (1.1) gives the principal component scores (PC scores) corresponding to the j-th PC, while in equation (1.4) is the SVD of the data matrix **X**. Using equation (1.1), the matrix of PC scores which includes all the available p PCs is given by **XV**, where **V** is the matrix of the coefficients between the PCs and the original variables. As noted before, these PC scores can also be obtained by multiplying U by S, were U and S arise from the SVD of X as given by equation (1.4). Thus, XV = US, and since $V^TV = VV^T = I_p$, post-multiplying this equation by V^T yields X = USV^T which is the SVD of X (as given by equation (1.4)). Hence, PCA is equivalent to the SVD of the data matrix X.

The equation above enables us to write each (i,j)-th element of X as in equation (1.6) (i = 1, 2, ..., *n* and j = 1, 2, ..., *p*). Thus, retaining only the first *k* PCs can be seen as equivalent to *modeling* the elements of X by the model

$$x_{ij} = \sum_{t=1}^{k} u_{it}\ s_t\ v_{tj} + \varepsilon_{ij}. \tag{2.7}$$

This type of model can be referred to as a *Principal Component* model or simply a PC model. Here, u_{it} are the elements in the i-th row of U, s_t are the diagonal elements of S, v_{tj} are the elements in the j-th column of V^T in equation (1.4) and ε_{ij} are the residuals. Eastment and Krzanowski (1982) suggested a method which utilizes the PC model above to choose the number of PCs to be retained. Wold (1976, 1978), on the other hand, worked with a PC model slightly different from the one in equation (2.7). This model was derived by Wold and Andersson (1973). In their derivation of the model, the latter authors also considered the fact that the PCs are linear combinations of the 'original' variables which are uncorrelated and ordered in such a way that the *first few* retain most of the variation or information present in the data. Following is a brief outline of the derivation.

Suppose that on each of *n* objects, the values of *p* variables have been measured giving the data matrix X with elements x_{ij}. If all the objects come from the same *population*, the values of each variable j are the same for all the objects except for small deviations ε_{ij} due to errors of measurements. Hence, for this simple case, the data can be described by the model

$$x_{ij} = \alpha_j + \varepsilon_{ij}. \tag{2.8}$$

Wold (1976, 1978) suggested that the effect α_j should be the mean of the j-th variable. However, equation (2.8) is often unrealistically simple. The underlying assumption that ‘the objects are so similar that they are virtually *identical*’ is hardly fulfilled in practice. If it is assumed that the objects differ slightly from each other, the model becomes

$$x_{ij} = \alpha_j + u_i v_j + \varepsilon_{ij}, \tag{2.9}$$

and a larger variation between the objects leads to the model

$$x_{ij} = \alpha_j + \sum_{t=1}^{k} u_{it} v_{tj} + \varepsilon_{ij}. \tag{2.10}$$

Here, u_i and u_{it} are the elements in the i-th row of **U** and v_j and v_{tj} are the elements in the j-th column of $\mathbf{V}^T$ in equation (1.4). It can be seen that, equation (2.10) is a PC model in which the observed variable x_{ij} is linearly related to the PC scores u_{it} using the weights v_{tj}. Here, u_{it} are referred to as PC scores because of the following reason: Recall from section 1.2 that the matrix of PC scores is given by **N** = **US**, where **S** is the diagonal matrix with diagonal elements $s_1 \geq s_2 \geq \ldots \geq s_p$ which are the non-negative square-roots of $\mathbf{X}^T\mathbf{X}$ or $\mathbf{X}\mathbf{X}^T$ as defined in equation (1.4). Post-multiplying each side of this equation by $\mathbf{S}^{-1}$ yields $\mathbf{U} = \mathbf{N}\mathbf{S}^{-1}$.

It was also noted in section 1.2 that the eigenvalues of the *sample covariance* matrix are given by the diagonal elements of $(n-1)^{-1}\mathbf{S}^2$, so that the variance of the PC scores for the t-th PC is $(n-1)^{-1}s_t^2$, $t = 1, 2, \ldots, p$. Hence, the scores u_{it} given by **U** are simply those given by **N**, but standardized to have variance $(n-1)^{-1}s_t$, $t = 1, 2, \ldots, p$. It can also be seen that, equations (2.8) - (2.10) are PC models with the number of retained PCs being 0, 1 and k, respectively. Note that when the number of retained PCs is set to zero, a case given by equation (2.8), it is assumed that *all* of the variation in the data is due to chance or is random, so that no PCA is performed. Further note that it is essential to consider the case $k = 0$ as this enables us to compute the variation or information supplied by *only* the *first* PC. This variation is the amount of reduction in the error sum of squares due to fitting the *first* PC and can be obtained by subtracting the sum of squares of the errors when $k = 1$ from the sum of squares of the errors when $k = 0$.

Setting ε_{ij} to zero in each of the two PC models given by equations (2.7) and (2.10) is equivalent to estimating (or *predicting*) the data using only the first k PCs. The sum of squared differences between the predicted and the actual values can then be calculated and used to *assess* the accuracy of the prediction process. If the original data are predicted with 'sufficient' accuracy, then it can be argued that the first k PCs represent these data adequately. Hence, the problem is reduced to choosing the *model* with the *least* number of components for adequate representation of the data. Therefore, this approach is less *subjective* than the one based solely on the structure of the eigenvalues.

However, the computation of u_{it}, s_t and v_{tj} in equations (2.7) and (2.10) utilizes the values x_{ij} itself. Hence, this prediction of

x_{ij} is *over-optimistic*. To obviate this clearly unsatisfactory feature, Wold (1976, 1978) and Eastment and Krzanowski (1982) used the idea of *cross-validation*. The principle of the cross-validation approach is to ensure that each data points that are predicted are not used in the estimation of the parameters, hence to avoid the *over-optimism* in the prediction process. Applied to the problem at hand, this approach suggests performing the SVD with part of the data deleted from the data matrix **X**. Quantities equivalent to u_{it}, s_t and v_{tj} arising from this SVD are then used to predict the deleted values.

To perform the cross-validatory prediction of x_{ij}, Wold (1976, 1978) suggested that, initially, the data matrix **X** should be divided into *g* groups. He recommended that *g* should be between 5 and 10, inclusive, and must not be a divisor of *p* and no group should contain the majority of the elements in any row or column. Quantities equivalent to u_{it} and v_{tj} are then computed *g* times, with each group of data deleted from **X**, one at a time. For a fixed *k* (*k* = 1, 2, ..., etc.), the number of PCs, the model in equation (2.10) is fitted to the data arising from the deletion of the h-th group, and the estimates obtained are used to predict the data in the h-th group, (h = 1, 2, ..., *g*). It should be noted that when *k* = 0, the model in equation (2.8) is the suitable choice for the analysis. With respect to the choice of the number of PCs, for a fixed *k* (*k* = 0, 1, 2, ..., etc.), the sum of squares of the prediction errors, with all *g* groups considered, is calculated and these sums of squares are then examined over the successive values of *k* to determine the optimal *k* at which adequate prediction of the data first occurs. This happens when the inclusion of the (*k*+1)-th PC and the successive PCs to the model in equation (2.10) no longer significantly improves the representation of the data, suggesting that the first k PCs are sufficient. Hence, some suitable test-statistic can be used over the consecutive values of k to determine the minimum value of k which corresponds to an optimal representation of the data.

Wold (1978) noted that whether or not the parameters α_j should be cross-validated depends on the type of application for the analysis and that the decision should be left to the investigator. It is also important to note that if the analysis precludes the cross-validation of α_j and if the data matrix X has been 'mean-centered', then the models in equations (2.8) - (2.10) do not contain α_j. The cross-validatory procedure arises in two forms, *single-cross* and *double-cross*. In the single-cross procedure, *assessment* (validation) relies on some process external to the system, whereas in the double-cross procedure, assessment results directly from the analysis itself. Following is an algorithm for the cross-validatory procedure suggested by wold (1976,1978), applicable to both single-cross and double-cross apart from the differences which arise at the assessment stage. In this algorithm, it is assumed that the analysis precludes the cross-validation of α_j and that the data matrix X has first been 'mean-centered'. Hence, the parameters α_j in the models in equations (2.8) - (2.10) are equal to zero.

Step 1: Divide the set of observations (data matrix X) into g groups ($5 \leq g \leq 10$).

Step 2: For each value of h (h = 1, 2, ..., g), perform steps 3 - 6.

Step 3: Omit the objects in group h from X and denote the resulting data matrix by $\mathbf{X}_h^{(-)}$.

Step 4: Perform the SVD of $X_h^{(-)}$ to obtain the estimates of the parameters u_{it} and v_{tj}.

Step 5: For a fixed k (k = 0, 1, 2, ..., p-2), fit the models in equations (2.8) and (2.10) when k = 0 and k = 1, 2, ..., p-2, respectively, and use the quantities in step 4 to compute estimates of the deleted part of the data.

Step 6: Compute the deviations ε_{ij} for each value of k for the deleted objects and denote the resulting matrix by $E_h^{(k)}$, k = 1, 2, ..., p-2.

Step 7: For a fixed k, concatenate matrices $E_1^{(k)}$, $E_2^{(k}$, ..., $E_g^{(k)}$ (corresponding to each deleted group) vertically to form the overall matrix of deviations $E^{(k)}$ and then compute PRESS(k), the sum of squares of these deviations. In other words, compute PRESS(k) = $\text{Trace}\left\{\left[E^{(k)}\right]^T\left[E^{(k)}\right]\right\}$. The notation PRESS, stands for PREdiction Sum of Squares, and this is taken in a similar sense as in linear *regression*. These PRESS(k) values are a measure of how well the model in equation (2.10) predicts the data for each k.

In the case of the single-cross procedure (see Wold (1976)), F-tests are then made on the statistic

$$F = \frac{\text{PRESS}(k-1) - \text{PRESS}(k)}{D_1} \div \frac{\text{PRESS}(k)}{D_2}, \qquad (2.11)$$

to decide whether or not the addition of the k-th component is significant. Here, $D_1 = n$, the number of degrees of freedom required to fit the k-th component and $D_2 = n(p-k-1)$, the number of degrees of freedom remaining after fitting the k-th component. The disadvantage of this criterion, however, is that it is dependent on normality assumptions (on the data matrix **X**) underlying the F-tests. The double-cross procedure (see Wold (1978)) requires that the sum of squared differences between observed and estimated data points (based on all the data, using the first (k-1) PCs)

$$S_{k-1} = \sum_{i=1}^{n} \sum_{j=1}^{p} \left[x_{ij}^{(k-1)} - x_{ij}\right]^2, \qquad (2.12)$$

be computed. S_{k-1} is then compared with PRESS(k) using the ratio

$$R = \text{PRESS}(k) \Big/ S_{k-1}. \qquad (2.13)$$

If R < 1, the implication is that better prediction can be achieved by using the first k rather than the first (k-1) PCs, so that the k-th PC should be included; otherwise inclusion of extra PCs is stopped. The optimal value of k then becomes the 'best' choice of the number of PCs for adequate representation of the data.

The method suggested by Eastment and Krzanowski (1982) utilizes the ideas described above. However, these authors argued that the computational nature of Wold's (1976, 1978) methods precludes the extraction of the maximum amount of cross-validatory information in the sense defined by Stone (1974). According to Stone (1974), the principle of cross-validation is to ensure that the same data points are not used in both the *prediction* and *assessment* stages, but nevertheless to use as much of the original data as possible in predicting each x_{ij}. This may be achieved by omitting a single element from the data rather than a subset or group in the cross-validation procedure. This approach is often called the *leave-one-out* technique. Eastment and Krzanowski (1982) attempted to meet Stone's requirement, hence, in their method, the quantity corresponding to v_{tj} in equation (2.7) is based on the data set with just the i-th observation deleted, the estimate of u_{it} is calculated with only the j-th variable deleted, and s_t is estimated from the combined information from the two cases above (i.e., with the i-th observation and the j-th variable omitted). Furthermore, prediction is effected by using the widely available SVD algorithms which are computationally fast, particularly when the recent algorithms for updating the SVD of a matrix on the addition or deletion of a row or column are incorporated (see Bunch and Nielsen (1978) and Bunch, Nielsen and Sorensen (1978)). On the other hand, computations in the methods suggested by Wold (1976, 1978) are performed using the *Non-Linear Iterative Partial Least Squares* (NIPALS) algorithm (see Wold and Lyttkens (1969)) which is slow and not as universally available as algorithms for the SVD of a matrix. A more detailed account of Eastment and Krzanowski's technique is presented in the next section.

2.2.3 Review of Eastment and Krzanowski's technique

In this section, we discuss, exclusively, the cross-validatory technique for choosing the number of components, *k*, proposed by Eastment and Krzanowski (1982). The reason for this is to show the nature of the computational procedure in more detail. This way, the use of the *Singular Value Decomposition* (SVD) to obtain the PCs and the cross-validation procedure to predict each element of the data matrix using *only* the first *k* PCs, and hence the choice of optimal *k* can be better understood. Following is the general outline of the computational procedure.

Consider the $\mathbf{X}_{(nxp)}$ data matrix obtained by observing n objects on p variables, 'mean-centered' and appropriately scaled. Associated with a given value of k is the predictor $\hat{\mathbf{X}}^{(k)}$; an estimate of $\mathbf{X}$ which arises from fitting only the first k PCs. Thus the prediction model is given by

$$\mathbf{X} = \hat{\mathbf{X}}^{(k)} + \mathbf{E}^{(k)}, \tag{2.14}$$

where $\mathbf{E}^{(k)}$ is the (nxp) matrix of error scores and $k = 0, 1, 2, \ldots,$ etc. Each row of $\mathbf{E}^{(k)}$ has a multivariate normal distribution under the usual distributional assumptions. The errors in any row of $\mathbf{E}^{(k)}$ are statistically independent of the errors in any other row since the rows of a data matrix generally represent randomly sampled subjects. An outline of the prediction process using the PCs is given in Figure 2.2 below.

Figure 2.2 *Outline of the prediction of the data matrix* **X** *using only the first k PCs.*

$$\begin{tikzcd}
\mathbf{X} \arrow[rr, "PCA", "Estimate"'] \arrow[dr] & & \hat{\mathbf{X}}^{(k)} \arrow[dl] \\
& \mathrm{PRESS}(k) &
\end{tikzcd}$$

Here,

$$\text{PRESS}(k) = \text{Trace}\left\{ \left[X - \hat{X}^{(k)}\right]^{T} \left[X - \hat{X}^{(k)}\right] \right\}$$

$$= \text{Trace}\left\{ \left[E^{(k)}\right]^{T} \left[E^{(k)}\right] \right\}, \qquad (2.15)$$

(k = 0, 1, 2, ..., etc.) and some suitable function of these PRESS values is considered in order to choose the optimum value of k.

As noted before, the singular value decomposition of the data matrix (see equation (1.4)) enables us to represent x_{ij}, the (i,j)-th element of the data matrix **X** in the form given by equation (1.6). Clearly, the elements of **X** can also be written in the form given by the model in equation (2.7) and this is equivalent to estimating the data using only the first k PCs. It was also noted that the technique of Eastment and Krzanowski (1982) utilizes this model and the cross-validation procedure. Details of the reasoning behind the approach adopted by these authors, such as the use of cross-validation, have already been presented in the previous section. Hence, presented below is the prediction procedure illustrated in more detail and the use of the *prediction sum of squares* to choose the optimal k at which adequate prediction of the data first occurs.

Let $\hat{\mathbf{U}}_{(nxp-1)}$, $\hat{\mathbf{S}}_{(p-1xp-1)}$ and $\hat{\mathbf{V}}_{(p-1xp-1)}$ be the matrices which arise from the SVD of the data matrix **X** without the j-th column. Here, $\hat{\mathbf{U}}$, $\hat{\mathbf{S}}$ and $\hat{\mathbf{V}}$ are respectively equivalent to **U**, **S**, and **V** in the SVD of the complete data matrix **X** in equation (1.4). Then by using $\mathbf{X}^{(-.j)}$ to denote **X** with the j-th column omitted, we can write

$$\mathbf{X}^{(-.j)} = \hat{\mathbf{U}}\ \hat{\mathbf{S}}\ \hat{\mathbf{V}}^{T}. \qquad (2.16)$$

Similarly, let $\bar{\mathbf{U}}_{(n-1xp)}$, $\bar{\mathbf{S}}_{(pxp)}$ and $\bar{\mathbf{V}}_{(pxp)}$ respectively correspond to **U**, **S**, and **V** when the i-th row of **X** is omitted. Once again, if we use $\mathbf{X}^{(-i.)}$ to denote **X** without the i-th row, then we can write

$$\mathbf{X}^{(-i.)} = \bar{\mathbf{U}}\ \bar{\mathbf{S}}\ \bar{\mathbf{V}}^{\mathrm{T}}. \tag{2.17}$$

Now using equations (2.16) and (2.17), the predictor $\hat{x}^{(k)}_{ij}$ of x_{ij} is given by

$$\begin{aligned}
\hat{x}^{(k)}_{ij} &= \hat{\mathbf{u}}^{(k)}_{i}\ \sqrt{\hat{\mathbf{s}}}^{(k)}\ \sqrt{\bar{\mathbf{s}}}^{(k)}\ \bar{\mathbf{v}}^{(k)}_{j} \\
&= \begin{bmatrix} \hat{u}_{i1} & \hat{u}_{i2} & \cdots & \hat{u}_{ik} \end{bmatrix} \begin{bmatrix} \sqrt{\hat{s}_1} \\ \sqrt{\hat{s}_2} \\ \vdots \\ \sqrt{\hat{s}_k} \end{bmatrix} \begin{bmatrix} \sqrt{\bar{s}_1} & \sqrt{\bar{s}_2} & \cdots & \sqrt{\bar{s}_k} \end{bmatrix} \begin{bmatrix} \bar{v}_{1j} \\ \bar{v}_{2j} \\ \vdots \\ \bar{v}_{kj} \end{bmatrix} \\
&= \sum_{t=1}^{k} \left(\hat{u}_{it}\ \sqrt{\hat{s}_t}\right)\left(\sqrt{\bar{s}_t}\ \bar{v}_{tj}\right).
\end{aligned} \tag{2.18}$$

Here, $\hat{\mathbf{u}}^{(k)}_{i}$ is the vector of the first k elements in the i-th row of $\hat{\mathbf{U}}$, $\sqrt{\hat{\mathbf{s}}}^{(k)}$ consists of the square-roots of the first k elements of the diagonal of $\hat{\mathbf{S}}$, $\sqrt{\bar{\mathbf{s}}}^{(k)}$ consists of the square-roots of the first k elements of the diagonal of $\bar{\mathbf{S}}$, and $\bar{\mathbf{v}}^{(k)}_{j}$ is the vector of the first k elements in the j-th column of $\bar{\mathbf{V}}^{\mathrm{T}}$.

The SVD is unique except for corresponding sign changes in **U** and **V** in equation (1.4). Since the matrices $\hat{\mathbf{U}}$ and $\bar{\mathbf{V}}$ (necessary in equation (2.18)) are found independently, the sign of $\hat{x}^{(k)}_{ij}$ is arbitrarily negative or positive. To overcome this, it is ensured at each stage of the prediction process that

$$\operatorname{sign}\left(\hat{u}_{it}\sqrt{\hat{s}_t}\sqrt{\bar{s}_t}\,\bar{v}_{tj}\right) = \operatorname{sign}\left(u_{it}s_tv_{tj}\right) \tag{2.19}$$

holds. Here, $u_{it}s_tv_{tj}$ arises from the singular value decomposition of the complete data matrix X as given by equation (1.6). After estimating all the elements, x_{ij} of X, PRESS(k) (k being consecutively, 0, 1, 2, ..., p-1) can be computed and the optimal k is the *largest* value of k at which the statistic

$$W = \frac{\text{PRESS}(k-1) - \text{PRESS}(k)}{D_k} \div \frac{\text{PRESS}(k)}{D_r}, \tag{2.20}$$

is greater than *unity*. Here, $D_k = n+p-2k$ is the number of degrees of freedom required to fit the k-th component and $D_r = p(n-1) + k(k+1-n-p)$ is the number of degrees of freedom remaining after fitting the k-th component. Note that both F (see equation (2.11) in section 2.2.2) and W can be interpreted as the increase in predictive information supplied by the k-th component, divided by the average information in each of the remaining components. The optimal k value above, then becomes the choice of the number of components to be retained in order to represent the data adequately.

2.3 Comparative studies on choosing the number of components

In the literature, assessment of the performance of the majority of the methods described in section 2.2 above, is mainly through analyses of real data; whether or not they give sensible results in a particular application, and whether or not the user understands the conclusions and finds them useful. As discussed in section 2.2, the criterion B (i.e., the choice according to Bartlett's test), though more formalized, is somewhat similar to the formulation of Cattell's scree graph. Looking for the 'elbow' in the graph corresponds to trying to identify the first two consecutive eigenvalues which are the same in the sense suggested by the null hypothesis in Bartlett's test. Cattell's scree graph differs from criterion B in that it starts from the largest eigenvalue, and compares successive eigenvalues in a pairwise fashion, while criterion B compares blocks of two, three, four, etc. The computational nature of W (i.e., the criterion proposed by Eastment and Krzanowski (1982)) suggests that it looks for 'large gaps' among the set of eigenvalues arranged in the order of decreasing magnitude and this process may be equated with using Cattell's scree graph. Hence, W can be seen as lending objectivity to Cattell's approach. When the chromatography retention index data published by McReynolds (1970) was subjected to R (i.e., the criterion suggested by Wold (1978)) and W, with and without outliers, R consistently chose fewer PCs than W.

Perhaps the most comprehensive comparative study of the various criteria for determining *k*, based on Monte Carlo experiments, is due to Krzanowski (1983). This author noted that criterion E (i.e., choosing PCs whose associated eigenvalues exceed the average of all the eigenvalues) always yields a choice of *k* which is less than or equal to the choice using the criterion T (i.e., retaining PCs whose cumulative variance exceeds 75% of the trace of the covariance matrix), while criterion B usually leads to a much larger number of selected PCs than really necessary. Choices according to criteria E and T depend minimally on sample size while criterion B gives smaller choices of *k* for small sample cases than for large ones. With the exception of data derived from diagonal dispersion matrices where the behaviour of W is very similar to that of B; for the majority of dispersion structures, W behaves almost like E. Krzanowski (1983) also found in his study that sample size is not a critical factor in the performance of W. He also noted that when W fails to choose the appropriate number of PCs, it tends to choose less components than the required number and consistently chooses less components than those chosen by the criterion T.

As mentioned in section 2.1.1, the criteria for choosing the number of PCs based on eigenvalues have no formal statistical justification. It would therefore seem that the only techniques with a more formal statistical base which are appropriate for this purpose are those described by Wold (1976, 1978) and Eastment and Krzanowski (1983). However, although the technique suggested by Eastment and Krzanowski (1982) is seen to be an improved version of the techniques suggested by Wold (1976, 1978) in the sense that it attempts to extract the maximum amount of cross-validatory information, perhaps the results it yields are sensible only as regards to cross-validation. In other words, the use of a different *resampling* procedure such as the *bootstrap methodology* which has worked well with various *multivariate techniques* such as *Discriminant Analysis* (DA) in many practical situations, could yield better results. Hence, in the current research project, we propose a method for choosing the number of PCs, based on the ideas of *bootstrapping*.

CHAPTER 3

BOOTSTRAP ESTIMATION OF THE NUMBER OF COMPONENTS

3.1 Introduction

Our exploration of the choice of the number of components (k) in the current research project includes introducing a new technique based on *bootstrapping* whose description is presented in this Chapter. There are three main reasons for this. Firstly, as mentioned in Chapter 2, in the literature, the only techniques for choosing k which seem to have a formal statistical justification are, the technique based on Bartlett's test (see section 2.2.1) and the cross-validatory techniques which were suggested by Wold (1976, 1978) and by Eastment and Krzanowski (1982) (see sections 2.2.2 and 2.2.3). Bartlett's test is based on distributional assumptions which are often unrealistic, and, in any case, it seems to retain more PCs than necessary in practice, hence leaves us with the choice of only the cross-validatory techniques. Secondly, although the cross-validatory techniques are less *ad hoc* than those based on eigenvalues, cross-validation suffers from a few but serious undesirable features. The procedure involves deletion of part of the sample and this can induce instability in the prediction process leading to estimates which are sensitive to sampling error and therefore unreliable, particularly in cases where sample size is small.

In order to illustrate this deficiency, we conducted a pilot study in which a computer program corresponding to the technique proposed by Eastment and Krzanowski (1982) was written in the matrix programming software GAUSS (1988) and used to analyse several real data sets. Among the sets of data analysed was the *Alate* data whose description can be found in section 1.3.2. As mentioned in this

Table 3.1 *Results of applying the Eastment and Krzanowski (1982) cross-validatory choice of the number of components on the Alate data.*

Number of PCs (k)	Eigenvalues of the correlation matrix	Cumulative percentage of variance explained	W
1	13.838	72.8	26.288
2	2.368	85.3	5.967
3	0.748	89.2	0.204
4	0.505	91.9	1.033
5	0.278	93.4	-0.395
6	0.263	94.7	0.576
7	0.183	95.7	0.119
8	0.159	96.5	0.116
9	0.145	97.3	0.232
10	0.134	98.0	0.444
11	0.092	98.5	0.047
12	0.078	98.9	0.044
13	0.072	99.3	0.296
14	0.044	99.5	0.145
15	0.032	99.7	0.064
16	0.024	99.8	0.045
17	0.019	99.9	0.024
18	0.013	1.00	0.012
19	0.004	1.00	—

section, the Alate data consists of 19 variables, each measured on 40 aphids (alate adelges). It is evident from the disparate nature of these variables that the data should be standardized before PCA. Earlier analysis, based on eigenvalues, performed by Jeffers (1967) suggested that the first two PCs could describe the data sufficiently. Krzanowski (1987) analysed the data using the cross-validatory scheme of Eastment and Krzanowski (1982) and concluded that it would take at least the first four components to represent the data adequately, but he did not supply the details of the analysis (i.e., the values of the W statistic corresponding to each component, etc.). Hence, we chose to re-analyse this set of data in order to obtain more useful results and these are presented in Table 3.1.

The computational nature of the method (see equation (2.20)), suggests that for all k, k = 1, 2, ..., p-1, the statistic W should be non-negative. This follows from equation (2.7) which suggests that, as the number of components (k) *increases*, the error term ε_{ij} should *decrease* and therefore PRESS(k) should be a strictly decreasing monotone function of the number of components, which, in turn, suggests that W should take non-negative values.

Despite this fact, the *fifth* component in Table 3.1 yields a negative value of the W statistic, and this means that PRESS(k) at k = 5 is greater than PRESS(k) at k = 4 which is an unsatisfactory feature of the cross-validatory procedure, possibly induced by the deletion involved at the prediction stage. However, among all the data sets considered, this feature was absent in cases where the size of the sample was approximately 100 or more and in all cases where covariance matrices were used to perform the PCA. For example, consider the *Venezuela data* which consists of 8 variables (examination marks) observed on each of 466 Venezuela students who were sponsored by the *British Council* to study *English Language* for a year at colleges in north of *England*. These students were distributed among 10 colleges. We considered only a subset of this data set which describes 135 of students from *three* of the colleges selected at random. From the disparate nature of the the variables, it was evident that the data should be standardized before the analysis. Eastment and Krzanowski's (1982) technique suggested that the first 2 PCs are sufficient for adequate representation of this set of data. It was also noticed that none of the components produced negative W values. However, when the same analysis was performed on the data on *only* 39 students from *one* of the three colleges above, 3 components were retained but the *fourth* component showed a negative value of W. Next we considered the GCRI (*Gas Chromatography Retention Index*) data published by McReynolds (1970). This data set consists of retention indices of 10 representative test compounds (variables) measured on 226 different liquid phases (observations). The variables did not reflect different scales of measurement; hence it was more appropriate to perform PCA using the covariance matrix. When the data were subjected to Eastment and Krzanowski's (1982) technique, the first 4 components were seen to be sufficient for adequate representation of the data and none of the components yielded a negative value for W. Analysis of *only* 32 observations of the GCRI data chosen at random showed that 3 components are sufficient to represent the data adequately, and all the values of the W statistic were positive. The results above and those from other cases considered led us to believe that the negative values of W occur only in cases where correlation matrices are used to perform PCA of small-sized data sets in Eastment and Krzanowski's cross-validatory technique.

Thirdly, the *bootstrap methodology*, in addition to sharing the desirable qualities present in the other *resampling plans* (i.e., cross-validation, jackknifing, etc.), has been shown to out-perform these resampling techniques in many situations, particularly for small sample sizes, for example, when used with multivariate methods such as *Discriminant Analysis* where it is desired to find the best estimator for the *true error rate of misclassification*. One plausible explanation for the superiority of the bootstrap over cross-validation and jackknifing for small samples is that, while in the latter techniques only a total of n samples of size n-1 are available, a much larger number of distinct samples of size n can be obtained using the bootstrap technique. Hence, the bootstrap sample resembles a typical sample from the *universe* sampled more closely than it resembles the original sample. Therefore, the estimates obtained by using the bootstrap technique are more stable and more general than the cross-validatory and jackknife estimates.

For clarity, descriptions of the general bootstrap methodology, are presented in section 3.4. *prior* to the presentation of the proposed bootstrap technique for choosing the number of components which is described fully in section 3.5. Furthermore, since the proposed bootstrap technique utilizes the *multivariate linear regression model* and the ideas from *principal component regression*, the theme behind multivariate regression modeling is discussed in section 3.2, while descriptive details of the principal component regression technique are presented in section 3.3.

3.2 The multivariate linear regression model

Multivariate linear regression is a straightforward generalization of *multiple linear regression*. Descriptions of the former technique can be found in Cramer and Nicewander (1979), Mardia, *et al.* (1979), McKay (1979), Muller (1982), Gittins (1985), and many more. McKay (1979) considered the study of variable selection in *Multivariate Regression Analysis*.

Consider two sets of jointly distributed variables $\underline{Z}_{(p\times1)}$ and $\underline{Y}_{(q\times1)}$ and let $\mathbf{Z}_{(n\times p)}$ and $\mathbf{Y}_{(n\times q)}$ be the respective data matrices from a sample of n observations measured on these variables. Suppose that the columns of $\mathbf{Y}$ represent the 'dependent' variables which are to be explained in terms of the 'independent' variables given by the columns of $\mathbf{Z}$. In other words, it is desired to predict the $\underline{Y}$ variables simultaneously from the $\underline{Z}$ variables. Multivariate regression attempts to address this problem through the model,

$$\mathbf{Y} = \mathbf{ZB} + \mathbf{E}, \tag{3.1}$$

where, **Y** consists of q response variables each measured on n subjects, **Z** consists of p predictor variables (for convenience, we assume that $q \le p$ and that **Z** and **Y** are of full rank), $\mathbf{B}_{(p\text{x}q)}$ is a matrix of regression coefficients, and $\mathbf{E}_{(n\text{x}q)}$ is a matrix of random disturbances (errors). Fitting the model in equation (3.1) to the observed data requires the estimation of the unknown parameter **B**. This can be done through *maximum likelihood* if *normality* is assumed for the matrix of errors **E**. These normality assumptions are taken in a similar sense to those on $\mathrm{E}^{(k)}$ in equation (2.14) in section 2.2.3. In other words, each row of **E** has a *multivariate normal distribution* under the usual distributional assumptions. The errors in any row of **E** are statistically independent of the errors in any other row since the rows of the data matrices **Z** and **Y** generally represent randomly sampled subjects. If no distributional assumptions are made on **E**, the estimate of the matrix of coefficients can be obtained by using the *least squares* method. Fortunately, both the maximum likelihood and least squares approaches lead to the same estimate, so that the actual assumptions made are not critical to the outcome of the analysis. Assuming that both the $\underline{Z}$ and $\underline{Y}$ variables have first been 'mean-centered', the least squares estimate of the matrix of regression coefficients, $\hat{\mathbf{B}}$, is obtained by minimizing trace($\mathrm{E}^{\mathrm{T}}\mathrm{E}$), i.e., the sum of squares of errors. This yields,

$$\hat{\mathrm{B}} = \left(\mathrm{Z}^{\mathrm{T}}\mathrm{Z}\right)^{-1} \mathrm{Z}^{\mathrm{T}}\mathrm{Y} = \mathrm{S}_{\mathrm{ZZ}}^{-1}\,\mathrm{S}_{\mathrm{ZY}}, \tag{3.2}$$

where, $S_{ZZ} = Z^TZ$ is the (pxp) matrix of the sums of squares and cross-products of the predictor variables, and $S_{ZY} = Z^TY$ is the (pxq) matrix of the sums of squares and cross-products of the predictor and the response variables. The columns of $\hat{B}$ are exactly the same as the respective vectors of coefficients which would be obtained if the response variables were predicted separately from *all* the predictor variables.

Corresponding to the sums of squares of the actual and the predicted Y values in ordinary (univariate multiple) regression, are the (pxp) sums of squares and cross-product matrices $S_{YY} = Y^TY$ and

$$S_{\hat{Y}\hat{Y}} = \hat{Y}^T\hat{Y} = \hat{B}^TZ^TZ\hat{B}^T = S_{YZ}\, S_{ZZ}^{-1}\, S_{ZY}, \tag{3.3}$$

since $\hat{Y} = Z\hat{B}$; $\hat{Y}$ being the predictor of Y. The *squared multiple correlation* for predicting each variable in the set $\underline{Y}$ from all the $\underline{Z}$ variables (i.e., as in univariate regression) is the ratio of a diagonal element of $S_{\hat{Y}\hat{Y}}$ to the corresponding element of S_{YY}. These correlations represent the proportion of variance in each $\underline{Y}$ variable, predictable from all the $\underline{Z}$ variables. Hence, the squared multiple correlation measures the degree to which each response variable depends on *all* the predictor variables. Such indices are generally referred to as measures of *association*. However, with the multiple correlation, each $\underline{Y}$ variable is considered in isolation. In other words, to calculate the squared multiple correlation corresponding to each $\underline{Y}$ variable, interdependencies among *all* the $\underline{Y}$ variables are ignored.

A measure of *multivariate association* (MVA) which takes into account the intercorrelations among the variables being predicted (i.e., the $\underline{Y}$ variables in our case) is the *squared canonical* *correlation* between the sets of variables $\underline{Z}$ and $\underline{Y}$. This is the *largest squared multiple correlation* which arises when a linear combination of the $\underline{Y}$ variables is predicted from *all* the $\underline{Z}$ variables. Another useful interpretation of this measure is that it is the *largest squared simple correlation* between a linear combination of the $\underline{Y}$ variables and a linear combination of the $\underline{Z}$ variables. However, since we aim to illuminate the similarities between the *canonical correlation* model and the *multivariate regression* model, we shall focus on the former definition of the *canonical correlation* coefficient.

Define a linear combination of the Y variables as $C_1 = \underline{a}_1^T\underline{Y}$, where $\underline{a}_1$ is a (qx1) vector of coefficients between the *new* variable C_1 and the Y variables. Then choosing the vector $\underline{a}_1$ such that the coefficient,

$$\hat{\rho}_1^2 = \frac{\underline{a}_1^T\, S_{\hat{Y}\hat{Y}}\, \underline{a}_1}{\underline{a}_1^T\, S_{YY}\, \underline{a}_1} = \frac{\underline{a}_1^T\, S_{YZ}\, S_{ZZ}^{-1}\, S_{ZY}\, \underline{a}_1}{\underline{a}_1^T\, S_{YY}\, \underline{a}_1}, \tag{3.4}$$

is maximized, yields the largest canonical correlation $\hat{\rho}_1$. The corresponding vector $\underline{a}_1$ is often referred to as the vector of *canonical weights* and the corresponding variable $C_1 = \underline{a}_1^T\underline{Y}$ is generally termed the *first canonical variate*. As noted before, the canonical correlation $\hat{\rho}_1$ can also be interpreted as the *simple correlation* between the linear combination of the $\underline{Y}$ variables C_1 and a linear combination of the $\underline{Z}$ variables (say, $D_1 = \underline{b}_1^T\underline{Z}$, where $\underline{b}_1$ is a (px1) vector of coefficients between the canonical variate D_1 and the $\underline{Z}$ variables). To obtain a unique value for the canonical correlation $\hat{\rho}_1$, the constraint $\text{Var}(C_1) = 1$ (i.e., the variance of C_1 is set equal to unity), which yields

$$\underline{a}_1^T \, S_{YY} \, \underline{a}_1 = 1, \qquad (3.5)$$

is added. The coefficient, $\hat{\rho}_1^2$ can be interpreted as the proportion of variation in the optimum *linear composite* of the $\underline{Y}$ variables C_1 predictable from *all* the $\underline{Z}$ variables. It is *invariant* in the sense that a linear transformation of the variables in the set $\underline{Z}$ or $\underline{Y}$ does not change the value of the measure. It is also *symmetric* in the sense that it has the same value for $\underline{Y}$ predicted from $\underline{Z}$ as for $\underline{Z}$ predicted from $\underline{Y}$. Hence, it can also be interpreted as the proportion of variation in the optimum linear composite of the $\underline{Z}$ variables D_1 predictable from *all* the $\underline{Y}$ variables. A problem with this measure, however, is that after partialling out the two concomitant canonical variates from the sets $\underline{Z}$ and $\underline{Y}$, the residualized variates may still be correlated. Hence, it becomes necessary to determine the successive canonical correlations. In this case, only l-1 more canonical correlations can be determined, where $l = \min(p,q)$.

The procedure above can be generalized to determine *all* the l consecutive squared canonical correlations $\left(\rho_1^2 \geq \hat{\rho}_2^2 \geq \ldots \geq \hat{\rho}_l^2\right)$, by choosing the corresponding vectors $\underline{a}_t$ ($t = 1, 2, \ldots, l$) such that $\hat{\rho}_t^2$ is maximized under the constraint,

$$\underline{a}_t^T \, S_{YY} \, \underline{a}_t = 1. \qquad (3.6)$$

A further constraint is added to the canonical variates $C_t = \underline{a}_t^T\underline{Y}$ so that $\mathrm{Cov}(C_i, C_j) = 0$, where $i \neq j$ and this yields,

$$\underline{a}_i^T \, S_{YY} \, \underline{a}_j = 0, \text{ for all } i \neq j. \tag{3.7}$$

This constraint ensures that the canonical variates are *orthogonal* to each other. Using equation (3.4), a general equation which allows the t-th canonical correlation (t = 1, 2, ..., l) to be determined can be established. This leads to the *characteristic* (or *eigen*) equation,

$$\left[S_{YY}^{-1} \, S_{\hat{Y}\hat{Y}} - \hat{\rho}_t^2 \, I \right] \underline{a}_t = 0,$$

or

$$\left[S_{YY}^{-1} \, S_{YZ} \, S_{ZZ}^{-1} \, S_{ZY} - \hat{\rho}_t^2 \, I \right] \underline{a}_t = 0, \tag{3.8}$$

using equation (3.3). This implies that, in general, the squared canonical correlations $\hat{\rho}_t^2$ (t = 1, 2, ..., l) between the sets **Z** and **Y** are the *eigenvalues* of the (qxq) matrix $S_{YY}^{-1} \, S_{YZ} \, S_{ZZ}^{-1} \, S_{ZY}$. The vector of canonical weights $\underline{a}_t$ is the corresponding *eigenvector*.

Let $\hat{\mathbf{R}} \left[= (\hat{\rho}_1^2 \;\; \hat{\rho}_2^2 \; \ldots \; \hat{\rho}_l^2) \right]$ be the diagonal matrix of the eigenvalues of $S_{YY}^{-1} \, S_{YZ} \, S_{ZZ}^{-1} \, S_{ZY}$ (i.e., the diagonal matrix of *all l* squared canonical correlations). Further let $\mathbf{A}^T = (\underline{a}_1^T \; \underline{a}_2^T \; \ldots \; \underline{a}_l^T)$ denote the corresponding (qxq) matrix of eigenvectors. Then equation (3.8) can be written as

$$\left[S_{YY}^{-1} \, S_{YZ} \, S_{ZZ}^{-1} \, S_{ZY} - \hat{\mathbf{R}} \right] \mathbf{A} = 0. \tag{3.9}$$

Here, **A** is the matrix of canonical weights under the constraint,

$$A^T S_{yy} A = I. \qquad (3.10)$$

This constraint means that the respective variances of the canonical variates C_t ($t = 1, 2, \ldots, l$) are all set equal to unity and that these canonical variates are orthogonal, as seen before.

From the descriptions above, it can be easily seen that the model for *canonical correlation analysis* is a special case of the *general multivariate linear regression model.* Further details of this relationship can be found in Muller (1982).

3.3 Principal component regression

The principal component regression (PC regression) technique has already been mentioned briefly in the context of variable selection in section 1.3.2. Reference of the details of this technique can be made to Mandel (1982) and Jolliffe (1986).

The key idea in PC regression is to replace the original predictor variables in the regression model by their PCs before estimating the parameters involved. The main advantage of the technique occurs when *multicollinearity* (i.e., linear or non-linear dependencies) is present among the predictor variables. In practice, exact linear dependencies among the variables rarely occur, hence, the term multicollinearity shall be used to express near-linear dependencies. The idea of using PCs rather than the original variables is not new (see Hotelling (1957), Muller (1981)), and it has a number of advantages. Since the PCs are orthogonal to each other, the calculation of the matrix of regression coefficients is much more straightforward than when the original variables have not first been orthogonalized. Furthermore, owing to the orthogonality of the transformed variables (PCs), contributions of each PC to the regression equation can be more easily interpreted than contributions of the original predictor variables. Thus, even when multicollinearity is not a problem, PC regression may have advantages for computation and interpretation over ‘ordinary’ regression. However, it is important to note that, despite this attraction towards PC regression, if the PCs have no clear meaning, interpretation of the regression equation itself may be hindered.

While the PC regression technique can be applied to every regression situation covered by equation (3.1) in section 3.2, it is particularly attractive in the case of multicollinearity. Multicollinearity can occur for a variety of reasons. Following are three examples of conditions under which it can be induced.

(a) In polynomial regression.

(b) In cases where the necessary substantive insight of the system is lacking, it is common practice to include a large number of variables in the hope that no pertinent variable will be overlooked, and some of these variables may be highly inter-related.

(c) In cases where the model is over-defined, i.e., where the sample is too small to permit all the potential coefficients to be estimated.

Multicollinearity has serious effects on *least squares* estimation, inference, variable selection and prediction. In its presence, $\mathbf{Z}^T\mathbf{Z}$ is singular (since rank(Z) < p; implying that determinant($\mathrm{Z}^T\mathrm{Z}$) = 0) and therefore not invertible. Since $(\mathrm{Z}^T\mathrm{Z})^{-1}$ is necessary for finding $\hat{\mathbf{B}}$, the estimator for the regression coefficients (see equation (3.2)), not only estimating $\hat{\mathrm{B}}$ is a problem, but also the precision of any predictions made from the regression equation is poor. PC regression can be used as an attempt to correct this deficiency because the PCs are orthogonal to each other, implying that the matrix of PC scores is of full rank. An alternative approach used to treat multicollinearity is to eliminate those variables which make no contribution to the system either in a practical sense or statistically. However, in our proposed bootstrap technique for choosing the number of components, we tackle multicollinearity through PC regression. Following is a description of the PC regression technique.

By introducing the SVD of Z (see equation (1.4)) into equation (3.1), we obtain

$$Y = \tilde{U}\,\tilde{S}\,\tilde{V}^{T}B + E, \tag{3.11}$$

where $\tilde{U}^{T}\tilde{U} = \tilde{V}\tilde{V}^{T} = \tilde{V}^{T}\tilde{V} = I_{p}$ and $\tilde{S}$ is diagonal with diagonal elements $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_p$. Here, $\tilde{s}_1 \geq \tilde{s}_2 \geq \ldots \geq \tilde{s}_p$ are the non-negative square-roots of the eigenvalues of $Z^{T}Z$ or ZZ^{T}. The columns of $\tilde{U}$ are the p orthonormalized eigenvectors of ZZ^{T} and the rows of $\tilde{V}^{T}$ are the orthonormalized eigenvectors of $Z^{T}Z$. Written in this form, the model is referred to as the *principal component regression model*. Equation (3.7) can be re-written as

$$Y = \tilde{U}\left[\tilde{S}\tilde{V}^{T}B\right] + E, \tag{3.12}$$

where $\tilde{S}\tilde{V}^{T}B$ is an $(r\text{x}p)$ matrix; r being the rank of Z. If

$$H = \tilde{S}\tilde{V}^{T}B, \tag{3.13}$$

then, the model becomes

$$\mathbf{Y} = \tilde{\mathbf{U}}\mathbf{H} + \mathbf{E}. \tag{3.14}$$

Since matrices **Y** and $\tilde{\mathbf{U}}$ are known, the least squares solution for the unknown elements of **H** are obtained by the *usual* matrix equation

$$\hat{\mathbf{H}} = \left[\tilde{\mathbf{U}}^T\tilde{\mathbf{U}}\right]^{-1} \tilde{\mathbf{U}}^T\mathbf{Y}, \tag{3.15}$$

which, as a result of the constraint $\tilde{\mathbf{U}}^T\tilde{\mathbf{U}} = \mathbf{I}_r$, becomes

$$\hat{\mathbf{H}} = \tilde{\mathbf{U}}^T\mathbf{Y}. \tag{3.16}$$

Hence,

$$\hat{\mathbf{Y}} = \tilde{\mathbf{U}}\tilde{\mathbf{U}}^T\mathbf{Y}. \tag{3.17}$$

If knowledge of the estimator for the matrix of regression coefficients, say $\hat{\mathbf{B}}$, is desired, equations (3.13) and (3.16) can be used to obtain

$$\hat{\mathbf{B}} = \tilde{\mathbf{V}}\tilde{\mathbf{S}}^{-1}\,\hat{\mathbf{H}} = \tilde{\mathbf{V}}\tilde{\mathbf{S}}^{-1}\tilde{\mathbf{U}}^T\mathbf{Y}. \tag{3.18}$$

3.4 The general bootstrap

Suppose that the data points $\underset{\sim}{x}_i$ (i = 1, 2, ..., n), i.e., the rows of the data matrix $\mathbf{X}_{(n\times p)}$, are independent observations from some multivariate distribution F in a p-dimensional space, and $\theta(F)$ is some real-valued parameter of interest. Then

(a) F is unknown, but can be estimated by the *empirical probability distribution* $\hat{F}$, for example,

$\hat{F}$: mass $1/n$ on each observation $\underline{x}_1$ $(i=1,2,\ldots,n)$,

(b) the bootstrap estimate of $\theta(F)$ is given by

$$\hat{\theta}_{\mathrm{BOOT}} = \theta(\hat{F}). \tag{3.19}$$

The basic philosophy underlying the bootstrap procedure for approximating $\hat{\theta}_{\mathrm{BOOT}}$ as outlined by Efron (1979, 1982), Efron and Gong (1983), etc., is to

Step 1: Construct the empirical probability distribution $\hat{F}$ as just described.

Step 2: Draw a bootstrap sample $\underline{x}_1^* \; \underline{x}_2^* \ldots \underline{x}_n^*$ by independent random sampling from $\hat{F}$, i.e., make n random draws with *replacement* from the rows of $\mathrm{X}_{(nxp)}$. If we let $\mathrm{X} = (\underline{x}_1 \; \underline{x}_2 \ldots \underline{x}_n)$ and $\mathrm{X}^* = (\underline{x}_1^* \; \underline{x}_2^* \ldots \underline{x}_n^*)$, then $\underline{x}_j^*$ may be equal to some $\underline{x}_i^*$ for $i,j = 1, 2, \ldots, n$.

Step 3: Compute the bootstrap replication $\hat{\theta}^*$, i.e., estimate $\theta(\hat{F})$ from X^* by $\hat{\theta}^*(\underline{x}_1^* \; \underline{x}_2^* \ldots \underline{x}_n^*)$.
Note that $\hat{\theta}^*$ is formed from X^* in precisely the same manner as $\hat{\theta}$ is formed from X.

Step 4: Repeat steps 2 and 3 a sufficiently large number (say b) of times, to obtain independent bootstrap replications $\hat{\theta}^{*1}$, $\hat{\theta}^{*2}$, ..., $\hat{\theta}^{*b}$ and compute the required bootstrap estimate of $\theta(\hat{F})$ as

$$\hat{\theta}_{\mathrm{BOOT}} = \left(1/b \right) \sum_{m=1}^{b} \hat{\theta}^{*m} . \tag{3.20}$$

If required, compute the variance due to bootstrapping

$$s^{*2}_{\mathrm{BOOT}} = \left(1 \Big/ b-1\right) \sum_{m=1}^{b} \left[\hat{\theta}^{*m} - \hat{\theta}_{\mathrm{BOOT}}\right]^2 . \qquad (3.21)$$

A large number of publications have been made on the bootstrap technique and its applications and these include Efron (1979, 1981, 1982, 1985, 1990), Efron and Gong (1983), Efron and Tibshirani (1986), McLachlan (1987), Gleason (1988) and Fisher and Hall (1989, 1990). Applications of the bootstrap technique to *Regression Analysis* can be found in Freedman (1981), Bunke and Droge (1984), Freedman and Peters (1984), Rice (1984), Stine (1985), De Wet and Van Wyk (1986), Gong (1986), Wu (1986), Bose (1988), Hardle and Bowman (1988), Thomas *et al.* (1989), Mammen (1989), Navidi (1989), Rocke (1989), Hall and Hart (1990), and Hardle and Marron (1990). Bootstrapping *correlation coefficients* have been discussed by Efron (1979, 1982), Dolker *et al.* (1982), Lunneborg (1985), Rasmussen (1987), Strube (1988), Young (1988), and Hall *et al.* (1989).

Several attempts have been made to improve the 'ordinary' bootstrap procedure described earlier. Such modifications include the *smoothed* bootstrap and the *balanced* bootstrap. Of these, the balanced bootstrap is the most popular choice. In the ordinary bootstrap, each datum $\underline{x}_i$ (i = 1, 2, ..., n) may not appear equally often in the aggregate of all b bootstrap samples. The balanced bootstrap procedure attempts to correct this deficiency. This simple balance also has the concomitant effect of reducing the probable error in the variance s^{*2}_{BOOT} due to bootstrapping. Further discussions of the balanced bootstrap can be found in Gleason (1988), Graham *et al.* (1990), and Hall (1990a, 1990b).

However, at this point, the virtues of the balanced bootstrap will be spared for future investigations. In the present research project, only the *ordinary* bootstrap procedure is explored as a means of introducing the bootstrap technique for choosing the number of components in PCA.

3.5 The bootstrap choice of the number of components

To estimate the number of principal components to retain in order to describe the data adequately, we need to define the appropriate statistic on which the bootstrap ideas in section 3.4 above can be applied. Hence, we begin by describing such a statistic. In order to do this, it is necessary to describe how the bootstrap choice of the number of components utilizes the ideas of *Multivariate Regression Analysis* (see section 3.2).

Consider our data matrix $\mathbf{X}_{(n \times p)}$ of p variables measured on each of n individuals or objects. Needless to say, performing PCA on this data matrix yields p components. As noted in Chapter 2, the aim is to choose the smallest number (say, k) so that the *first* k of these PCs can be used to 'reproduce' the data with sufficient accuracy. If this is the case, then it can be argued that the first k PCs can be used for adequate representation of the data.

Next consider the *Singular Value Decomposition* (SVD) of the data matrix X which is given by equation (1.4). Use $\mathbf{X}^{(k)}_{(n \times p)}$ to denote the estimate of the data matrix which arises from using *only* the *first* k columns of $\mathbf{U}$, *only* the *first* k columns of the *first* k rows of $\mathbf{S}$ and *only* the *first* k rows of $\mathbf{V}^{\mathrm{T}}$. This is seen as equivalent to fitting *only* the *first* k PCs. Hence, it can be argued that, if the original data matrix X can be sufficiently predicted from $\mathbf{X}^{(k)}$ in the *multivariate regression* sense, then the first k PCs can be used for adequate representation of the data. Therefore, the problem is reduced to the choice of the smallest value of k for which the 'model fit' of the *multivariate linear regression* equation

$$X = X^{(k)}\ B^{(k)} + E^{(k)} \qquad (3.22)$$

is adequate, where k = 1, 2, ..., etc. Here, $B_{(pxp)}$ is the matrix of regression coefficients determined in such a way that the random disturbances in $E^{(k)}_{(nxp)}$ are minimized. Notice that, equation (3.22) above is equivalent to equation (2.14) in section 2.2.3 used to illustrate the cross-validatory technique of Eastment and Krzanowski (1982). The difference is that here, the predictor $\hat{X}^{(k)}$ of the data matrix X is given by $X^{(k)}B^{(k)}$, while in Eastment and Krzanowski's technique, this predictor is obtained by using the *cross-validation* procedure. However, the idea in both cases is to use the Prediction Sums of Squares, i.e., PRESS(k), k = 1, 2, ..., etc. (see equation (2.15)), to determine the *optimum* value of k. The matrix of errors $E^{(k)}$ is the measure of the discrepancy between the original data matrix X and its predictor $\hat{X}^{(k)}$. Hence, PRESS(k) is, in a sense, the measure of *multivariate association* (MVA) between the data matrix X and the estimate $X^{(k)}$. Therefore, bootstrapping in our context is used to find the best estimator for the *true* error sums of squares due to *not fitting* the *last* p-k PCs (k = 1, 2, ... etc.).

Bootstrapping is a methodology whose utilization depends intrinsically on the use of high-speed computing power, used for assessing the variability in an estimate (PRESS(k) in our context) using only the data at hand. Bootstrap samples are obtained by *resampling* the original observations in such a way that the data structure is preserved, as described in section 3.4. These samples are then used to assess the estimate of interest PRESS(k). No distributional assumptions are necessary, and the parameter estimates obtained by bootstrapping are more robust than those obtained from the original sample.

Next, we describe how bootstrapping is applied to the *multivariate linear regression* model in equation (3.22) in order to choose the number of components in PCA. Several authors, in particular, Freedman (1981), Efron (1982), Freedman and Peters (1984) and Stine (1985), have suggested that, when bootstrapping is used for estimating regression coefficients, the centered residuals (errors) of the fitted model should be resampled instead of the data matrix itself. However, these authors also pointed out that this approach is appropriate *only* when $X^{(k)}$ in equation (3.22) is not random and the errors $\underline{e}_i^{(k)}$ ($i = 1, 2, \ldots, n$) in the matrix $E^{(k)}$ are *homoscedastic*. In our method, $X^{(k)}$ is an estimate of the data matrix **X** obtained from using only the first k PCs. As a result, since **X** is random $X^{(k)}$ is also random and there is; in general, some dependence between each row of the matrix of errors $E^{(k)}$ and the corresponding row of $X^{(k)}$. Hence the errors are *heteroscedastic*. In such cases, it is inappropriate to resample the errors as this would obliterate the heteroscedasticity. Thus, in our method the '*correlation* sense' of the model in equation (3.22) is assumed and the rows of the data matrix $X = (\underline{x}_1 \; \underline{x}_2 \ldots \underline{x}_n)$ are resampled rather than the errors. Bootstrap samples $X^{*m} = (\underline{x}_1^{*m} \; \underline{x}_2^{*m} \; \ldots \; \underline{x}_n^{*m})$ ($m = 1, 2, \ldots, b$) are obtained by applying the 'ordinary bootstrap scheme' described in section 3.4 and the corresponding estimates $X^{*m(k)} = \left\{\underline{x}_1^{*m(k)} \; \underline{x}_2^{*m(k)} \; \ldots \; \underline{x}_n^{*m(k)}\right\}$, and hence $\text{PRESS}^{*m}(k)$ ($k = 0, 1, 2, \ldots, p-2$) are computed. Figure 3.1 gives a general outline of the application of the *bootstrapping* ideas in section 3.4 and the *Multivariate Linear Regression* (MVLR) model in equation (3.22) to our bootstrap technique for choosing the number of components.

Figure 3.1 *Outline of the prediction of the m-th bootstrap sample* X^{*m} *using only the first k PCs in the bootstrap technique for choosing the number of components in PCA.*

$$\begin{tikzcd}
X \arrow[r, "\textit{Resample}", "\textit{BOOTSTRAP}"'] \arrow[dr] & X^{*m} \arrow[r, "\textit{Estimate}", "\textit{PCA}"'] & X^{*m(k)} \arrow[r, "\textit{Predict}", "\textit{MVLR}"'] & \hat{X}^{*m(k)} \arrow[dll] \\
& \mathrm{PRESS}^{*m}(k) & &
\end{tikzcd}$$

Consider our data matrix $X_{(n\times p)}$ of p variables measured on each of n individuals or objects. Further let $X^{*m}_{(n\times p)}$ be the m-th bootstrap sample (m = 1, 2, ..., b) obtained by resampling the rows of the data matrix X according to the 'ordinary bootstrap scheme' in section 3.4. Then by performing a PCA on the m-th bootstrap sample we can re-write the model in equation (3.22) as

$$X^{*m} = \hat{X}^{*m(k)} + E^{*m(k)}, \tag{3.23}$$

Here, $\hat{X}^{*m(k)}$ is the predictor of X^{*m} based on *only* the first k PCs (m = 1, 2, ..., b and k = 0, 1, 2, ..., p-2) and $E^{*m(k)}$ is the corresponding matrix of errors. Notice that, when k = 0, it is assumed that *all* of the variation in X^{*m} is due to chance, and hence the model (3.23) becomes $X^{*m} = E^{*m(k)}$.

Next, we describe how the predictor $\hat{X}^{*m(k)}$ is computed from the first k PCs of the corresponding bootstrap sample X^{*m}. The SVD of X^{*m} which takes the form

$$X^{*m} = U^{*m} S^{*m} \left[V^{*m}\right]^T, \tag{3.24}$$

enables us to write each (i,j)-th element (i = 1, 2, ..., n and j = 1, 2, ..., p) of X^{*m} using the model (2.7), which retains only the first k PCs. Here the constraints on U^{*m}, S^{*m}, and V^{*m} are taken in a similar sense as those on the matrices **U**, **S** and **V** in equation (1.4). In other words, $\left[U^{*m}\right]^T U^{*m} = \left[V^{*m}\right]^T V^{*m} = V^{*m}\left[V^{*m}\right]^T = I_p$ and S^{*m} is diagonal with diagonal elements, $s_1^{*m} \ge s_2^{*m} \ge \dots \ge s_p^{*m}$. Written out in detail, equation (3.24) becomes

$$X^{*m} = \begin{bmatrix} u_{11}^{*m} & u_{12}^{*m} & \dots & u_{1p}^{*m} \\ u_{21}^{*m} & u_{22}^{*m} & \dots & u_{2p}^{*m} \\ \vdots & & & \\ u_{n2}^{*m} & u_{n2}^{*m} & \dots & u_{np}^{*m} \end{bmatrix} \begin{bmatrix} s_1^{*m} & & & 0 \\ & s_2^{*m} & & \\ & & \ddots & \\ 0 & & & s_p^{*m} \end{bmatrix} \begin{bmatrix} v_{11}^{*m} & v_{21}^{*m} & \dots & v_{p1}^{*m} \\ v_{12}^{*m} & v_{22}^{*m} & \dots & v_{p2}^{*m} \\ \vdots & & & \\ v_{1p}^{*m} & v_{2p}^{*m} & \dots & v_{pp}^{*m} \end{bmatrix}. \tag{3.25}$$

Using only the *first* k columns of U^{*m}, only the *first* k columns of the *first* k rows of S^{*m}, and only the first k rows of $\left[V^{*m}\right]^T$ to estimate the m-th bootstrap sample X^{*m} in equation (3.25), is equivalent to retaining only the *first* k PCs. In other words, if $X^{*m(k)}$ is the estimated value of X^{*m} using the *first* k components, then

$$X^{*m(k)} = \begin{array}{c} \hfill k \\ \left[\begin{array}{ccc:} u_{11}^{*m} & u_{12}^{*m} & \dots \\ u_{21}^{*m} & u_{22}^{*m} & \dots \\ \vdots & & \\ u_{n2}^{*m} & u_{n2}^{*m} & \dots \end{array}\right. \end{array} \quad \begin{array}{c} \hfill k \\ \left[\begin{array}{cc:} s_1^{*m} & 0 \\ 0 & s_2^{*m} \; \ddots \\ \hdashline \end{array}\right. \end{array}_{k} . \quad {}_{k}\left[\begin{array}{cccc} v_{11}^{*m} & v_{21}^{*m} & \dots & v_{p1}^{*m} \\ v_{12}^{*m} & v_{22}^{*m} & \dots & v_{p2}^{*m} \\ \hdashline \vdots & & & \end{array}\right], \tag{3.26}$$

which can be written in a more compact form as,

$$X^{*m(k)} = U^{*m(k)}\, S^{*m(k)} \left[V^{*m(k)}\right]^T. \tag{3.27}$$

Note here that, $\left[U^{*m(k)}\right]^T U^{*m(k)} = \left[V^{*m(k)}\right]^T V^{*m(k)} = V^{*m(k)}\left[V^{*m(k)}\right]^T$

$= I_k$. In order to determine the predictor $\hat{X}^{*m(k)}$ of X^{*m}, as predicted from $X^{*m(k)}$, our approach utilizes the *multivariate linear regression* model,

$$X^{*m} = X^{*m(k)} B^{*m(k)} + E^{*m(k)}, \tag{3.28}$$

which corresponds to equations (3.1) and (3.22). Here, $B^{*m(k)}$ is the matrix of regression coefficients determined in such a way that the random disturbances in $E^{*m(k)}$ are minimized. Note that no distributional assumptions need to be made on $E^{*m(k)}$. There are two reasons for this, firstly, $B^{*m(k)}$ is estimated using the *least squares* method, and secondly, no distributional assumptions are necessary for the application of the *bootstrap methodology*. Let $\hat{B}^{*m(k)}$ be the estimator for $B^{*m(k)}$, then

$$\hat{X}^{*m(k)} = X^{*m(k)} \hat{B}^{*m(k)}, \tag{3.29}$$

which illuminates the correspondence between equations (3.28) and (3.23). The least squares estimate of the matrix of regression coefficients, $\hat{B}^{*m(k)}$, may be found in the usual way, i.e., by minimizing

$$\text{Trace} \left\{ \left[E^{*m(k)} \right]^T \left[E^{*m(k)} \right] \right\},$$

the prediction sum of squares.

Note here that, despite the fact that the columns of the matrices $\left[U^{*m(k)} S^{*m(k)} \right]$ and $V^{*m(k)}$ from which $X^{*m(k)}$ is obtained are orthogonal (see equation (3.27)), the columns (variables) of $X^{*m(k)}$ are not necessarily orthogonal. This can be explained as follows: In a similar sense as **US** in equation (1.4) gives the PC scores of the data matrix **X**, multiplying $\mathbf{U}^{*m(k)}$ by $\mathbf{S}^{*m(k)}$ yields the PC scores corresponding to the m-th bootstrap sample. This is equivalent to projecting the original sample points onto the space defined by the orthogonal PC axes. However, multiplying these PC scores by $\left[\mathbf{V}^{*m}\right]^{T}$ is equivalent to projecting the points back onto the original set of axes (but retaining information provided by only the first k PCs) which, in practice, are not necessarily orthogonal. Following this argument, it is seen that the rank of $\mathbf{X}^{*m(k)}$ (say, r) can take values less than p (full rank) indicating the presence of *multicollinearity*. Hence, computation of the estimator $\hat{\mathbf{B}}^{*m(k)}$ of the matrix of regression coefficients, may not be possible through the usual formulation as

$$\hat{\mathbf{B}}^{*m(k)} = \left\{\left[\mathbf{X}^{*m(k)}\right]^{T} \left[\mathbf{X}^{*m(k)}\right]\right\}^{-1} \left[\mathbf{X}^{*m(k)}\right]^{T} \mathbf{X}^{*m}, \qquad (3.30)$$

which corresponds to equation (3.2) in section 3.2. To evade this difficulty, the *principal component regression* technique described in section 3.3 is utilized and following are the details of its application.

Let

$$\mathbf{X}^{*m(k)} = \tilde{\mathbf{U}}^{*m(k)} \; \tilde{\mathbf{S}}^{*m(k)} \; \left[\tilde{\mathbf{V}}^{*m(k)}\right]^{T} \qquad (3.31)$$

be the SVD of $\mathbf{X}^{*m(k)}$, where $\tilde{\mathbf{U}}^{*m(k)}$, $\tilde{\mathbf{S}}^{*m(k)}$, and $\tilde{\mathbf{V}}^{*m(k)}$ are taken in a similar sense to the matrices **U**, **S**, and **V** in equation (1.4). In other words, $\left[\tilde{\mathbf{U}}^{*m(k)}\right]^{T}\tilde{\mathbf{U}}^{*m(k)} = \left[\tilde{\mathbf{V}}^{*m(k)}\right]^{T}\tilde{\mathbf{V}}^{*m(k)} = \tilde{\mathbf{V}}^{*m(k)}\left[\tilde{\mathbf{V}}^{*m(k)}\right]^{T} = \mathbf{I}_{r}$ and $\tilde{\mathbf{S}}^{*m(k)}$ is diagonal with diagonal elements, $\tilde{s}_{1}^{*m(k)} \geq \tilde{s}_{2}^{*m(k)} \geq \ldots \geq \tilde{s}_{r}^{*m(k)}$. Then the equation (3.28) becomes

$$\mathrm{X}^{*m} \quad = \tilde{\mathrm{U}}^{*m(k)} \left\{ \tilde{\mathrm{S}}^{*m(k)} \left[\tilde{\mathrm{V}}^{*m(k)} \right]^{T} \mathrm{B}^{*m(k)} \right\} + \mathrm{E}^{*m(k)}. \tag{3.32}$$

Note that equation (3.32) corresponds to equation (3.12) in section 3.3. Now, following the steps provided by equations (3.13)-(3.17); $\hat{\mathrm{X}}^{*m(k)}$, the predictor of the original m-th bootstrap sample, X^{*m}, can be written as

$$\hat{\mathrm{X}}^{*m(k)} = \tilde{\mathrm{U}}^{*m(k)} \left[\tilde{\mathrm{U}}^{*m(k)} \right]^{T} \mathrm{X}^{*m}, \tag{3.33}$$

and this corresponds to equation (3.17). Finally, using equations (3.23) and (3.33), we obtain

$$\mathrm{E}^{*m(k)} = \mathrm{X}^{*m} - \left\{ \tilde{\mathrm{U}}^{*m(k)} \left[\tilde{\mathrm{U}}^{*m(k)} \right]^{T} \mathrm{X}^{*m} \right\}, \tag{3.34}$$

from which for a fixed k (k = 0, 1, 2, ..., p-2),

$$\mathrm{PRESS}^{*m}(k) = \mathrm{Trace} \left\{ \left[\mathrm{E}^{*m(k)} \right]^{T} \left[\mathrm{E}^{*m(k)} \right] \right\}, \tag{3.35}$$

can be computed, m = 1, 2, ..., b. Note that, using the theme behind the *multivariate linear regression model* in section 3.2, $\mathrm{PRESS}^{*m}(k)$ can be regarded as the measure of departure between the m-th bootstrap sample and the corresponding sample structure that arises from fitting only the first k PCs, after partialling out the respective multiple correlations between the two structures. Furthermore, *multivariate linear regression analysis* and *canonical correlation analysis* share the same general model, as seen in section 3.2. Thus, $\mathrm{PRESS}^{*m}(k)$ can be classified as a *measure of* *multivariate association.*

For a fixed value of k, if we let $\overline{\text{PRESS}}^{*}(k)$ be the *arithmetic mean* of these PRESS values computed over all b bootstrap samples, i.e.,

$$\overline{\text{PRESS}}^{*}(k) = \left[1\Big/b\right]\sum_{m=1}^{b}\text{PRESS}^{*m}(k) \qquad (3.36)$$

which corresponds to equation (3.20) in section 3.4, then a suitable function of $\overline{\text{PRESS}}^{*}(k)$ ($k = 1, 2, \ldots, p-2$) can be used to determine optimal k. The analogy of the present technique with *polynomial regression* (see Draper and Smith (1966) and Wold (1976)) suggests

$$\tilde{W} = \frac{\overline{\text{PRESS}}^{*}(k-1) - \overline{\text{PRESS}}^{*}(k)}{D_1} \div \frac{\overline{\text{PRESS}}^{*}(k)}{D_2} \qquad (3.37)$$

which is analogous to the F ratio statistic used for model choice in variable selection with linear regression as a suitable criterion for determining *optimal* k. Here D_1 and D_2 are taken exactly as in equation (2.11): $D_1 = n$ and $D_2 = n(p-k-1)$ are the degrees of freedom associated with $\left[\overline{\text{PRESS}}^{*}(k-1) - \overline{\text{PRESS}}^{*}(k)\right]$ and $\overline{\text{PRESS}}^{*}(k)$, respectively. In other words D_1 is the number of degrees of freedom required to fit the k-th PC while D_2 is the number of degrees of freedom remaining after fitting the k-th PC. Note also that $\tilde{W}$ has the same interpretation as Eastment-Krzanowski's (1982) W statistic (see equation (2.21)). In other words, $\tilde{W}$ is the increase in predictive information supplied by the k-th PC divided by the the average information in the remaining $p-k$ principal components. We therefore choose **optimal** k to be the *largest* value of k for which $\tilde{W}$ is greater than unity.

CHAPTER 4
MONTE CARLO COMPARISON OF THE BOOTSTRAP AND THE CROSS-VALIDATORY CHOICE OF THE NUMBER OF COMPONENTS

4.1 Introduction

In this Chapter, the performance of the proposed bootstrap technique for choosing the number of components is compared and contrasted with the cross-validatory scheme of Eastment and Krzanowski (1982) by means of Monte Carlo simulation experiments. The optimality criteria for these procedures are $\tilde{W}$ and W respectively, hence, these two methods will be hereafter referred to as the $\tilde{W}$ technique and the W technique, respectively. Computer programs corresponding to each of these techniques written in the matrix programming software GAUSS (1988) are used to conduct the simulation study and these programs are presented in Appendix B of the thesis. The advantage of using simulated data for these evaluations is that, unlike with real data where there is no *prior* knowledge of the nature of the data, the artificial data can be constructed in such a way that the number of PCs necessary for a sufficient description of the data is *known prior* to the analysis.

4.2 Simulation study plan

The design for the simulation study we adopt in this investigation utilizes models 1 through 4 of Jolliffe (1972), who constructed these models in such a way that within each model, some variables are linear combinations of others except for random disturbances and they are therefore redundant. Jolliffe's models make use of randomly generated variates Z_l ($l = 1, 2, \dots, v$) which are identically and independently distributed (*iid*) as $N(0, 1)$.

Table 4.1: *Definition of constructed variables for models 1-4 of Jolliffe (1972)*

Variable	Relationship with independent standard normal variates			
	Model 1	Model 2	Model 3	Model 4
X_1	Z_1	Z_1	Z_1	Z_1
X_2	Z_2	Z_2	Z_2	Z_2
X_3	Z_3	Z_3	Z_3	Z_2+Z_3
X_4	$Z_1+0.5Z_4$	$Z_1+0.5Z_4$	$Z_1+0.8Z_2+0.6Z_4$	Z_4
X_5	$Z_2+0.7Z_5$	$Z_2+0.7Z_6$	$Z_2+0.7Z_5$	$Z_4+0.75Z_5$
X_6	Z_3+Z_6	Z_2+Z_6	$Z_3+0.5Z_6$	$2Z_4+0.75Z_5+1.5Z_6$
X_7	——	——	——	Z_7
X_8	——	——	——	$Z_7+0.5Z_8$
X_9	——	——	——	$2Z_7+0.5Z_8+Z_9$
X_{10}	——	——	——	$3Z_7+Z_8+Z_9+Z_{10}$

Being linear combinations of these *iid* variates, the constructed variables X_j ($j = 1, 2, \ldots, p$) in each model are distributed as $N(0, \sigma_j^2)$, where σ_j^2 is the variance of X_j. The relationship between the constructed variables X_j ($j = 1, 2, \ldots, p$) and the *iid* variates Z_l ($l = 1, 2, \ldots, v$) is shown in Table 4.1. It can be seen from this table that the models were constructed so that the new variables X_j ($j = 1, 2, \ldots, p$) fall into groups and within which the variables are linear combinations of each other (plus random disturbances), while variables from different groups are independent. For example, in model 2 variables X_1 and X_4 fall into one group, describing one dimension; variables X_2, X_5, and X_6 fall into another group, describing a second dimension; and variable X_3 on its own describes a third dimension. Hence, the effective dimensionality of this model is 3. Using this 'clustering' procedure, the effective dimensionality of a data set generated in accordance with each of the other models can be found. In each model, the total number of variables is fixed and so is the number of groups or dimensions. For models 1 through 3, there are 6 constructed variables (i.e., p = 6) falling into 3 groups (i.e, k = 3), and for model 4 there are 10 variables (i.e., p = 10) in 4 groups (i.e., k = 4). This way, for each model, the number of components needed to represent the data adequately is *known prior* to the attempt to determine it using the cross-validatory (W) or the bootstrap ($\tilde{W}$) 'dimensionality detection' techniques. The performance of these techniques can be assessed using the percentage of the total number of simulations, each technique chooses the 'correct' number of components (as indicated by the number of groups in each model) to represent the 'signal' in the data.

The data sets for the Monte Carlo simulation study are therefore generated in accordance with models 1 through 4 in Table 4.1 using a computer program written in GAUSS (1988). Sample sizes are fixed at two levels: small (n = 25) and large (n = 50) for each choice of the number of variables in the data. The reason for this is not only to compare the two techniques W and $\tilde{W}$, but also to examine any significant influence on their behaviour due to the sizes of samples. Further, in the case of the $\tilde{W}$ technique, the number of bootstrap samples is fixed at two levels, b = 50 and b = 100. This is done to determine whether or not an increase in the number of bootstrap samples would significantly improve the performance of the proposed bootstrap technique. The number of Monte Carlo training samples for which each technique (W or $\tilde{W}$) is applied to detect the dimensionality of the data under each of the sampling considerations described above (i.e., model, sample size and number of bootstrap samples in the case of $\tilde{W}$) is fixed at 50.

4.3 Monte carlo results

The simulation results of choosing k using the techniques W and $\tilde{W}$ for Jolliffe's models 1 - 4 are summarized in Tables 4.2 - 4.9. For each model, the percentage of the total number of times each technique retains the correct (expected) number of components to sufficiently model the 'signal' in the data under each sampling consideration is shown in these tables. Tables 4.2 - 4.5 give the results for which PCA is performed using covariance matrices (i.e., when $\hat{X}^{(k)}$ in equation (2.14) and $\hat{X}^{*m(k)}$ in equation (3.23) are obtained from the PCA of variables which have not first been standardized), while the results in Tables 4.6 - 4.9 are based on PCA using correlation matrices. Also given in Tables 4.2 and 4.5 for models 1 and 4, respectively, is the computational real (not CPU) time (in seconds) taken by each method to complete the analysis from a single simulation on a 386SX 20Mhz personal computer. This computational time does not appear in Tables 4.3 and 4.4 because models 2 and 3 being similar to model 1 (in terms p = 6 and k = 3) would use almost exactly the same time as model 1 under the same sampling considerations. Furthermore, the only additional computing time needed in the analysis to obtain the results in Tables 4.6 - 4.9 is that which corresponds to the standardization of the variables and this extra time is negligible. Therefore, it seems reasonable to expect that the computational time taken by the methods when PCA is performed using the correlation matrices would be nearly the same as the corresponding time when PCA is performed using the covariance matrices. Hence, the information on the computational time has also been omitted for Tables 4.6 - 4.9.

In the case of our proposed bootstrap technique, the percentage of the total number of times $\tilde{W}$ chooses the right number of components, can also be used as a measure of the performance of $\tilde{W}$ relative to the number of bootstrap samples used. However, it was decided to use an additional measure due to the bootstrapping itself, called the *bootstrap error*, and for each sampling consideration, this is taken to be the *arithmetic mean* of

$$\left[1 \Big/ p-2\right] \sum_{k=0}^{p-2} s_{\text{BOOT}}^{*2}(k),$$

computed over all the Monte Carlo training samples considered. Here, for a fixed value of k, $s_{\text{BOOT}}^{*2}(k)$ is the variance of the $\text{PRESS}^{*m}(k)$ values computed over all b bootstrap samples. In other words,

$$s_{\text{BOOT}}^{*2}(k) = \left[1 \Big/ b-1\right] \sum_{m=1}^{b} \left(\text{PRESS}^{*m}(k) - \overline{\text{PRESS}}^{*}(k)\right)^2, \qquad \textbf{(4.1)}$$

where, $\overline{\text{PRESS}}^{*}(k)$ is the arithmetic mean of the $\text{PRESS}^{*m}(k)$ values (m = 1, 2, ..., b and k = 0, 1, 2, ..., p-2) computed given by equation (3.36). Notice that equation (4.1) corresponds to equation (3.21) in section 3.4.

First, consider the results based on *covariance* matrices. As we noted in section 4.1, the correct (expected) number of components necessary to model the 'signal' in the data for models 1 - 3 is 3 (i.e., k = 3). In model 1 (Table 4.2), for large samples (n = 50), W chooses the correct number of components 96% of the time, while the choice is of 2 components (2%) and zero components (2%) for the remainder of the simulations. Note that the choice of zero

Table 4.2 *Cross-validatory choice* (W) *and the bootstrap choice* ($\tilde{W}$) *of the number of components in model 1 when covariance matrices are used to perform PCA.*

Method	Number of Components (k = 3)	Number of choices (%)			
		n = 50		n = 25	
W	0	2		18	
	1	0		0	
	2	2		10	
	3	96		72	
	4	0		0	
	Computational time (sec.)	117.93		53.39	
$\tilde{W}$		b = 50	b = 100	b = 50	b = 100
	0	0	0	0	0
	1	0	0	0	0
	2	0	0	2	2
	3	98	100	98	98
	4	2	0	0	0
	Computational time (sec.)	89.58	165.27	61.46	108.37
	Bootstrap Error	0.0782	0.0780	0.0604	0.0602

components arises when none of the components yields a value of W greater than the cut-off point 0.9. The choice of zero components is undesirable because it would not make sense to expect zero dimensions (components) to describe the data adequately (at least in our simulation study). For the same sample size (n = 50), $\tilde{W}$ performs slightly better than W; choosing the correct number of components 98% of the time when b, the number of bootstrap samples is 50 and *all* the time when b = 100. When sample size is decreased to n = 25, the performance of $\tilde{W}$ remains virtually unaltered; choosing the correct number of components with 98% probability both when b = 50 and when b = 100. On the other hand, the performance of W deteriorates markedly; the choice being of the correct number of components 72%, 2 components 10% and zero components 18% of the time. Thus, it would seem that neither the size of the sample nor the number of bootstrap samples has a 'significant' effect on the performance of $\tilde{W}$. However, W's performance seems to be heavily dependent on the sample size. Hence, it seems that $\tilde{W}$ consistently retains the correct number of components more than W does, and this behaviour is more prominent when samples are of small size. Furthermore, whenever $\tilde{W}$ fails to choose the correct number of components, it chooses smaller values for k when samples are small and larger values for large samples. On the other hand, W seems to choose smaller values for k for both small and large sample sizes.

For large samples in model 2 (Table 4.3) W performs poorly; retaining 2 components more frequently (82%), 1 component 8% of the time and choosing the correct number of components (k = 3) only 10% of the time. On the other hand, even though $\tilde{W}$'s performance is not as good as it is in model 1, it is much better than that of W. For both levels of the number of bootstrap samples (i.e., for both b = 50 and b = 100) $\tilde{W}$ chooses the correct number of components 86% of the time. Hence, it seems that for samples of large sizes, $\tilde{W}$ is the most successful of the two candidates in choosing the correct number of components in PCA. For the remaining 14% of the time, $\tilde{W}$ chooses one extra component than the expected number. The small samples case seems to improve the performance of W slightly. The number of times it chooses the correct number of components rises by 30%. However, $\tilde{W}$'s performance becomes slightly better for small samples than it is for large samples, only when a large number of bootstrap samples (b = 100) is used, where $\tilde{W}$ chooses the correct number of components

Table 4.3 *Cross-validatory choice (W) and the bootstrap choice ($\tilde{W}$) of the number of components in model 2 when covariance matrices are used to perform PCA.*

Method	Number of Components (k = 3)	Number of choices (%)			
		n = 50		n = 25	
W	0	0		2	
	1	8		10	
	2	82		48	
	3	10		40	
	4	0		0	
$\tilde{W}$		b = 50	b = 100	b = 50	b = 100
	0	0	0	0	0
	1	0	0	0	0
	2	0	0	18	10
	3	86	86	82	90
	4	14	14	0	0
	Bootstrap Error	0.0850	0.0834	0.0636	0.0630

with 90% probability. When b = 50, $\tilde{W}$ chooses the correct number of components only with 82% probability. Thus, for a small number of bootstrap samples, decreasing the size of the sample seems to cause a slight drop in the performance of $\tilde{W}$. For a large number of bootstrap samples, the reverse seems to be true. Overall, $\tilde{W}$ performs much better than W, for both large and small samples. Also notable is the fact that for large samples, when $\tilde{W}$ fails to choose the correct number of components, it tends to choose one extra component than necessary, whereas for small samples, the alternative choice tends to be just short of the correct one. W's choice, however, tends to be too small, regardless of the size of the sample.

For both large and small samples in model 3 (Table 4.4), the methods behave almost similarly to their behaviour in model 1 (Table

Table 4.4 *Cross-validatory choice (W) and the bootstrap choice ($\tilde{W}$) of the number of components in model 3 when covariance matrices are used to perform PCA.*

Method	Number of Components (k = 3)	Number of choices (%)			
		n = 50		n = 25	
W	0	2		6	
	1	0		2	
	2	2		14	
	3	96		78	
	4	0		0	
$\tilde{W}$		b = 50	b = 100	b = 50	b = 100
	0	0	0	0	0
	1	0	0	0	0
	2	0	0	4	4
	3	100	100	96	96
	4	0	0	0	0
	Bootstrap Error	0.0804	0.0798	0.0619	0.0610

4.2). However, a change of behaviour can be seen for small sample cases, where W shows a slight improvement in its performance; choosing the correct number of components 78% of the time. For the remaining 22% of the time, W chooses 2 components (14%), 1 component (2%) and zero components (6%). For large samples, $\tilde{W}$'s performance is slightly better than it is in model 1; choosing the correct number of components *all* the time. However, its performance is also slightly worse than it is in model 1 for small samples; choosing the correct number of components with 96% probability. Nevertheless, for both large and small samples, $\tilde{W}$ still performs better than W. It can also be seen that as in models 1 and 2, for small samples, whenever $\tilde{W}$ fails to choose the correct number of components, there is a tendency for k being chosen smaller. However, W seems to exhibit

Table 4.5 *Cross-validatory choice* (W) *and the bootstrap choice* ($\tilde{W}$) *of the number of components in model 4 when covariance matrices are used to perform PCA.*

Method	Number of Components (k = 4)	Number of choices (%)			
		n = 50		n = 25	
W	0	0		0	
	1	0		0	
	2	60		62	
	3	40		38	
	4	0		0	
	Computational time (sec.)	483.90		178.50	
$\tilde{W}$		b = 50	b = 100	b = 50	b = 100
	0	0	0	0	0
	1	0	0	0	0
	2	0	0	0	2
	3	8	10	34	28
	4	60	62	60	66
	5	30	28	6	4
	6	2	0	0	0
	Computational time (sec.)	348.67	684.38	218.76	426. 33
	Bootstrap Error	0.1344	0.1322	0.1070	0.1059

this behaviour regardless of the sample size. It can also be deduced that the number of bootstrap samples is not an important factor in the performance of $\tilde{W}$.

It was noted in the previous section that data generated in accordance with model 4 will have 4 true dimensions. It is clear from Table 4.5 that W fails *completely* to detect this *dimensionality* of model 4 for both large and small samples. In other words, the choice of the correct number of components *does not* occur at all.

Its choice for k is mainly 2 or 3 components; choosing 2 components approximately 20% more than it chooses 3 components. Once again, $\tilde{W}$'s performance is far better than that of W. Its choice is mainly of the true number of components, occurring with 60% and 62% probability when b = 50 and b = 100 respectively for large samples, while this choice occurs with 60% and 66% probability for the small samples case. An increase in the number of bootstrap samples from b = 50 to b = 100, improves $\tilde{W}$'s performance only very slightly. Hence, $\tilde{W}$ chooses the true number of components, although the alternative choice for k tends to be larger by one component for large samples and smaller by one component for small samples. However, with the W technique, the choice for k tends to be too small for both large and small samples.

Next we shall focus on the behaviour of the two methods W and $\tilde{W}$ for choosing the number of components in cases where the PCs arise from *correlation* matrices. Both methods in model 1 (Table 4.6) behave nearly the same as when covariance matrices are used for PCA to choose the number of components (see Table 4.2). For small samples, W shows a slight improvement in its performance; retaining the correct number of components 74% of the time, compared to 72% in Table 4.2. However, this desired change of behaviour is also accompanied by a slight rise (6%) in the probability of W to choose zero components. Nevertheless, for large samples, W performs very well; choosing the correct number of components 94% of the time. On the other hand, the performance of $\tilde{W}$ remains superior to that of W for both large and small samples; choosing the correct number of components with 98% probability under *all* the sampling considerations (i.e., sample size and the number of bootstrap samples used in the analysis). This behaviour of $\tilde{W}$ is very similar

Table 4.6 *Cross-validatory choice (W) and the bootstrap choice ($\tilde{W}$) of the number of components in model 1 when correlation matrices are used to perform PCA.*

Method	Number of Components (k = 3)	Number of choices (%)			
		n = 50		n = 25	
W	0	4		24	
	1	0		0	
	2	2		2	
	3	94		74	
	4	0		0	
$\tilde{W}$		b = 50	b = 100	b = 50	b = 100
	0	0	0	0	0
	1	0	0	0	0
	2	0	0	2	2
	3	98	98	98	98
	4	2	2	0	0
	Bootstrap Error	0.0395	0.0388	0.0285	0.0282

Table 4.7 *Cross-validatory choice (W) and the bootstrap choice ($\tilde{W}$) of the number of components in model 2 when correlation matrices are used to perform PCA.*

Method	Number of Components (k = 3)	Number of choices (%)			
		n = 50		n = 25	
W	0	2		2	
	1	0		12	
	2	88		68	
	3	10		18	
	4	0		0	
$\tilde{W}$		b = 50	b = 100	b = 50	b = 100
	0	0	0	0	0
	1	0	0	0	0
	2	0	0	0	0
	3	70	70	90	90
	4	30	30	10	10
	Bootstrap Error	0.0353	0.0350	0.0258	0.0250

to its behaviour when covariance matrices are used to perform PCA in Table 4.2. However, its performance is slightly worse here for large sample sizes when b = 100 than the corresponding performance in Table 4.2.

For small samples, using correlation matrices for PCA in model 2 (Table 4.7) leads to a behaviour of W which is quite different from that for which the PCs arise from covariance matrices. However, when samples are of large size, the change of behaviour of W is very minor. Overall though, W still performs poorly; retaining the correct number of components only 10% of the time for small samples and with 18% probability for large samples. Its choice is mainly of 2 components; being with 88% probability when samples are of large size and 68% for small samples. $\tilde{W}$ chooses the correct number of components with 70% probability for large samples and with 90% probability for small samples, both when the number of bootstrap samples is 50 and when it is 100. Thus, even though for small samples using correlation matrices for PCA leaves the performance of $\tilde{W}$ nearly the same as when covariance matrices are used, there is a corresponding drop in its performance for large samples. Despite this behaviour, $\tilde{W}$'s performance remains superior over that of W for both sample sizes. It can also be seen that the number of bootstrap samples is not a factor in the performance of $\tilde{W}$, but its alternative choice of k is one additional component in both situations.

For small samples in model 3 (Table 4.8), use of correlation matrices for PCA yields a better performance of W than when covariance matrices are used, although there is a slight increase (4%) in the choice of zero components. W chooses the correct number of components with 90% probability compared to 78% when covariance matrices are used for PCA (Table 4.4). However, for large samples,

Table 4.8 *Cross-validatory choice* (W) *and the bootstrap choice* ($\tilde{W}$) *of the number of components in model 3 when correlation matrices are used to perform PCA.*

Method	Number of Components (k = 3)	Number of choices (%)			
		n = 50		n = 25	
W	0	0		10	
	1	0		0	
	2	4		0	
	3	96		90	
	4	0		0	
$\tilde{W}$		b = 50	b = 100	b = 50	b = 100
	0	0	0	0	0
	1	0	0	0	0
	2	0	0	0	0
	3	100	100	100	100
	4	0	0	0	0
	Bootstrap Error	0.0380	0.0380	0.0267	0.0266

the performance of W remains virtually unaffected when correlation matrices are used for PCA instead of covariance matrices, but it is still better than when samples are of small size. On the other hand, $\tilde{W}$ performs slightly better here for small samples than it does when covariance matrices are used to perform PCA for the same sample size. Once more, it performs better than W; choosing the correct number of components *all* the time for both levels of the number of bootstrap samples (b = 50 and b = 100) within each case of sample sizes (n = 25 and n = 50). Thus, $\tilde{W}$ is the better of the two methods for choosing the number of components. Furthermore, sample size and the number of bootstrap samples is not a factor in its performance.

When correlation matrices are used for PCA in model 4 (Table 4.9), the behaviours of W and $\tilde{W}$ differ from the corresponding

Table 4.9 *Cross-validatory choice (W) and the bootstrap choice ($\tilde{W}$) of the number of components in model 4 when correlation matrices are used to perform PCA.*

Method	Number of Components (k = 4)	Number of choices (%)			
		n = 50		n = 25	
W	0	0		0	
	1	0		0	
	2	6		14	
	3	60		60	
	4	34		26	
$\tilde{W}$		b = 50	b = 100	b = 50	b = 100
	0	0	0	0	0
	1	0	0	0	0
	2	0	0	0	0
	3	0	0	6	2
	4	28	28	54	58
	5	66	66	38	40
	6	6	6	2	0
	Bootstrap Error	0.0266	0.0266	0.0209	0.0208

behaviours when the PCs arise from covariance matrices. In fact, although the methods show such a change of behaviour in the other models, this change seems to be more prominent in model 4. The performance of W improves considerably; the choice being of the correct number of components (k = 4) 34% of the time for large samples and 26% for small samples. However, the choice of a smaller number of components than necessary remains high (66% for large samples and 74% for small samples). For small samples, $\tilde{W}$ performs slightly worse here than when covariance matrices are used for PCA; choosing the correct number of components with probability slightly less than 60% and mainly 5 components otherwise. Nevertheless, for this sample size (small) $\tilde{W}$ still outperforms W. At both levels of the number of bootstrap samples (*b* = 50 and *b* = 100) with large samples, $\tilde{W}$'s choice is mainly of 5 components (66%) and the correct number of components being chosen with only 28% probability. Thus, for large samples, W performs slightly better than $\tilde{W}$. Once again, we can notice that $\tilde{W}$ mainly adds one more component to the required number, while W's choice tends to be too small. Remembering that a smaller choice for *k* may lead to exclusion of essential components, it would seem that $\tilde{W}$ still performs better than W. One possible reason for the slight 'over-estimation' of *k* by the technique $\tilde{W}$ in model 4 is that in this model, the variables within each group (dimension) are not as strongly interrelated as the variables in the other models. Hence, the variables within each dimension are, in a sense, describing 'extra' dimensions (sub-dimensions), yielding an overall effect of slightly 'more' than 4 dimensions for this model.

It has already been noted that, in general, increasing the number of bootstrap samples from *b* = 50 to *b* = 100 yields very little and in some cases no improvement in the performance of $\tilde{W}$. In *all* cases (Tables 4.2 - 4.9), the corresponding decrease in the *bootstrap error* also seems negligible. Since, both levels of the number of bootstrap samples yield virtually the same results, the computational time of $\tilde{W}$ can be minimized by using the smaller of the two levels. However, in all cases, there is a considerable drop in the bootstrap error associated with an increase in sample size. Hence, increasing the size of the sample improves the results in terms of the bootstrap error. This confirms the pattern observed with models 1, 3 and 4 where the increase in sample size improves the probability of the methods to choose the correct number of components. While the bootstrap error when correlation matrices are used for PCA is generally less than the corresponding bootstrap error when covariance matrices are used for PCA, such a comparison is not sensible as the magnitudes of $\mathrm{PRESS}^{*m}(k)$, m = 1, 2, ..., b and k = 1, 2, ..., p-2 (see equation (4.1)) arising from each case (covariance or correlation matrices) would, in general, not be similar. In fact, while in some cases, the methods perform better when correlation matrices are used for PCA, this is not generally the case.

Finally, we compare the techniques W and $\tilde{W}$ on the basis of the amount of computational time needed to perform the analysis in each case for a given set of sampling considerations. The most computationally intensive procedure common to both techniques is the SVD of a matrix. With W, when the data matrix X is estimated from only the first k PCs, the estimates of the x_{ij} (i = 1, 2, ..., n; j = 1, 2, ..., p) elements of X are computed separately and *two* SVDs are required; one for X without the i-th row and the other for X without the j-th column. Hence, np SVDs of matrices each with size (n-1 x p) and np SVDs of matrices each with size (n x p-1) are required; the total being 2np SVDs. Turning to our proposed bootstrap technique $\tilde{W}$, for each bootstrap sample, (p-1) SVDs are required; one for decomposing the m-th bootstrap sample (X^{*m}) itself and the other for decomposing $X^{*m(k)}$, k = 1, 2, ..., p-2. Hence, b + b(p-2) SVDs are required in total. Note that for both W and $\tilde{W}$, no SVD is required when k = 0. As an illustration, first consider the case where n = 25, p = 6 and b = 50. In this case, $\tilde{W}$ requires 250 SVD's while W requires 300 SVDs. It is therefore expected that $\tilde{W}$ would take a shorter computational time than W. In the simulation experiment corresponding to this case (Table 4.2), $\tilde{W}$ is slower; taking 61.46 seconds while W takes 53.39 seconds. One possible explanation for this is that with W, the SVDs are of smaller matrices. If the number of bootstrap samples is increased to b = 100, $\tilde{W}$ would require 500 SVDs, but W still requires 300 SVDs. This time, W is computationally faster than $\tilde{W}$ as we can see in Table 4.2, $\tilde{W}$ takes 108.37 seconds while W still takes 53.39 seconds. Next, consider the case where n = 50, p = 10 and b = 50. Here, $\tilde{W}$ requires 450 SVD's while W requires 1000 SVDs. Hence, we would expect $\tilde{W}$ to be faster computationally. Table 4.6 shows that $\tilde{W}$ takes a shorter time (348.67 seconds) to complete the analysis than W (483.90 seconds), as expected. Increasing the number of bootstrap samples to b = 100 leads 900 SVDs and the analysis for $\tilde{W}$ takes 684.38 seconds. It must be realized, however, that according to the Monte Carlo simulation study, there is very little and in some cases no improvement in the quality of the results due to an increase in the number of bootstrap samples. Hence, the computational time of $\tilde{W}$ can be kept at a minimum by using a considerably small number of bootstrap samples without a serious loss of power in the ability of $\tilde{W}$ to choose the correct number of components.

4.4 Conclusions

The results of the Monte Carlo simulation study in the previous section demonstrate the general usefulness of the bootstrap ($\tilde{W}$) and the cross-validatory (W) techniques for choosing the number of components in PCA, although $\tilde{W}$ receives more support than W. However, preliminary comparisons of these techniques with the B, E and T criteria for choosing the number of components discussed in section 2.3 (not presented here) showed that W and $\tilde{W}$ often give results that differ from those that would be obtained from B, E and T. It seems that the central idea of the techniques $\tilde{W}$, W and R (i.e., the cross-validatory choice of the number of components suggested by Wold (1978)) is that of adequate representation of the data using the first few components. In other words, if k (= 1, 2, ..., etc.) is consecutively the number of components retained in a model designed to represent the data, the aim is to determine the smallest value of k for which the retained components can be used to 'reproduce' the data with sufficient accuracy. This suggests that the purposes of performing PCA must be borne in mind when choosing the optimum method. If the aim is simply to describe the variability in the data, then the criteria, B, E and T are appropriate. However, if it is anticipated that the PCs might be used in future analyses, then our proposed bootstrap technique $\tilde{W}$ and the cross-validatory techniques R and W should be preferred. For example, as noted in Chapter 1, one objective of PCA is to reduce the *dimensionality* of the data without disturbing its overall sample features. This can be achieved by retaining *only* the first few PCs, as the PCs are determined in such a way that the first few include most of the variation (information) in the data. However, the simplicity of working with the original variables instead of their PCs suggests that there is a need to incorporate the selection of subsets of important variables in PCA. In this case, the objective is to choose subsets of the original variables which contribute most of the sample information in the data. Hence, the criteria used to select the variables should utilize the first few PCs, sufficient for adequate representation of the data. Thus, it seems preferable to use the techniques $\tilde{W}$, W or R to make the choice of the number of PCs to be used for this purpose.

While, the results in Tables 4.2 - 4.9 show that, in some cases using correlation matrices for PCA instead of covariance matrices slightly improves the performance of the techniques $\tilde{W}$ and W, in other cases the reverse is true. Hence, there is no general indication whether the correlation or the covariance matrices should be used for PCA in conjunction with $\tilde{W}$ or W. In general, $\tilde{W}$ performs satisfactorily, although there is a slight tendency for k to be chosen slightly too small when samples are of small size and slightly too large for large samples. The tendency for the choice of k to be too large seems to be greater than its tendency to be too small. On the other hand, depending on the degree to which the data set to be analysed has a clear-cut indication of its dimensionality, W also performs satisfactorily for large samples, although there is a slight tendency for k being chosen too small. However, for small samples, W frequently yields an *under-estimated* choice of the number of components. A similar experience has been reported by Krzanowski (1983) who examined the behaviour of W over different types of structure for the *covariance* and *correlation* matrices, and reached the conclusion that W chooses about the right number of PCs in each case, although there is a tendency for k to be chosen too small. His study and the earlier study conducted by Eastment and Krzanowski (1982) also found that, in some cases, one or more values of the statistic W are very close, but less than the cut-off point unity, making it unclear whether or not to retain the components corresponding to these values. In such cases, these authors chose to move the cut-off point slightly below unity, i.e., 0.9. The main idea behind this approach is to overcome the problem of *under-estimation* of the number of components. Furthermore, Krzanowski (1983) argued that, while the cut-off point at $W = 1$ seems reasonable, the reasoning behind it is not rigid and it could be slightly relaxed to account for sampling variation. Wold (1978) also found, in a small simulation study that R has the tendency to retain too few PCs. This feature is undesirable because there is a greater danger in components with vital information for adequate description of the data being ignored. Hence, $\tilde{W}$ for which this feature is largely less prominent, is more appropriate for choosing the number of components. In fact, the study shows that $\tilde{W}$ generally performs better than W, particularly for small sample cases. The slight tendency for $\tilde{W}$ to choose one extra component than necessary for large samples seems to be an advantage over the tendency for W to choose a number too small, because with $\tilde{W}$, the loss of essential components is not likely. The simulation study also shows that for a small number of bootstrap samples, $\tilde{W}$ takes slightly longer to be computed than W when samples are of small size; and approximately 1.3 times quicker than W to complete the analysis, for large samples. For a large number of bootstrap samples, $\tilde{W}$ is generally slower than W. However, it is also evident from this study that an increase in the number of bootstrap samples yields very little and in some cases no improvement in the ability of $\tilde{W}$ to choose the correct number of components. Hence, the computational time for $\tilde{W}$ can be minimized by using a considerably small number of bootstrap samples without effectively decreasing its performance. Having stated the advantages of $\tilde{W}$ over W (and R), we would strongly recommend this $\tilde{W}$ as the criteria for choosing the number of components in PCA.

CHAPTER 5
REVIEW OF LITERATURE ON VARIABLE SELECTION IN PRINCIPAL COMPONENT ANALYSIS

5.1 Introduction

One of the major aims of the current research project is, to explore the choice of subsets of the original variables of some given data in a PCA, that will retain the overall features or the multivariate structure present in the entire set of variables. As a lead-in to this investigation, this Chapter focuses on the currently existing criteria for selecting such subsets of variables. An extensive amount of literature is available in the area of variable selection in the context of *Regression Analysis* and *Discriminant Analysis*. The criteria used for variable selection in these techniques are, most naturally, based on minimizing the residual mean square error in the case of Regression Analysis and error rate of misclassification in Discriminant Analysis. A list of authors who have addressed the problem of variable selection in these areas has already been given in section 1.3.2. However, very little seems to have been done in the context of variable selection with reference to PCA.

A variable selection technique which, however, is not directly linked with PCA makes use of *Principal Component Regression* and has already been mentioned briefly in section 1.3.2. In this technique, unlike in 'ordinary' PCA, the variables do not arise on an 'equal footing'. In essence, the variables arise in the regression context where there is a *response* variable which is dependent on a set of *predictor* variables. In cases where the predictor variables are linearly or near-linearly dependent on each other (a problem referred to as *multicollinearity*), it is common practice to transform these variables into PCs (which are orthogonal) and then regress the response variable on these new variables (the PCs). The resulting regression equation can then be interpreted in the usual way. Apart from overcoming the problem of multicollinearity, and the higher interpretable nature of the regression equations that arise from PC regression (compared to equations that arise from untransformed predictors), this approach provides an alternative way of selecting subsets of the original predictors to include in a regression equation. Subsets of selected variables are those for which the coefficients between the corresponding PCs and the response variable are statistically different from zero. Details of this approach can be found in Jolliffe (1986). However, since variables in this approach arise in the regression rather than the PCA context, we shall end its discussion at this point and focus on techniques that use the PCs directly, to select subsets that retain adequate sample information or variation in the data.

Jolliffe (1972, 1973) suggested several methods of choosing the best subset of *q* variables which preserve the main sample features in a 'Principal Component' manner. McCabe (1984) presented various optimality criteria on which selection of variables containing the maximum sample variation can be based. These methods are discussed in section 5.2.1 of this Chapter. Krzanowski (1987) suggested a technique based on *Procrustes Analysis* for choosing subsets of the original variables which carry the overall features of the sample, present in the entire set of variables. Details of this technique are presented in section 5.2.2 followed by a review of comparative studies on variable selection in PCA in section 5.3.

5.2 Some existing criteria for the selection of variables in PCA

5.2.1 Criteria based on eigen_analysis and various other optimality criteria

The variable selection techniques discussed in this section have also been mentioned briefly in section 1.3.2. A detailed account of these techniques can also be found in Jolliffe (1972, 1986).

The first technique we shall discuss , henceforth referred to as B1, was first suggested by Beale *et al.* (1967). In this method the main aim is to choose a number (say, l_o) that is used in the *stopping rule* for the sequential rejection of variables. First, PCA is performed using *all* the original p variables and if p_1 eigenvalues are less than l_o, the corresponding eigenvectors (PCs) are considered in turn beginning with the PC whose corresponding eigenvalue is the smallest, then the PC corresponding to the second smallest eigenvalue and so on until the p_1^{th} PC. The original variables are then associated with the p_1 PCs (eigenvectors) and the first p_1 variables which have the largest coefficients, in absolute value, with these PCs discarded. A second PCA is carried out on the remaining $(p-p_1)$ variables. Once again, if p_2 of the eigenvalues are less than l_o, the original variables are associated with each of the corresponding components, and these p_2 variables are discarded. A third PCA is then performed on the remaining $(p-p_1-p_2)$ variables and this procedure is continued until *all* the eigenvalues in the latest PCA are greater than l_o. The procedure is stopped at this stage with the number of variables having dropped to say, k which depends on the choice of l_o.

The next four methods, namely, J1, J2, J3 and J4 that we will discuss, require a single PCA to be performed on the entire set of p variables, hence, they are computationally faster than B1. The method J1 is the same as B1 except that only the first PCA (using *all* the variables) is performed. If k variables are to be retained, then $(p-k)$ variables which have the largest coefficients with the last $(p-k)$ PCs are discarded. In the method J2, the *sum of squares* of the coefficients in the last $(p-k)$ PCs is computed for each variable. If these sums of squares are sorted in descending order, the variables corresponding to the first $(p-k)$ sums of squares in the sorted list are then discarded. Thus, J2 retains those variables for which the sum of squares

$$\sum_{j=k+1}^{p} c_{ji}^2, \tag{5.1}$$

is *minimal*. Here, c_{ji} is the coefficient of the i-th variable in the j-th PC. The next method, J3 is computationally similar to J2 and it retains those variables which are best predictable from the first k PCs. When the correlation matrix is used for PCA, the proportion of variation explained by the i-th variable, predictable from the first k PCs is given by

$$\sum_{j=1}^{k} l_j \, c_{ji}^2, \tag{5.2}$$

where l_j is the j-th eigenvalue. Thus, by using the method J3 we choose those k variables associated with first k largest values of this proportion of variation. Among the methods based on *eigen_analysis*, the last method we shall discuss, namely J4, is in some sense, related to J1. The difference is that with J4, the retained variables are those whose coefficients in the first k PCs are the largest and the remaining $(p-k)$ variables are discarded.

Note that, the choice of the number k in the methods J1 - J4 is critical. In the same way that the value of k is determined by the choice of l_o in the method B1, this choice in J1 - J4 could be equated with the number of eigenvalues greater than some l_o. Alternatively, k could be set equal to the minimum number of PCs for which the proportion of variation explained is greater than some α_o.

Next we discuss the various optimality criteria suggested by McCabe (1984) for selection of variables in PCA. McCabe's approach has already been mentioned briefly in section 1.3.2. In his approach, McCabe (1984) used the fact that the PCs satisfy a number of optimality criteria and termed subsets of the original variables which optimize one of these criteria as sets of *principal variables*. A traditional property of PCs is that they *maximize* the variance subject to a set of constraints. Equivalent to this property is the concept that for a given number of PCs, the sum of the variances is the largest possible. Since the eigenvalues, l_j, j = 1, 2, ..., p, are the variances corresponding to the PCs, this property of maximum total variance of the first k PCs is given by the criterion

$$\text{Max} \sum_{j=1}^{k} l_j. \tag{5.3}$$

Thus, a set of k of the original variables for which this criterion is optimized satisfies the property of maximum total variance of the first k PCs. Notice that this criterion is equivalent to criterion (1.5b) (i.e., criterion (b) of the set of criteria (1.5) in section 1.3.2). In other words, maximizing the retained variation, represented by criterion (5.3) is equivalent to minimizing the lost variation, represented by criterion (1.5b). Since the determinant of the covariance matrix is the generalized variance, the criterion

$$\text{Max} \prod_{j=1}^{k} l_j, \qquad (5.4)$$

is similar to criterion (5.3), though the variance here is expressed in a *multivariate* sense. In a similar way that criterion (5.3) is equivalent to criterion (1.5b), this criterion (5.4) is equivalent to criterion (1.5a). Following a similar argument to the ones just presented, a subset of k of the original variables which optimizes criterion (1.5c), satisfies the property of maximum cumulative variance of the first k PCs. Such a subset also represents the variables which are best predictable from the retained k PCs. Notice that, for this reason, criterion (1.5c) is somewhat similar to method J3 among the methods in Jolliffe (1972, 1973). Criterion (1.5d) can also be viewed as arising from the concept of maximum cumulative variance of the first k PCs. The sum of the squared *canonical correlations* between the retained k variables and the discarded $(p-k)$ variables, represents the amount of variation in the retained variables predictable from the discarded variables. Hence, for a given k, the larger the value of criterion (1.5d), the higher the redundancy in the set of discarded variables and the smaller the total variance explained by these variables. Now, recalling from section 2.2.1 that the sum of the variances of *all* variables in a given set of variables is the same as the sum of the variances of *all* the corresponding PCs, the total variance of the $(p-k)$ PCs corresponding to the discarded variables will be small. This variance, in turn, corresponds to a larger total variance of the k PCs corresponding to the retained variables.

5.2.2 The M^2-procrustes criterion

When Krzanowski (1987) applied the variable selection criteria in Jolliffe (1972, 1973) and McCabe (1984) on the *Alate aldeges* data (described in section 1.3.2), he noted that while the subsets of variables chosen by these criteria satisfy several optimality conditions, they fail to satisfactorily 'reproduce' the overall structure of the complete data (with the entire set of variables). He argued that, in practice, an investigator would be interested in those subsets of the original variables which can 'reproduce' as closely as possible, the general features of the data with the entire set of variables. One possible explanation for the failure of Jolliffe's and McCabe's methods to recover the *multivariate structure* of the complete data is that, they are concerned with overall features, either of the complete data (in the case of Jolliffe (1972, 1973)) or of the subset data (in the case of McCabe (1984)). Hence, they lack an appropriate criterion to preserve structure (such as *groupings*) among units (individual data points). A suitable criterion would involve comparisons between individual data points of the complete data configuration and the corresponding points of the subset configuration. To meet this requirement, Krzanowski (1987) considered a criterion based on *Procrustes Analysis*. Procrustes Analysis is a long standing technique for comparing two *n*-point configurations which owes its name to Hurley and Cattel (1962), but its origins date back at least as far as a paper by Moisier (1939). Chief references in this area include Green (1952), Schönemann (1966, 1968), Schönemann and Carroll (1970), Gower (1971, 1975), Krzanowski (1971, 1979, 1982, 1988), Sibson (1978) and Peay (1988). Gower (1975) generalized the idea of Procrustes Analysis so that a single analysis can allow the investigator to compare more than two configurations simultaneously. However, in our current research project, the term Procrustes Analysis shall be used to refer to the comparison of *two* configurations *only*. This is because, our present approach to variable selection in PCA requires the computation of a measure of the relationship between the subset configuration and the complete data configuration (in this case, using Procrustes Analysis). Thus, at each stage of the variable selection procedure, only two configurations are involved. The application of Procrustes Analysis to the variable selection problem (with reference to PCA) will now be outlined.

Let X be the $(n\mathrm{x}p)$ data matrix, consisting of p variables measured on each of n individuals in the sample, and Y be the $(n\mathrm{x}k)$ transformed data matrix of PC scores yielding the best k-dimensional approximation to the original data. The value of k may be determined using either $\tilde{W}$ (i.e., the proposed bootstrap technique described in Chapter 3) or W (i.e., the cross-validatory technique suggested by Eastment and Krzanowski (1982) described in section 2.2.3). Similarly, let $\tilde{X}$ denote the $(n\mathrm{x}q)$ *reduced* data matrix which retains only q selected variables, and $\tilde{Y}$ be the corresponding $(n\mathrm{x}k)$ matrix of PC scores. It should be noted that since k components may be sufficient to model the 'signal' in the data, the remaining p-k dimensions are a reflection of the 'noise'. Hence, it would seem reasonable to set q, the number of variables to retain, to k. Krzanowski (1987) worked with the *Procrustes* criterion

$$M^2 = \mathrm{Trace}\left(Y^TY + \tilde{Y}^T\tilde{Y}\right) - 2\ \mathrm{Trace}(D), \qquad (5.5)$$

where, M^2 is the sum of squared differences between corresponding points of the configurations Y and $\tilde{Y}$; and D is the $(k\mathrm{x}k)$ diagonal matrix obtained from the SVD of $\tilde{\mathbf{Y}}^{T}\mathbf{Y}$. In fact, **D** corresponds to the matrix **S** in the SVD equation (1.4) in section 1.2. M^2 is computed to maximize the *congruence* between the two configurations *matching* under *translation* (i.e., to mean-center both **Y** and $\tilde{\mathbf{Y}}$), *rotation* and *reflexion*. In this process, the configuration of **Y** is fixed first (*naturally*) and the subset configuration $\tilde{\mathbf{Y}}$ is transformed. This is because it is desired to make $\tilde{\mathbf{Y}}$ look like **Y** as much as possible, and *not* vice versa. Krzanowski (1987) claimed that this residual sum of squares is therefore a measure of *loss of information* of the original data structure when only q instead of all p variables are utilized. Hence, the 'best' subsets of retained variables are those for which M^2 is minimized.

5.3 Comparative studies on variable selection in principal component analysis

Jolliffe (1970, 1972) investigated the methods J2 and J3 and demonstrated that their performance is unsatisfactory, since they consistently fail to choose appropriate subsets of variables for data with simple correlation structures. For example, if the data are simulated in such a way that there are k groups of variables and within each group the variables are highly intercorrelated, with variables from different groups being independent, satisfactory subsets can be obtained by including *one* variable from each group. In such cases, the method J2 tends to reject all the variables from one of the groups, and hence the retained subset will not contain any of the variables from this group (an unsatisfactory feature).

A difficulty encountered with method J4 is that, even if the value of k is known (from using l_o or α_o), among the first k PCs a variable may have large coefficients in some PCs and small coefficients in other PCs. Hence, unless a few variables have large coefficients in the first *k* PCs, by choosing the variables with the first *k* largest coefficients we may be loosing the structure (or interpretability) of the PCs.

Jolliffe (1972) used simulated data to compare the methods J1 and J4 and several other variable subset selection methods based on *clustering techniques* which, however, do not use the PCs. Jolliffe constructed *five* models using randomly generated variates which are *identically* and *independently distributed* (*iid*) as N(0,1), in such a way that within each model the constructed variables fall into groups. The variables within each group are linear combinations of each other (plus random disturbances), while variables from different groups are independent. Hence, if a group contains *t* variables, *t*-1 of these variables are redundant. For each model, Jolliffe ranked the subsets as *best*, *good*, *moderate* or *bad* according to the extent to which they maximize the minimum *multiple correlation* between the selected variables and any of the rejected ones. The results of applying the various methods on the simulated data suggested that the methods J1 and J4 retain the 'best' subsets more frequently than the other methods considered. However, the probability of J1 and J4 to choose 'bad' subsets as opposed to 'good' and 'moderate' subsets was also shown to be higher than for the other methods. In particular, the method J4 was also shown to be peculiar, choosing the 'best' and 'bad' subsets more than it chooses 'good' and 'moderate' subsets.

Jolliffe (1973) applied the methods above on *four* data sets, namely, *Pitprops*, *Alate*, *Crime rates* and *British towns*. The *Pitprops* data consist of 13 variables measured on 80 pitprops of Corsican pine. A full list of the variables, and their correlation matrix, can be found in Jeffers (1967). As noted in sections 1.3.2 and 4.1, the *Alate* data consist of 19 variables measured on 40 alate adelges (winged aphids). Here, the object is to determine the number of distinct *groups* (possibly species) of the aphids in the sample. The *Crime rates* data, on the other hand, consist of 18 variables (i.e., the number of crimes committed in the U.K. within 18 different categories) measured for each of the 14 years 1950 - 1963. The final set of data studied by Jolliffe (1973) consists of 57 sociological variables measured for each of 157 *British towns* with a population greater than 50,000 in 1951. In each of the cases above, it was suspected that fewer variables can yield PCA results *similar* to those obtained from PCA using the entire set of variables. Jolliffe used *two* types of measures of *similarity* to assess the performance of the subsets of variables retained by the various methods. For the *first three* sets of data, he used a weighted average of the largest *simple correlation coefficient* between each of the first few (say k) PCs from the full set data and any of the first k PCs from the reduced set data. The weights are proportional to the relative importance of the first k PCs in the full set data. However, for the *fourth* data set, which is much larger, he chose to use a measure similar to the one above, except that this measure is based on a *rank correlation coefficient*, which is easier to calculate. In this investigation, Jolliffe found that, while none of the methods is uniformly the best, the methods J1 and J4 identify reasonable subsets in most cases.

McCabe (1984) used *two* sets of data to evaluate the performance of his *principal variables* method. The first data set is due to Fisher (1936). There are 4 *size* measurements (variables) on 50 subjects of *Iris versicolor*. The second set of data is taken from a study of the constituent elements in coal samples conducted by

Orheim (1981). Nine elements (variables) were measured on 50 samples. In each case, McCabe used the criteria (1.5a) and (1.5b) to calculate the proportion of variation explained by the retained subsets of variables for *all possible subsets* (in the case of criterion (1.5a)) and using the *forward selection* procedure (in the case of criterion (1.5b)). He compared these proportions with the proportion of variation explained by the first few PCs. Using this approach, subsets of retained variables are those for which the corresponding proportion of variation explained is nearly as large as the proportion of variation explained by the first few PCs. In the examples above, McCabe found that his *principal variables* method chooses subsets of variables with properties (explained variation) similar to those of the PCs for a given number of dimensions. As noted before, by definition, the PCs maximize the variation explained in the data for a given number of dimensions. Hence, a subset of the original variables with the same size as the number of retained PCs cannot explain more variation than the variation explained by the PCs. However, McCabe concluded that if a small number of additional variables is used, more comparable results can be obtained. He also argued that, among his criteria for selecting principal variables, criterion (1.5a) is the only computationally feasible choice if all possible subsets are to be explored. Despite this fact, criterion (1.5b) can be incorporated into a suitable stepwise variable selection strategy. Apart from the descriptions of the criteria (1.5c) and (1.5d), these criteria received no further investigation in McCabe's paper.

Although Krzanowski (1987) concluded that his M^2-*procrustes* criterion successfully identifies the *structure-bearing* variables, the results were optimal particularly when groups were present in the data. His study was based on a single real data set and a well defined simulation of artificial data sets with *large* samples. It is important to point out that our objective in the current research project is to select subsets of variables which preserve any (*unknown*) structure that may be present in the data. While Krzanowski's approach is robust towards preserving *group-structure*, it lacks a criterion for preserving or even enhancing other interesting multivariate patterns of the data. Such a criterion would make a natural comparison between the configuration of the complete data and the subset data without forcing one configuration to fit the other (a feature present in the procrustes method), in an attempt to find whatever natural relationship that exists between the two configurations. Detailed descriptions of several such criteria are given in Chapter 6 of this Thesis. The behaviour of these criteria under different *sample sizes* should also be examined. This objective is met in Chapter 7.

CHAPTER 6
THE PROPOSED CRITERIA FOR VARIABLE SELECTION IN PRINCIPAL COMPONENT ANALYSIS

6.1 Introduction

In order to identify *structure-bearing* variables using PCA, our approach in the current research project is, on the one hand motivated by the long standing and well established technique of *Canonical Correlation Analysis*, and a *Graph-Theoretic* approach using *interpoint-distances* on the other. In the former case we use *measures of multivariate association* (MVA) based on *canonical correlations* as criteria for selecting variables in PCA, whereas in the latter we introduce criteria which measure the discrepancy between corresponding interpoint-distances in the two configurations $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$ (as defined and described in section 5.2.2). The idea in both cases is to maximize the similarity or 'overlap' between S_{Y} and $S_{\tilde{\mathrm{Y}}}$, the spaces spanned by the sets of PCs $\underline{Y}$ and $\underline{\tilde{Y}}$, respectively. The best subset of retained variables can then be regarded as that for which this *similarity* is optimal. Notice that $\underline{Y}_{(k\times1)}$ and $\underline{\tilde{Y}}_{(k\times1)}$ are the vectors of PCs associated with the PC score matrices $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$, respectively. In other words the set of PCs $\underline{Y}$ arises from the full set data while the set $\underline{\tilde{Y}}$ arises from the subset data. Details of the proposed criteria are presented in section 6.2 (for canonical correlations based criteria) and section 6.3 (for graph-theoretic criteria).

6.2 Criteria based on canonical correlations

Initially, we shall consider the details of criteria based on measures of multivariate association. The study of the relationship between two sets of variables inevitably brings to mind Hotelling's

(1936) method of *canonical analysis* (CA). Literature in this area include Anderson (1958), Rozeboom (1965), Levine (1977), Knapp (1978), Mardia *et al.* (1979), Muller (1981), Muller (1982), Chatfield and Collins (1986) and Gittins (1985). Levine (1977) and and Muller (1982) stated the advantages of performing preliminary orthogonalization (say, PCA) of each of the two sets of variables to be compared *prior* to Canonical Correlation Analysis, one of which is that it eliminates the problem of collinearity among variables within a set.

Now let $\tilde{Z} = \begin{bmatrix} Y & \tilde{Y} \end{bmatrix}$ be the (nx$2k$) partitioned matrix which arises from the horizontal concatenation of the matrices of PC scores Y and $\tilde{Y}$. Denote the corresponding ($2k$x1) matrix of PCs by $\underline{\tilde{Z}} = \begin{bmatrix} \underline{Y} & \underline{\tilde{Y}} \end{bmatrix}^T$. Then the corresponding ($2k$x$2k$) partitioned correlation matrix between the PCs $\underline{\tilde{Z}}$ can be written as

$$\mathbf{R} = \left[\begin{array}{c:c} R_{YY} & R_{Y\tilde{Y}} \\ \hdashline R_{\tilde{Y}Y} & R_{\tilde{Y}\tilde{Y}} \end{array}\right]. \tag{6.1}$$

Here, $R_{Y\tilde{Y}}$ is the (kxk) matrix of correlations between the PCs of the set $\underline{Y}$ and of $\underline{\tilde{Y}}$, and since the correlation matrix, **R** is *symmetric*,

$$R_{Y\tilde{Y}} = R^{T}_{\tilde{Y}Y}. \tag{6.2}$$

Furthermore, since the PCs in $\underline{Y}$ are orthogonal to each other and similarly, the PCs in $\underline{\tilde{Y}}$ are orthogonal to each other and therefore uncorrelated,

$$R_{YY} = R_{\tilde{Y}\tilde{Y}} = I_{k}, \tag{6.3}$$

where, I_k is the $(k{\times}k)$ identity matrix. Now, using equation (3.9) in section 3.2, the *squared canonical correlations* between the two sets of PCs $\underline{Y}$ and $\underline{\tilde{Y}}$ are given by the eigenvalues of

$$R^{-1}_{YY}\ R_{Y\tilde{Y}}\ R^{-1}_{\tilde{Y}\tilde{Y}}\ R_{\tilde{Y}Y}. \tag{6.4}$$

Using equations (6.2) and (6.3) in (6.4), these squared canonical correlations are the *k* eigenvalues of

$$R_{Y\tilde{Y}}\ R^{T}_{Y\tilde{Y}}, \tag{6.5}$$

arranged in descending order. Clearly, working with expression (6.5) instead of expression (6.4) makes computations easier and faster and this is one of the attractive features of the canonical correlations approach to the selection of important variables in PCA. The canonical correlations can also be interpreted as the simple correlations between linear combinations of the PCs of the set $\underline{Y}$ and those of set $\underline{\tilde{Y}}$, computed in a specific manner. Such linear combinations are usually referred to as *canonical variates*. The maximum canonical correlation between the sets of PCs $\underline{Y}$ and $\underline{\tilde{Y}}$ (say, ρ_1) is a possible choice for a *symme*tric measure of MVA (*Multivariate Association*) between Y and $\tilde{Y}$. A problem with this measure, however, is of course that after partialling out the two concomitant canonical variates from the sets, the residualized variates may still be correlated.

For a reasonable index of the total association between the sets (of variables), it would seem appropriate to combine in some way the successive canonical correlations which can be extracted. Cramer and Nicewander (1979) and some other authors (Coxhead (1974), Cohen (1982), Serlin (1982) and van Den Burg and Lewis (1988)) have defined various combinations and shown that these are robust measures of MVA. While these measures may yield different values for a given situation, initial investigations of our study revealed that some of these measures have similar behaviours when used as criteria for selecting variables in PCA. For this reason we will consider only two of the indices recommended by Cramer and Nicewander (1979), as criteria for the present purpose. These measures are

$$\hat{\gamma}_5 = 1 - \left[\prod_{j=1}^{k}\left(1 - \hat{\rho}_j^2\right)\right]^{1/k}, \tag{6.6}$$

and

$$\hat{\gamma}_6 = \frac{1}{k}\left[\sum_{j=1}^{k}\hat{\rho}_j^2\right], \tag{6.7}$$

where $\hat{\rho}_1, \hat{\rho}_2, \ldots, \hat{\rho}_k$ are the canonical correlations between the sets of PCs $\underline{Y}$ and $\underline{\tilde{Y}}$ arranged in descending order.

Three of the most useful properties of these γ-measures, as pointed out by Cramer and Nicewander (1979) are that:

(a) They are *symmetric*, i.e., yield the same value for **Y** related to $\tilde{\mathbf{Y}}$ as for $\tilde{\mathbf{Y}}$ related to **Y**,

(b) They are *invariant*, in that a linear transformation of the PCs of the set $\underline{Y}$ or of $\underline{\tilde{Y}}$ does not change the value of the measure, and

(c) They have a clear *proportion of variance interpretation*. The *average squared canonical correlation*, $\hat{\gamma}_6$, gives the average proportion of variance of the *canonical variates* arising from the PCs in $\underline{Y}$ predictable from the PCs of the set $\tilde{\underline{Y}}$. The index $\hat{\gamma}_5$, has a similar interpretation, where, the geometric mean of $1-\hat{\rho}_i^2$ is the unpredictable variance, and of course one minus this quantity yields the variance of the *canonical variates* of the PCs in $\underline{Y}$ predictable from the PCs of $\tilde{\underline{Y}}$. This is an attractive property of the two indices $\hat{\gamma}_5$ and $\hat{\gamma}_6$, since in PCA preserving the maximum sample variation constitutes the major aim of the analysis. Hence, as we are interested in assessing the strength of *association* between the two sets of PC scores, these γ-measures constitute appropriate criteria for selecting variables in PCA.

While Krzanowski's procrustes-M^2 criterion chooses subsets of variables to preserve the (*group-*)*structure* among individual points, the criteria based on MVA described above, choose subsets to preserve the original sample variation and the canonical properties of the data. Since the MVA criteria give a broader set of additional diagnostically useful statistics and tests of significance, they gain favour as the methods of choice over *Procrustes Analysis* for this purpose. Overall (1974) noted that *Procrustes Analysis* tries to confirm a particular hypothesis rather than to falsify it. Conventionally, it is unsatisfactory to claim the validity for a hypothesis after having forced the data to fit the *hypothesis*. In some way, such *Procrustes Rotation* forces empirical data to fit the target and hence the method is vulnerable to making almost any data to fit almost any hypothesis. A Monte Carlo study which was conducted by Nesselroade and Baltes (1970) confirms this fact.

6.3 Graph-theoretic criteria

The criteria we propose next is based on the interdependence among individuals within each of the configurations **Y** and $\tilde{\mathbf{Y}}$. This is a *distance* based approach, and since the PC scores (say, **Y**) can be regarded as a set of *n* points P_i (i = 1,2,...,*n*) in the reduced *k*-space, we may use the *Euclidean distance* d_{ij} between points P_i and P_j. The spatial configuration of the sample is of interest only if d_{ij} satisfactorily measures the similarity between the i-th and j-th individuals. When the PCs within the set reflect different scale measurements, d_{ij} becomes an unsatisfactory distance measure. For example, suppose *one* of the original variables is nearly independent of *all* the others and that the remaining variables are highly intercorrelated, measuring essentially the same property. The *Euclidean distance* will give more weight to this property than to the property described by the 'independent' variable. To evade this difficulty, it is common practice to renormalize the PCs by dividing each by its sample standard error. This way, the PCs are given equal weight and so is each property measured by the original variables. Gower (1966), noted that since d_{ij} makes no attempt to allow for correlations, it has similar properties to distance measures based on various similarity coefficients currently in favour in classification work, and hence can be contrasted with distances used in *Discriminant Analysis*, such as *Mahalanobis distance*. Let d^*_{ij} be the corresponding distance between the i-th and the j-th individuals in the $\tilde{Y}$ configuration. Our motive is to choose subsets for which the discrepancy between the corresponding interpoint distances d_{ij} and d^*_{ij} is *minimal* under pre-specified constraints. Since such constraints may have *Graph-Theoretic* implications, we begin by reviewing some terms from *Graph Theory*.

Consider a graph for which every observation point is a *node*, and each node pair defines an *edge*. Such a graph is called *complete graph* and has *n* nodes and *n(n-1)/2* edges. Assign the *Euclidean distance* d_{ij} as a weight to the edge between the i-th and the j-th nodes. A *subgraph* of a given graph is a graph with all its nodes and edges falling within the given graph. A *Spanning subgraph* has its node set identical to the node set of the given graph. Friedman and Rafsky (1979, 1983), proposed *Graph-Theoretic Measures of Multivariate Association and Prediction* based on the number of edges shared by the two spanning subgraphs constructed independently from two sets (Y and $\tilde{Y}$ in our context), respectively. Like these authors the spanning subgraphs that we have found useful are the *K nearest neighbour graph* (*K*NN) and the *K minimal spanning tree* (*K*MST). The former (*K*NN) is simply a spanning subgraph that has an edge between each point and its *K* closest points, while the latter (*K*MST) is as defined below.

A *path* between two prescribed nodes is an alternating sequence of nodes and edges with the prescribed nodes as first and last elements, all other nodes distinct, and each edge linking the two nodes adjacent to it in the sequence. Now define a *cycle* as a path beginning and ending with the same node. A *spanning tree* of a graph is a spanning graph that forms a tree, i.e., has no cycles. An *edge weighted graph* is a graph with a real number assigned to each edge. A *minimal spanning tree* (MST) of an edge weighted graph is a spanning tree for which the sum of edge`weights is minimum. A second MST is said to be *orthogonal* to the first, if it connects all of the nodes with minimal total weight subject to the constraint that the two MST's have no edges in common. Similarly, a third MST links all the nodes with minimal total weight subject to being orthogonal to the second and the first MSTs. More generally, let a *K*-th MST be the MST orthogonal to the first *K*-1 MSTs. For a given set of nodes, a *K*MST is then the graph defined by all of the edges of the first *K* MST's, each orthogonal to the other in the manner just described.

If we wish to choose subsets of variables which preserve structure among *all* the $n(n-1)/2$ distances, then a feasible choice of criterion is the measure

$$\delta = \sum_{i<j} |d_{ij} - d^{*}_{ij}|. \tag{6.8}$$

Note that δ and the canonical correlations based measures (γ-measures) have the symmetric property in common. Suppose, on the other hand, that we wish to choose subsets which preserve and possibly enhance 'local' structure, i.e., structure among short distances without (or with little) regard for the larger distances. Then a natural choice would be a criterion, δ^{*} (similar to δ) which uses *only* the distance pairs which define either the *K*NN graph or the *K*MST. In other words,

$$\delta^{*} = \sum_{i<j} |\bar{d}_{ij} - \bar{d}^{*}_{ij}|, \tag{6.9}$$

where, $\bar{d}_{ij}$ is the *edge-weight* (*Euclidean distance*) of each of the edges of the *K*NN graph or the *K*MST in the **Y** configuration and $\bar{d}^{*}_{ij}$ is the corresponding distance in the $\tilde{\mathbf{Y}}$ configuration. Note here that, the subgraph (*K*NN or *K*MST) is constructed from **Y** rather than from $\tilde{\mathbf{Y}}$, because it is the $\tilde{\mathbf{Y}}$ configuration that we wish to make as similar as possible to the **Y** configuration and not vice versa. Hence, unlike δ and the *γ-measures*, δ^{*} is *non-symmetric*.

Our preliminary investigation not only found, that both *KNN* graph and *KMST* approaches have similar behaviour with respect to choosing subsets of variables for PCA, but also showed that the computations involved in constructing a *KNN* graph are less intensive than the construction of a *KMST*. Hence, we have chosen the *KNN* approach for the rest of our investigation. The number of edges in a *KNN* graph increases as K increases, thus larger the value of K more the measure δ^* would behave like δ. Since the aim with δ^* is to choose subsets for which *local structure* (eg. group structure) is preserved, it seems appropriate to set K to a much smaller number than the total number of interpoint distance pairs. In our current study we have chosen $K = 1$ as it is the smallest possible choice.

So far, we have defined and explained five criteria, namely M^2, $\hat{\gamma}_5$, $\hat{\gamma}_6$, δ and δ^*, for selecting variables in Principal Component Analysis. These criteria are assessed and compared in a Monte Carlo fashion in Chapter 7, and by means of real data in Chapter 8. Following is a remark on some computational aspects for implementing the above criteria into the selection procedure.

6.4 Computational aspects

In principle, the 'best' subset(s) can be selected by computing the criterion of interest for all possible subsets of variables, and choosing the one(s) with *optimal* criterion value (i.e. *minimum* for M^2, δ and δ^*; and *maximum* for $\hat{\gamma}_5$ and $\hat{\gamma}_6$). Although this can be done for each desired choice of the subset-size q, in practice this approach may not be computationally feasible. To overcome this problem, however, any of the standard procedures such as *forward selection*, *backward elimination* or *stepwise selection* can easily be implemented. Given the efficient algorithms for obtaining an amended singular value decomposition after the deletion of a single variable from a data set (Bunch and Nielsen (1978), Bunch *et.al.* (1978)), *backward elimination* is computationally easier and faster than the other procedures. Hence, to implement the various criteria, the following *backward elimination* procedure is used:

Step 1: Set $q = p$ (i.e. $\tilde{X} = X$) and for a fixed/chosen k, compute the score matrix **Y**.

Step 2: For each variable omitted from $\tilde{X}$ in turn, compute the score matrix $\tilde{Y}$ and use this matrix together with **Y** to compute the criterion of interest.

Step 3: Delete the variable (say, X_d) whose omission gives rise to the optimal value of the criterion, and denote the resulting matrix by $\tilde{X}_{(d)}$.

Step 4: Set $\tilde{X} = \tilde{X}_{(d)}$, and go to Step 2.

This cycle is continued until the desired subset size is reached (i.e. when $q = k$ at end of Step 3).

The proposed variable selection methods are compared and contrasted among themselves as well as with the M^2-*procrustes* criterion in Chapter 7. This comparative study is based mainly on Monte Carlo simulations, and the behaviours of these methods are compared by means of the performance of the selected variables. Although major conclusions are mainly based on the Monte Carlo simulation study, these methods are also applied to some real data sets in Chapter 8.

Although the backward elimination procedure above is used throughout the thesis, it was decided to compare the Monte Carlo results based on this procedure with those based on the stepwise selection procedure (in Chapter 7). The reason for this is that it was considered important to know whether or not using the stepwise procedure (which is computationally slower) improves the performance of the M^2-procrustes criterion and the various proposed criteria for variable selection in PCA.

CHAPTER 7

MONTE CARLO EVALUATIONS OF THE VARIABLE SELECTION TECHNIQUES

7.1 Simulation study plan

The design for the simulation study we adopt in this investigation makes use of models 1 through 4 of Jolliffe (1972), described in section 4.1. Jolliffe constructed these models in such a way that within each model, some variables are linear combinations of others except for a random disturbance and hence are redundant. This way, not only is the 'true' dimensionality of the data is known *prior* to the selection of variables, but also the subsets of variables which satisfactorily define this dimensionality are known.

In Jolliffe's work, choice regarding satisfactory subsets of retained variables was based on maximizing the minimum multiple correlation (say, R_m) between the selected variables and any of the rejected ones; R_m^* being the maximized value of R_m's. It was decided, however, that it would be more appropriate to monitor such a choice according to whether the PC scores Y and $\tilde{Y}$ are obtained from using *covariance* matrices or *correlation* matrices (remembering the fact that PCA is heavily dependent on the type of *variation* matrix used).

Jolliffe's models make use of randomly generated variates Z_l (l = 1, 2, ..., v) which are distributed independently as N(0,1). Being linear combinations of these *iid* variates, the variables X's in the models are distributed as $N(0,\sigma_j^2)$, where σ_j^2 is the variance of X_j (j = 1, 2, ..., p). The relationships between the constructed variables, X_j and the *iid* variates Z_l are shown in Table 4.1. In our investigation, ranking of satisfactory subsets of retained variables when Y and $\tilde{Y}$ are obtained from using the covariance matrices, is based on maximizing the percentage of variance explained by the selected subsets. When the score matrices Y and $\tilde{Y}$

come from correlation matrices however, we adopt Jolliffe's scheme of ranking subsets which is based on maximizing R_m.

Following is an illustration of our scheme of ranking of subsets obtained from using covariance matrices. Consider Jolliffe's model 2. In this model, X_4 equals X_1 plus a random disturbance or alternatively, X_1 equals X_4 plus a (different) random disturbance. Thus, X_1 or X_4 should be discarded and this implies that subsets for which both X_1 and X_4 occur, are actually *bad* subsets. Similarly, X_5 and X_6 are both X_2 plus random disturbances, hence subsets for which at least two variables from the group $\{X_2, X_5, X_6\}$ occur are bad subsets. Clearly, X_1 and X_4 seem to be describing a single dimension and the group $\{X_2, X_5, X_6\}$ describe another dimension. The only isolated variable is X_3 and it seems to be describing a third dimension. Hence the dimensionality (k or q) in Jolliffe's model 2 is 3. In our ranking scheme; of the $\begin{pmatrix} p \\ q \end{pmatrix}$ possible subsets ($\begin{pmatrix} 6 \\ 3 \end{pmatrix}$ in this model), only the sensible ones (not termed *bad*) are considered. These subsets are shown in Table 7.1, alongside with the percentages of variance explained by each of them and the rank of each subset. Note that the subset {3,4,6} in Table 7.1 refers to the subset $\{X_3, X_4, X_6\}$. A similar notation is used throughout the rest of the thesis. Note also that the subset with the largest percentage of explained variation becomes the best and ranked number one, the second best subset is ranked number 2 and so on. Hence, for model 2 the subset {3,4,6} becomes the best set of retained variables for principal component studies based on covariance matrices. Subsets from models 1, 3 and 4 are ranked similarly and the results are summarized in Table 7.2.

Table 7.1: *Percentage of variance explained by each subset and the corresponding subset ranks for model 2 of Jolliffe (1972).*

Subset	Variance	%ge of variance explained	Subset rank
{3,4,6}	4.25	54.9	1
{1,3,6}	4.00	51.6	2
{3,4,5}	3.74	48.4	3
{1,3,5}	3.49	45.1	4
{2,3,4}	3.25	42.0	5
{1,2,3}	3.00	38.7	6

Table 7.2: *Subsets of retained variables ranked according to the percentage of variance explained for models 1, 3 and 4 of Jolliffe (1972).*

Model 1		Model 3		Model 4					
Subset	Rank	Subset	Rank	Subset	Rank	Subset	Rank	Subset	Rank
{4,5,6}	1	{4,5,6}	1	{1,3,6,10}	1	{1,3,6,8}	9	{1,3,5,8}	16
{1,5,6}	2	{3,4,5}	2	{1,2,6,10}	2	{1,3,6,7}	10	{1,3,5,7}	17
{2,4,6}	3	{2,4,6}	3	{1,3,5,10}	3	{1,2,6,8}	11	{1,3,4,8}	18
{1,2,6}	4	{2,3,4}	4	{1,3,4,10}	4	{1,2,6,7}	12	{1,3,4,7}	19
{3,4,5}	5	{1,5,6}	5	{1,2,5,10}	5	{1,3,5,9}	12	{1,2,5,8}	20
{1,3,5}	6	{1,3,5}	6	{1,3,6,9}	6	{1,3,4,9}	13	{1,2,5,7}	21
{2,3,4}	7	{1,2,6}	7	{1,2,4,10}	7	{1,2,5,9}	14	{1,2,4,8}	22
{1,2,3}	8	{1,2,3}	8	{1,2,6,9}	8	{1,2,4,9}	15	{1,2,4,7}	23

Next we shall consider Jolliffe's (1972) choice of best subsets (based on 'correlations') for each of models 1 - 4 based on maximizing the minimum multiple correlation R_m. When this ranking scheme was applied to the four models, it was noted that Jolliffe incorrectly ranked some of the subsets for models 2 and 3. We may justify this finding as follows: Let σ_{ij} and ρ_{ij} be the population variance-covariance and correlation coefficients, respectively between X_i and X_j; and consider for example, model 2. Jolliffe correctly determined the values of ρ_{14}, ρ_{25} and ρ_{26} which are 0.894, 0.819 and 0.707, respectively. However, his reported value 0.579 for

ρ_{56} corresponding to this model is incorrect. To verify this fact, let the symbol $\mathbb{E}$ denote the expected value, σ_i the variance of X_i, $\sigma(Z_i)$ the variance of Z_i and $\sigma(Z_i,Z_j)$ denote the covariance between Z_i and Z_j. Then

$$\begin{aligned}
\sigma_{56} &= \mathbb{E}(X_5X_6) - \mathbb{E}(X_5)\mathbb{E}(X_6) \\
&= \mathbb{E}\left\{(Z_2+0.7Z_6)(Z_2+Z_6)\right\} - \mathbb{E}(Z_2+0.7Z_6)\mathbb{E}(Z_2+Z_6) \\
&= \left\{\mathbb{E}(Z_2^2)-(\mathbb{E}(Z_2))^2\right\} + 0.7\left\{\mathbb{E}(Z_6^2)-(\mathbb{E}(Z_6))^2\right\} \\
&\quad + 1.7\left\{\mathbb{E}(Z_2Z_6)-\mathbb{E}(Z_2)\mathbb{E}(Z_6)\right\} \\
&= \sigma(Z_2) + 0.7\sigma(Z_6) + 1.7\sigma(Z_2,Z_6); \qquad (7.1)
\end{aligned}$$

$$\begin{aligned}
\sigma_5 &= \sigma(Z_2+0.7Z_6) \\
&= \sigma(Z_2) + (0.7)^2\sigma(Z_6) + 2(0.7)\sigma(Z_2,Z_6); \qquad (7.2)
\end{aligned}$$

and $$\sigma_6 = \sigma(Z_2+Z_6) = \sigma(Z_2) + \sigma(Z_6) + 2\sigma(Z_2,Z_6). \qquad (7.3)$$

Since the *iid* variates Z_i are distributed as $N(0,1)$, $\sigma_{56} = 1.7$, $\sigma_5 = 1.49$ and $\sigma_6 = 2.0$, yielding $\rho_{56} = 0.985$. Jolliffe's *subset-classification* results in which he ranked the subsets as *best*, *good*, *moderate* or *bad* for model 2, are partially dependent on the value of ρ_{56}. Hence, in his Table 3, part of the classification for this model is incorrect. The analytical results of this ranking scheme were checked by means of a computer program written in GAUSS (1988) to calculate the multiple correlation between the q retained variables and each of the $p-q$ rejected variables for all $\binom{p}{q}$ possible subsets. The data sets used were constructed in accordance with Jolliffe's model 1 through 4 from randomly generated independent variables distributed as N(0,1). To ensure *sample normality*, etc., large samples were generated (size ≥ 10000). Subsets were then ranked in accordance with the extent to which they maximized the minimum multiple correlation R_m between the q selected variables and any of the $p-q$ discarded variables. The results were

Table 7.3 *Corrected ranking of subsets in Jolliffe's models 1 - 4.*

Type of subset	Model 1 *(k=q=3)*	Model 2 *(k=q=3)*	Model 3 *(k=q=3)*	Model 4 *(k=q=4)*

Best (BT)
$(R_m = R_m^*)$
	Any subset containing one variable from each of the following groups: {1,4},{2,6}, {3,6}	{1,3,5}
		{3,4,5}
			{1,2,3}
			{1,2,6}
			{1,3,5}
			{1,5,6}
			{2,3,4}
			{2,4,6}
				Any subset containing one variable from each of the following groups: {1},{2,3},{4,5,6} {7,8,9,10}

Good (GD)
$(0.7R_m^* \leq R_m < R_m^*)$
	——	{1,2,3}
		{1,3,6}
		{2,3,4}
		{3,4,6}
			{3,4,5}
			{4,5,6}
				——

Moderate (MD)
$(0.5R_m^* \leq R_m < 0.7R_m^*)$
	——	——	{1,3,4}
			{1,4,6}
				——

Bad (BD) $(R_m < 0.5R_m^*)$	All other subsets			

consistent with Jolliffe's choices, except for models 2 and 3 in which the corrected procedure ranked some of the subsets differently. Table 7.3 summarizes the correct(ed) choice of the subsets ranked as *best*, *good*, *moderate* and *bad* for all four models. For model 2 in Jolliffe's classification, the subsets {1,2,3} and {2,3,4} were classified as *best*, while the subsets {1,3,5}, {1,3,6}, {3,4,5} and {3,4,6} were classified as *good*, and all other subsets were classified as *bad*. For model 3, the subsets {1,2,3} and {1,2,6} were classified as *best*, the subsets {1,3,5}, {1,5,6}, {2,3,4}, {2,4,5}, {3,4,5} and {4,5,6} were classified as *good*, the subsets {1,3,4} and {1,4,6} were classified as *moderate*, and all other subsets were classified as *bad*.

In order to conduct our Monte Carlo study, large samples (*n* = 100), moderate sized samples (*n* = 50) and small samples (*n* = 25) are generated in accordance with each of Jolliffe's models 1 - 4. The aim is to compare all five *variable selection methods* among themselves and to see whether the size of the samples has any significant influence on the behaviour of these methods. For each method, 100 replications / data sets are generated for each choice of the sample size. The samples are then subjected to each of the five criteria M^2, $\hat{\gamma}_5$, $\hat{\gamma}_6$, δ and δ^* for variable selection; and the results are presented below. The matrix language programming software GAUSS (1988) is used throughout this Monte Carlo study.

7.2 Monte Carlo results

The results of applying the various criteria on the simulated data are summarized in Tables 7.4 - 7.13. These tables show the number of times (as %) each method retains various types of subsets of q (= k) variables ranked 1, 2, etc. or *best*, *good*, *moderate or bad* for models 1 - 4. Tables 7.4 - 7.11 show the results obtained from using the *backward elimination* procedure. Among these, Tables 7.4 - 7.7 give the results for which $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$ are obtained from covariance matrices, while Tables 7.8 - 7.11 show the results for which PCA (to obtain $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$) is performed on correlation matrices. Also shown in Tables 7.4 and 7.7 is the computational time (in secs) each method takes to complete a single analysis for a given sample size on the **386SX, 20Mhz** personal computer. This information about computational time has been omitted from Tables 7.5 and 7.6 and Tables 7.8 - 7.11 for reasons similar to those noted in section 4.2. This is that, among the similar models (i.e., models 1 - 3) and for a given sample size, the computational time taken by each method to complete the analysis in one model is exactly the same as the corresponding time in the other models. Furthermore, under the same sampling considerations the computational time taken by each method to complete the analysis when covariance matrices are used to perform PCA is very nearly the same as the corresponding time for PCA using correlation matrices.

A preliminary Monte Carlo study showed that for models 1, 3 and 4 of Jolliffe (1972), the behaviour of the various variable selection criteria is virtually unchanged by performing the selection using the *stepwise selection* procedure instead of the *backward elimination* procedure. However, in model 2, the performance of $\hat{\gamma}_5$, $\hat{\gamma}_6$ and M^2 showed a considerable improvement due to the stepwise selection procedure when correlation matrices were used for PCA in conjunction with these criteria. When covariance matrices were used to perform the PCA, the performance of M^2 was virtually the same as when the backward elimination procedure was used to select the variables, while the performance of the γ-measures was slightly worse. Hence, presented in Tables 7.12 and 7.13 are the results obtained when the stepwise selection procedure is used to select the variables for model 2 *only*. The results in Table 7.12 are based on PCA using covariance matrices, while reported in Table 7.13 are the results based on correlation matrices. Once again, the computational time in Table 7.13 is very nearly the same as the corresponding time reported in Table 7.12. Hence, information regarding the computational time appears *only* in Table 7.12.

7.2.1 *Results based on the backward elimination procedure*

First, we shall consider the results based on backward elimination and we shall begin by discussing those results for which *covariance matrices* are used to perform PCA. For model 1 (see Table 7.4) with large samples, the criteria M^2, $\hat{\gamma}_5$ and $\hat{\gamma}_6$ behave equivalently to each other, while not only δ seems superior to δ^*

Table 7.4 *Subsets retained by the various methods for model 1 when covariance matrices are used to perform PCA and the backward elimination procedure is used to select variables.*

	n = 100		n = 50		n = 25	
Method	Subset (Rank)	%ge	Subset (Rank)	%ge	Subset (Rank)	%ge
M^2	{4,5,6} (1) {1,5,6} (2)	99 1	{4,5,6} (1) {1,5,6} (2)	96 4	{4,5,6} (1) {1,5,6} (2) {2,4,6} (3)	77 15 8
	Comp. time (sec.)	6.79	4.59		3.47	
$\hat{\gamma}_5$	{4,5,6} (1) {1,5,6} (2)	99 1	{4,5,6} (1) {1,5,6} (2)	90 10	{4,5,6} (1) {1,5,6} (2) {2,4,6} (3) {1,2,6} (4) {3,4,5} (5) {3,5,6} (BD)	70 17 6 1 5 1
	Comp. time (sec.)	7.11	4.84		3.66	
$\hat{\gamma}_6$	{4,5,6} (1) {1,5,6} (2)	99 1	{4,5,6} (1) {1,5,6} (2)	90 10	{4,5,6} (1) {1,5,6} (2) {2,4,6} (3) {1,2,6} (4) {3,4,5} (5)	79 13 6 1 1
	Comp. time (sec.)	7.07	4.80		3.63	
δ	{4,5,6} (1) {1,5,6} (2) {2,4,6} (3)	98 1 1	{4,5,6} (1) {1,5,6} (2) {1,2,6} (4)	93 6 1	{4,5,6} (1) {1,5,6} (2) {2,4,6} (3) {1,2,6} (4) {3,4,5} (5)	76 11 10 2 1
	Comp. time (sec.)	61.85	16.49		6.63	
δ^*	{4,5,6} (1) {1,5,6} (2) {2,4,6} (3)	91 8 1	{4,5,6} (1) {1,5,6} (2) {2,4,6} (3) {1,2,6} (4)	73 16 9 2	{4,5,6} (1) {1,5,6} (2) {2,4,6} (3) {1,2,6} (4) {3,4,5} (5) {1,3,5} (6)	58 17 13 2 5 5
	Comp. time (sec.)	65.50	16.94		6.66	

Table 7.5 *Subsets retained by the various methods for model 2 when covariance matrices are used to perform PCA and the backward elimination procedure is used to select variables.*

	n = 100		n = 50		n = 25	
Method	Subset (Rank)	%ge	Subset (Rank)	%ge	Subset (Rank)	%ge
M^2	{3,4,6} (1)	100	{3,4,6} (1) {1,3,6} (2)	94 6	{3,4,6} (1) {1,3,6} (2) {2,3,4} (5) {1,2,6} (BD) {2,4,6} (BD) {4,5,6} (BD)	77 8 1 1 10 3
$\hat{\gamma}_5$	{3,4,6} (1)	100	{3,4,6} (1) {1,3,6} (2)	90 10	{3,4,6} (1) {1,3,6} (2) {2,3,4} (5) {1,2,3} (6) {1,2,6} (BD) {2,4,6} (BD)	76 14 7 1 1 1
$\hat{\gamma}_6$	{3,4,6} (1)	100	{3,4,6} (1) {1,3,6} (2)	91 9	{3,4,6} (1) {1,3,6} (2) {2,3,4} (5) {1,2,3} (6) {1,2,6} (BD) {2,4,6} (BD)	76 14 7 1 1 1
δ	{3,4,6} (1) {1,3,6} (2)	99 1	{3,4,6} (1) {1,3,6} (2)	90 10	{3,4,6} (1) {1,3,6} (2) {2,3,4} (5) {1,2,3} (6) {1,2,6} (BD) {2,4,6} (BD)	81 11 4 2 1 1
δ^*	{3,4,6} (1) {1,3,6} (2)	95 5	{3,4,6} (1) {1,3,6} (2)	77 23	{3,4,6} (1) {1,3,6} (2) {2,3,4} (5) {1,2,3} (6) {1,2,6} (BD) {2,4,6} (BD)	69 24 3 2 1 1

but also its behaviour is almost as good as the other methods. Once again the methods $\hat{\gamma}_5$ and $\hat{\gamma}_6$ behave identically to each other when the number of observations in the sample is moderate. However, M^2 and δ perform better than the $\hat{\gamma}$'s; M^2 being the best. δ^* on the other hand is the poorest method for moderate-sized samples. When samples are of small size, $\hat{\gamma}_6$ becomes the best method for selecting variables. The M^2 and δ methods fall behind $\hat{\gamma}_6$, while $\hat{\gamma}_5$and δ^* behave poorly; δ^* being the worst.

For large and moderate samples in model 2 (Table 7.5), the methods behave almost similarly to their behaviour for model 1. However, a change of behaviour can be seen for small sample cases, where M^2 becomes the worst criteria, selecting *bad* subsets more frequently (14%) than other methods. δ is the most successful candidate in this case, picking the *best* subset 81% of the time with only 2% *bad* selections. Criteria $\hat{\gamma}_5$ and $\hat{\gamma}_6$ behave identically to each other and yet better than δ^*.

For all sample sizes in model 3 (Table 7.6), M^2 seems to be superior to the rest of the selection methods. However, the other criteria may not be as inferior as they appear. To clarify this we shall refer to the construction of variables X_4 and X_5 in Jolliffe's model 3 (also see Table 4.1). These two variables can be regarded as X_2 plus (different) random disturbances. Variable X_5, however, is more closely related to X_2 than to X_4, and has a larger variance than X_2; hence gets included in the subsets chosen by the various methods more often than X_2. Similarly, X_6 is X_3 plus a random disturbance and it has a greater chance of being included in the selected subsets because of its larger variance. Now, since model 3 is *3-dimensional*, to complete the subset of three variables to be retained, we need an extra variable. Using the 'clustering technique' described earlier, the choice is between X_1 and X_4. Since the variable X_4 is X_1 plus a random disturbance and has larger variance than X_1, we would expect it to be included in the retained subsets more often than X_1. This is evident in the simulated samples when using the M^2 criterion. However, since X_4 can be clustered in the same group as X_5 (i.e., perceived as describing the same

Table 7.6 *Subsets retained by the various methods for model 3 when covariance matrices are used to perform PCA and the backward elimination procedure is used to select variables.*

	n = 100		n = 50		n = 25	
Method	Subset (Rank)	%ge	Subset (Rank)	%ge	Subset (Rank)	%ge
M^2	{4,5,6} (1)	100	{4,5,6} (1)	92	{4,5,6} (1)	75
			{3,4,5} (2)	6	{3,4,5} (2)	12
			{2,4,6} (3)	2	{2,4,6} (3)	5
					{2,3,4} (4)	1
					{1,5,6} (5)	6
					{1,3,5} (6)	1
$\hat{\gamma}_5$	{4,5,6} (1)	47	{4,5,6} (1)	58	{4,5,6} (1)	46
	{3,4,5} (2)	3	{3,4,5} (2)	3	{2,4,6} (3)	3
	{2,4,6} (3)	1	{2,4,6} (3)	2	{1,5,6} (5)	34
	{1,5,6} (5)	49	{1,5,6} (5)	31	{1,3,5} (6)	9
			{1,3,5} (6)	4	{1,2,6} (7)	1
			{1,2,6} (7)	2	{3,4,6} (BD)	7
$\hat{\gamma}_6$	{4,5,6} (1)	46	{4,5,6} (1)	51	{4,5,6} (1)	44
	{3,4,5} (2)	3	{3,4,5} (2)	3	{3,4,5} (2)	8
	{2,4,6} (3)	1	{2,4,6} (3)	2	{2,4,6} (3)	3
	{1,5,6} (5)	49	{1,5,6} (5)	38	{1,5,6} (5)	35
	{1,3,5} (6)	1	{1,3,5} (6)	4	{1,3,5} (6)	9
			{1,2,6} (7)	2	{1,2,6} (7)	1
δ	{4,5,6} (1)	52	{4,5,6} (1)	54	{4,5,6} (1)	47
	{3,4,5} (2)	2	{3,4,5} (2)	6	{3,4,5} (2)	9
	{1,5,6} (5)	45	{2,4,6} (3)	2	{2,4,6} (3)	3
	{1,3,5} (6)	1	{1,5,6} (5)	37	{1,5,6} (5)	33
			{1,2,6} (7)	1	{1,3,5} (6)	6
					{1,2,6} (7)	1
					{1,4,6} (BD)	1
δ^*	{4,5,6} (1)	49	{4,5,6} (1)	40	{4,5,6} (1)	30
	{3,4,5} (2)	3	{3,4,5} (2)	13	{3,4,5} (2)	13
	{2,4,6} (3)	1	{2,4,6} (3)	6	{2,4,6} (3)	4
	{1,5,6} (5)	43	{2,3,4} (4)	1	{2,3,4} (4)	3
	{1,3,5} (6)	4	{1,5,6} (5)	30	{1,5,6} (5)	29
			{1,3,5} (6)	7	{1,3,5} (6)	13
			{1,2,6} (7)	2	{1,2,6} (7)	5
			{1,2,3} (8)	1	{1,2,3} (8)	1

dimension), its inclusion in a subset containing X_5 may not be desirable. This gives the subset {1,5,6} the potential of being chosen perhaps as much as {4,5,6} and all the methods except M^2 reflect this behaviour.

Table 7.7 *Subsets retained by the various methods for model 4 when covaraince matrices are used to perform PCA and the backward elimination procedure is used to select the variables.*

	n = 100		n = 50		n = 25	
Method	Subset (Rank)	%ge	Subset (Rank)	%ge	Subset (Rank)	%ge
M^2	{1,3,6,10} (1)	85	{1,3,6,10} (1)	72	{1,3,6,10} (1)	52
	{3,6,9,10} (BD)	15	{3,5,6,10} (BD)	3	{1,2,6,10} (2)	1
			{3,6,9,10} (BD)	25	{3,5,6,10} (BD)	8
					{3,6,9,10} (BD)	39
	Comp. time (sec.)	36.53	22.18		14.91	
$\hat{\gamma}_5$	{1,3,6,10} (1)	99	{1,3,6,10} (1)	93	{1,3,6,10} (1)	78
	{3,5,6,10} (BD)	1	{1,3,6,9} (6)	1	{1,2,6,10} (2)	2
			{3,5,6,10} (BD)	6	{1,3,6,9} (6)	3
					{1,2,6,9} (8)	1
					{2,4,6,10} (BD)	1
					{3,4,6,10} (BD)	2
					{3,5,6,9} (BD)	2
					{3,5,6,10} (BD)	6
					{3,6,9,10} (BD)	5
	Comp. time (sec.)	37.74	23.02		15.76	
$\hat{\gamma}_6$	{1,3,6,10} (1)	99	{1,3,6,10} (1)	93	{1,3,6,10} (1)	74
	{3,5,6,10} (BD)	1	{1,3,6,9} (6)	2	{1,2,6,10} (2)	3
			{3,5,6,10} (BD)	5	{1,3,5,10} (3)	1
					{1,3,6,9} (6)	4
					{1,2,6,9} (8)	1
					{2,4,6,10} (BD)	1
					{3,4,6,10} (BD)	2
					{3,5,6,9} (BD)	3
					{3,5,6,10) (BD)	6
					{3,6,9,10} (BD)	5
	Comp. time (sec.)	37.66	22.95		15.66	
δ	{1,3,6,10} (1)	99	{1,3,6,10} (1)	94	{1,3,6,10} (1)	79
	{3,5,6,10} (BD)	1	{3,5,6,10} (BD)	6	{1,2,6,10} (2)	1
					{1,3,6,9} (6)	3
					{1,2,6,9} (8)	1
					{2,4,6,10} (BD)	1
					{3,4,6,10} (BD)	2
					{3,5,6,9} (BD)	2
					{3,5,6,10} (BD)	6
					{3,6,9,10} (BD)	5
	Comp. time (sec.)	207.30	59.93		24.69	
δ^*	{1,3,6,10} (1)	99	{1,3,6,10} (1)	85	{1,3,6,10} (1)	65
	{3,5,6,10} (BD)	1	{1,2,6,10} (2)	1	{1,2,6,10} (2)	8
			{1,3,6,9} (6)	8	{1,3,5,10} (3)	2
			{3,5,6,10} (BD)	6	{1,3,6,9} (6)	7
					{1,2,4,10} (7)	1
					{1,2,6,9} (8)	1
					{2,5,6,9} (BD)	1
					{2,5,6,10} (BD)	3
					{3,4,6,10} (BD)	2
					{3,5,6,9} (BD)	1
					{3,5,6,10} (BD)	5
					{3,6,9,10} (BD)	4
	Comp. time (sec.)	203.58	58.53		24.30	

In model 4 (Table 7.7), the M^2 criterion behaves worse than the other methods for all sample sizes. It is at its worst when samples are of small size, where it chooses the *bad* subsets with 47% chance. The other methods behave very well and similar to each other for large samples. However, though not as much as with the M^2 criterion, the performance of these methods deteriorates as sample size decreases. δ becomes the best method for selecting variables when samples are of moderate and small size.

Among the results based on the backward elimination procedure, we shall next describe the behaviour of our selection criteria when *correlation* matrices are used for PCA. For model 1 (see Table 7.8), all the methods choose the *best* subsets for all sample sizes. M^2 is more consistent for large and small samples, by selecting the same subset more often, while δ is the most consistent criteria for moderate samples.

In model 2 (Table 7.9) for large and moderate samples, all five methods select *good* subsets though δ is more consistent for moderate samples than the others choosing {1,3,6} 62% of the time. Although for small samples, the M^2 method chooses *bad* subsets more often (2%) than the other methods (1%), it is the most consistent selection criterion.

The performance of all criteria in model 3 (Table 7.10) diminishes progressively as the sample size decreases. The criteria $\hat{\gamma}_5$ and $\hat{\gamma}_6$ behave almost identically to each other for all sample sizes, choosing the *best* subsets with 100% probability for large samples. All the methods, except M^2, perform nearly identically to each other for moderate and small sample cases. For moderate sized samples M^2 chooses the *best* subsets 87% of the time compared to a choice of approximately 94% by the other methods. The comparison is

Table 7.8 *Subsets retained by the various methods for model 1 when correlation matrices are used to perform PCA and the backward elimination procedure is used to select variables.*

	n = 100		n = 50		n = 25	
Method	Subset (Rank)	%ge	Subset (Rank)	%ge	Subset (Rank)	%ge
M^2	{1,2,3} (BT)	9	{1,2,3} (BT)	14	{1,2,3} (BT)	9
	{1,2,6} (BT)	12	{1,2,6} (BT)	12	{1,2,6} (BT)	13
	{1,3,5} (BT)	11	{1,3,5} (BT)	17	{1,3,5} (BT)	6
	{1,5,6} (BT)	13	{1,5,6} (BT)	7	{1,5,6} (BT)	10
	{2,3,4} (BT)	9	{2,3,4} (BT)	13	{2,3,4} (BT)	11
	{2,4,6} (BT)	20	{2,4,6} (BT)	16	{2,4,6} (BT)	11
	{3,4,5} (BT)	16	{3,4,5} (BT)	11	{3,4,5} (BT)	16
	{4,5,6} (BT)	10	{4,5,6} (BT)	10	{4,5,6} (BT)	24
$\hat{\gamma}_5$	{1,2,3} (BT)	9	{1,2,3} (BT)	17	{1,2,3} (BT)	13
	{1,2,6} (BT)	13	{1,2,6} (BT)	14	{1,2,6} (BT)	11
	{1,3,5} (BT)	16	{1,3,5} (BT)	13	{1,3,5} (BT)	8
	{1,5,6} (BT)	8	{1,5,6} (BT)	6	{1,5,6} (BT)	11
	{2,3,4} (BT)	10	{2,3,4} (BT)	14	{2,3,4} (BT)	11
	{2,4,6} (BT)	18	{2,4,6} (BT)	16	{2,4,6} (BT)	14
	{3,4,5} (BT)	13	{3,4,5} (BT)	10	{3,4,5} (BT)	18
	{4,5,6} (BT)	13	{4,5,6} (BT)	10	{4,5,6} (BT)	14
$\hat{\gamma}_6$	{1,2,3} (BT)	10	{1,2,3} (BT)	17	{1,2,3} (BT)	13
	{1,2,6} (BT)	12	{1,2,6} (BT)	14	{1,2,6} (BT)	10
	{1,3,5} (BT)	15	{1,3,5} (BT)	13	{1,3,5} (BT)	8
	{1,5,6} (BT)	9	{1,5,6} (BT)	5	{1,5,6} (BT)	11
	{2,3,4} (BT)	10	{2,3,4} (BT)	14	{2,3,4} (BT)	10
	{2,4,6} (BT)	18	{2,4,6} (BT)	16	{2,4,6} (BT)	17
	{3,4,5} (BT)	13	{3,4,5} (BT)	11	{3,4,5} (BT)	17
	{4,5,6} (BT)	13	{4,5,6} (BT)	10	{4,5,6} (BT)	14
δ	{1,2,3} (BT)	17	{1,2,3} (BT)	19	{1,2,3} (BT)	8
	{1,2,6} (BT)	7	{1,2,6} (BT)	13	{1,2,6} (BT)	10
	{1,3,5} (BT)	10	{1,3,5} (BT)	12	{1,3,5} (BT)	13
	{1,5,6} (BT)	12	{1,5,6} (BT)	8	{1,5,6} (BT)	9
	{2,3,4} (BT)	12	{2,3,4} (BT)	16	{2,3,4} (BT)	10
	{2,4,6} (BT)	16	{2,4,6} (BT)	11	{2,4,6} (BT)	10
	{3,4,5} (BT)	13	{3,4,5} (BT)	14	{3,4,5} (BT)	22
	{4,5,6} (BT)	13	{4,5,6} (BT)	7	{4,5,6} (BT)	18
δ^*	{1,2,3} (BT)	7	{1,2,3} (BT)	12	{1,2,3} (BT)	8
	{1,2,6} (BT)	17	{1,2,6} (BT)	12	{1,2,6} (BT)	13
	{1,3,5} (BT)	13	{1,3,5} (BT)	14	{1,3,5} (BT)	13
	{1,5,6} (BT)	15	{1,5,6} (BT)	18	{1,5,6} (BT)	14
	{2,3,4} (BT)	9	{2,3,4} (BT)	15	{2,3,4} (BT)	9
	{2,4,6} (BT)	16	{2,4,6} (BT)	8	{2,4,6} (BT)	17
	{3,4,5} (BT)	16	{3,4,5} (BT)	10	{3,4,5} (BT)	9
	{4,5,6} (BT)	7	{4,5,6} (BT)	11	{4,5,6} (BT)	17

Table 7.9 *Subsets retained by the various methods for model 2 when correlation matrices are used to perform PCA and the backward elimination procedure is used to select variables.*

	n = 100		n = 50		n = 25	
Method	Subset (Rank)	%ge	Subset (Rank)	%ge	Subset (Rank)	%ge
M^2	{1,3,6} (GD)	50	{1,3,6} (GD)	55	{1,2,3} (GD)	1
	{3,4,6} (GD)	50	{3,4,6} (GD)	45	{1,3,6} (GD)	48
					{3,4,6} (GD)	49
					{1,2,6} (BD)	1
					{2,4,6} (BD)	1
$\hat{\gamma}_5$	{1,3,6} (GD)	51	{1,3,6} (GD)	55	{1,2,3} (GD)	4
	{3,4,6} (GD)	49	{3,4,6} (GD)	45	{1,3,6} (GD)	46
					{2,3,4} (GD)	6
					{3,4,6} (GD)	43
					{2,4,6} (BD)	1
$\hat{\gamma}_6$	{1,3,6} (GD)	51	{1,3,6} (GD)	55	{1,2,3} (GD)	4
	{3,4,6} (GD)	49	{3,4,6} (GD)	45	{1,3,6} (GD)	45
					{2,3,4} (GD)	6
					{3,4,6} (GD)	44
					{2,4,6} (BD)	1
δ	{1,3,6} (GD)	49	{1,3,6} (GD)	62	{1,2,3} (GD)	7
	{3,4,6} (GD)	51	{3,4,6} (GD)	38	{1,3,6} (GD)	44
					{2,3,4} (GD)	3
					{3,4,6} (GD)	45
					{1,2,6} (BD)	1
δ^*	{1,3,6} (GD)	51	{1,2,3} (GD)	3	{1,2,3} (GD)	8
	{3,4,6} (GD)	49	{1,3,6} (GD)	48	{1,3,6} (GD)	37
			{2,3,4} (GD)	1	{2,3,4} (GD)	9
			{3,4,6} (GD)	48	{3,4,6} (GD)	45
					{1,2,6} (BD)	1

worse for M^2 in small sample situation where it selects the *best* subsets only for 77% of the replicates while the value for other criteria is roughly 90%.

Table 7.10 *Subsets retained by the various methods for model 3 when correlation matrices are used to perform PCA and the backward elimination procedure is used to select variables.*

	n = 100		n = 50		n = 25	
Method	Subset (Rank)	%ge	Subset (Rank)	%ge	Subset (Rank)	%ge
M^2	{1,2,3} (BT)	46	{1,2,3} (BT)	36	{1,2,3} (BT)	24
	{1,2,6} (BT)	30	{1,2,6} (BT)	37	{1,2,6} (BT)	31
	{1,3,5} (BT)	8	{1,3,5} (BT)	6	{1,3,5} (BT)	7
	{1,5,6} (BT)	6	{1,5,6} (BT)	8	{1,5,6} (BT)	9
	{3,4,5} (GD)	5	{3,4,5} (GD)	9	{2,3,4} (BT)	1
	{4,5,6} (GD)	5	{4,5,6} (GD)	4	{2,4,6} (BT)	5
					{3,4,5} (GD)	11
					{4,5,6} (GD)	11
					{2,4,5} (BD)	1
$\hat{\gamma}_5$	{1,2,3} (BT)	40	{1,2,3} (BT)	31	{1,2,3} (BT)	24
	{1,2,6} (BT)	35	{1,2,6} (BT)	26	{1,2,6} (BT)	23
	{1,3,5} (BT)	14	{1,3,5} (BT)	21	{1,3,5} (BT)	23
	{1,5,6} (BT)	11	{1,5,6} (BT)	19	{1,5,6} (BT)	17
			{3,4,5} (GD)	1	{2,3,4} (BT)	1
			{4,5,6} (GD)	2	{2,4,6} (BT)	2
					{3,4,5} (GD)	4
					{4,5,6} (GD)	3
					{1,3,4} (MD)	1
					{1,4,6} (MD)	2
$\hat{\gamma}_6$	{1,2,3} (BT)	40	{1,2,3} (BT)	31	{1,2,3} (BT)	23
	{1,2,6} (BT)	36	{1,2,6} (BT)	26	{1,2,6} (BT)	23
	{1,3,5} (BT)	14	{1,3,5} (BT)	21	{1,3,5} (BT)	24
	{1,5,6} (BT)	10	{1,5,6} (BT)	19	{1,5,6} (BT)	17
			{3,4,5} (GD)	1	{2,3,4} (BT)	1
			{4,5,6} (GD)	2	{2,4,6} (BT)	2
					{3,4,5} (GD)	4
					{4,5,6} (GD)	3
					{1,3,4} (MD)	2
					{1,4,6} (MD)	1
δ	{1,2,3} (BT)	40	{1,2,3} (BT)	35	{1,2,3} (BT)	16
	{1,2,6} (BT)	31	{1,2,6} (BT)	23	{1,2,6} (BT)	21
	{1,3,5} (BT)	16	{1,3,5} (BT)	22	{1,3,5} (BT)	26
	{1,5,6} (BT)	12	{1,5,6} (BT)	16	{1,5,6} (BT)	28
	{3,4,4} (GD)	1	{3,4,5} (GD)	1	{2,3,4} (BT)	2
			{4,5,6} (GD)	3	{2,4,6} (BT)	1
					{3,4,5} (GD)	3
					{4,5,6} (GD)	2
					{1,3,4} (MD)	1
δ^*	{1,2,3} (BT)	25	{1,2,3} (BT)	25	{1,2,3} (BT)	7
	{1,2,6} (BT)	31	{1,2,6} (BT)	24	{1,2,6} (BT)	23
	{1,3,5} (BT)	15	{1,3,5} (BT)	28	{1,3,5} (BT)	25
	{1,5,6} (BT)	27	{1,5,6} (BT)	15	{1,5,6} (BT)	22
	{3,4,5} (GD)	1	{2,3,4} (BT)	1	{2,3,4} (BT)	5
	{2,4,6} (GD)	1	{2,4,6} (BT)	1	{2,4,6} (BT)	7
			{3,4,5} (GD)	2	{3,4,5} (GD)	3
			{4,5,6} (GD)	4	{4,5,6} (GD)	6
					{1,3,4} (MD)	1
					{1,4,6} (MD)	1

Table 7.11 *Performance of the various methods in model 4 when correlation matrices are used to perform PCA and the backward elimination procedure is used to select variables.*

Method	n = 100		n = 50		n = 25	
	Subset rank	%ge	Subset rank	%ge	Subset rank	%ge
M^2	BT	100	BT	100	BT	99
	GD	0	GD	0	GD	0
	MD	0	MD	0	MD	0
	BD	0	BD	0	BD	1
$\hat{\gamma}_5$	BT	100	BT	100	BT	99
	GD	0	GD	0	GD	0
	MD	0	MD	0	MD	0
	BD	0	BD	0	BD	1
$\hat{\gamma}_6$	BT	100	BT	100	BT	99
	GD	0	GD	0	GD	0
	MD	0	MD	0	MD	0
	BD	0	BD	0	BD	1
δ	BT	100	BT	100	BT	99
	GD	0	GD	0	GD	0
	MD	0	MD	0	MD	0
	BD	0	BD	0	BD	1
δ^*	BT	100	BT	100	BT	100
	GD	0	GD	0	GD	0
	MD	0	MD	0	MD	0
	BD	0	BD	0	BD	0

As it can be seen in Table 7.3, the number of subsets falling in each category (*best, good* etc.) is large for model 4 compared to other models. In our simulations for model 4, each selection method chooses a large number of different subsets, though they fall in the same category (e.g., *best*) almost all the time. We would therefore require a massive table to present all these subsets. Hence, the results for the performance of the methods with respect to this model are presented in a more summarized form (in Table 7.11) than for the other models. The conclusions in this case are easy to make, where for large and moderate samples all five methods choose the *best* subsets with 100% probability. For small samples, apart from δ^* all the criteria choose *bad* subsets 1% of the time and *best* subsets otherwise. Under these sampling considerations, δ^* always chooses the *best* subsets, hence it becomes slightly better than the rest of the methods.

7.2.2 *Results based on the stepwise selection procedure*

Next, we shall describe the behaviour of our criteria when *stepwise selection* is used as the *sequential* procedure to select variables in PCA. In the backward elimination procedure, initially the system contains all the variables. The criterion of interest is computed for each variable omitted in turn, and the variable whose omission optimizes this criterion is deleted. The second stage of the procedure is similar to the first except that it is performed without the variable deleted at the first stage. The first two stages of the stepwise selection procedure are exactly the same as the corresponding stages of the backward elimination procedure except that from the second stage onwards, each stage of the backward elimination procedure is accompanied by *forward selection.* This allows previously deleted variables to re-enter the system, one at a time, and the variable whose return (addition) optimizes the criterion of interest is retained. The popularity of this stepwise procedure is the realization of the fact that a variable may be redundant in the presence of certain variables, but may be essential once these variables have been deleted from the system. Hence, this procedure is an improved search for *optimal* subsets of variables.

As noted before, the behaviour of these criteria when stepwise selection is used to select variables in models 1, 3 and 4 is almost identical to their behaviour when backward elimination is used to perform the selection. For example, consider the behaviour of the criteria in model 4 for moderate samples when covariance matrices are used to perform PCA. When backward elimination is used to select variables (see Table 7.7) δ^* chooses the subsets {1,3,6,10}, {1,2,6,10}, {1,3,6,9} and {3,5,6,10} with 85%, 1%, 8% and 6% probability, respectively. This criterion chooses the same subsets, respectively with 87%, 1%, 6% and 6% probability when stepwise selection is used to perform the selection. Under the same sampling considerations as above, when backward elimination is used as the sequential selection procedure, δ chooses the subsets {1,3,6,10} and {3,5,6,10} with 94% and 6% probability, respectively. This is quite comparable with its choice of the subsets {1,3,6,10}, {1,3,6,9} and {3,5,6,10} with 93%, 1% and 6% probability, respectively when stepwise selection is used to select variables. The rest of the criteria exhibit exactly the same behaviour for stepwise selection as for backward elimination. On the other hand, the criteria M^2, $\hat{\gamma}_5$ and $\hat{\gamma}_6$ show a considerable change of behaviour in model 2 when stepwise selection is used to select variables instead of backward elimination. Hence, among those results for which stepwise selection is used to choose the variables, we choose to present the results for *only* model 2.

First we shall consider the results for which *covariance* matrices are used to perform PCA. For all sample sizes in model 2 (see Table 7.12) the criterion M^2 shows a behaviour very similar (exactly the same for large samples) to its behaviour when backward elimination is used to select variables (see also Table 7.5). However, this criterion selects *bad* subsets (1%) here for moderate samples, while for small samples *bad* selections occur with 2% less probability than when the backward elimination procedure is used for selection. For all sample sizes, the behaviour of the criteria δ and δ^* is exactly the same for stepwise selection as for backward elimination. Finally, for all sample sizes, the criteria $\hat{\gamma}_5$ and $\hat{\gamma}_6$ perform slightly worse with the stepwise selection procedure than when backward elimination is used to perform the selection of variables. These criteria choose the *third best* subset {3,4,5} (see

Table 7.12 *Subsets retained by the various methods for model 2 when covariance matrices are used to perform PCA and the stepwise selection procedure is used to select variables.*

	n = 100		n = 50		n = 25	
Method	Subset (Rank)	%ge	Subset (Rank)	%ge	Subset (Rank)	%ge
M^2	{3,4,6} (1)	100	{3,4,6} (1) {1,3,6} (2) {3,4,5} (3) {4,5,6} (BD)	93 5 1 1	{3,4,6} (1) {1,3,6} (2) {3,4,5} (3) {1,3,5} (4) {1,2,6} (BD) {2,4,6} (BD) {4,5,6} (BD)	76 6 4 2 1 7 4
	Comp. time (sec.)	12.52	8.10		5.95	
$\hat{\gamma}_5$	{3,4,5} (3)	100	{3,4,5} (3) {1,3,5} (4)	93 7	{3,4,6} (1) {1,3,6} (2) {3,4,5} (3) {1,3,5} (4) {1,2,6} (BD) {4,5,6} (BD)	4 1 78 12 2 3
	Comp. time (sec.)	13.20	8.71		6.57	
$\hat{\gamma}_6$	{3,4,5} (3)	100	{3,4,5} (3) {1,3,5} (4)	92 8	{3,4,6} (1) {1,3,6} (2) {3,4,5} (3) {1,3,5} (4) {1,2,6} (BD) {4,5,6} (BD)	4 1 78 12 2 3
	Comp. time (sec.)	13.12	8.64		6.51	
δ	{3,4,6} (1) {1,3,6} (2)	99 1	{3,4,6} (1) {1,3,6} (2)	90 10	{3,4,6} (1) {1,3,6} (2) {2,3,4} (5) {1,2,3} (6) {1,2,6} (BD) {2,4,6} (BD)	81 11 4 2 1 1
	Comp. time (sec.)	140.11	36.10		13.35	
δ^*	{3,4,6} (1) {1,3,6} (2)	95 5	{3,4,6} (1) {1,3,6} (2)	77 23	{3,4,6} (1) {1,3,6} (2) {2,3,4} (5) {1,2,3} (6) {1,2,6} (BD) {2,4,6} (BD)	69 24 3 2 1 1
	Comp. time (sec.)	138.53	35.31		13.23	

Table 7.1) with the highest probability, compared to the choice of the *best* subset {3,4,6} when used with the backward elimination procedure. Furthermore, for small samples, the probability that these criteria choose *bad* subsets increases slightly (3%). However,

for moderate and small samples the choice of the subset {3,4,5} is more consistent than the choice of {3,4,6} when these criteria are used with backward elimination to select variables. For a small number of observations in the sample, δ remains the best method, picking the best subset 81% of the time and making only 2% bad selections.

Next we shall discuss the results for which *correlation* matrices are used for PCA with the stepwise selection procedure. For large and moderate samples in Table 7.13, the behaviour of δ is similar (exactly the same for large sample cases) to its behaviour when variables are selected using the backward elimination procedure (see also Table 7.9). The criterion δ^*, on the other hand, becomes slightly more consistent; choosing the subsets {1,3,6} and {3,4,6} more frequently for large and moderate samples, respectively. When samples are of small size, both the criteria δ and δ^* behave slightly better with the stepwise selection procedure, making a few *best* selections and no *bad* selections, while with the backward elimination procedure in Table 7.9 these criteria make a few *bad* selections and no *best* selections. These criteria are also more consistent here than when backward elimination is used to perform the selection of variables. On the other hand, the performance of the criteria M^2, $\hat{\gamma}_5$ and $\hat{\gamma}_6$ shows a considerable improvement for all sample sizes. Here, the *best* subsets {1,3,5} and {3,4,5} are chosen by the stepwise selection procedure with high probabilities as opposed to the choice of the subsets {1,3,6} and {3,4,6} in backward elimination. Note that, while these *best* subsets {1,3,5} and {3,4,5} are selected here by the stepwise selection procedure, there are no *best* selections made by the backward elimination procedure. Further note that the criteria $\hat{\gamma}_5$ and $\hat{\gamma}_6$ become slightly more consistent

Table 7.13 *Subsets retained by the various methods for model 2 when correlation matrices are used to perform PCA and the stepwise selection procedure is used to select the variables.*

Method	n = 100		n = 50		n = 25	
	Subset (Rank)	%ge	Subset (Rank)	%ge	Subset (Rank)	%ge
M^2	{1,3,5} (BT)	50	{1,3,5} (BT)	55	{1,3,5} (BT)	48
	{3,4,5} (BT)	50	{3,4,5} (BT)	45	{3,4,5} (BT)	51
					{1,2,6} (BD)	1
$\hat{\gamma}_5$	{1,3,5} (BT)	51	{1,3,5} (BT)	55	{1,3,5} (BT)	54
	{3,4,5} (BT)	49	{3,4,5} (BT)	45	{3,4,5} (BT)	45
					{3,4,6} (GD)	1
$\hat{\gamma}_6$	{1,3,5} (BT)	51	{1,3,5} (BT)	55	{1,3,5} (BT)	55
	{3,4,5} (BT)	49	{3,4,5} (BT)	45	{3,4,5} (BT)	44
					{3,4,6} (GD)	1
δ	{1,3,6} (GD)	49	{1,3,6} (GD)	54	{1,3,5} (BT)	2
	{3,4,6} (GD)	51	{3,4,6} (GD)	46	{1,2,3} (GD)	2
					{1,3,6} (GD)	51
					{2,3,4} (GD)	3
					{3,4,6} (GD)	42
δ^*	{1,3,6} (GD)	52	{1,2,3} (GD)	2	{1,3,5} (BT)	1
	{3,4,6} (GD)	48	{1,3,6} (GD)	47	{1,2,3} (GD)	6
			{2,3,4} (GD)	1	{1,3,6} (GD)	45
			{3,4,6} (GD)	50	{2,3,4} (GD)	8
					{3,4,6} (GD)	40

than M^2 when used with the stepwise selection procedure, choosing the subset {1,3,5} with the highest frequency.

We have noted above that, only the criteria M^2, $\hat{\gamma}_5$ and $\hat{\gamma}_6$ show a 'significant' change in their behaviour when the stepwise selection procedure is used instead of backward elimination, and this change occurs for model 2 *only*. Use of the stepwise selection procedure yields a considerable improvement in the performance of M^2, $\hat{\gamma}_5$ and $\hat{\gamma}_6$ when *correlation* matrices are used to perform the PCA. On the other hand, when *covariance* matrices are used for PCA with the stepwise selection procedure, the behaviour of M^2 remains virtually the same as when the backward elimination procedure is used to select the variables. However, this time the γ-measures mainly choose the subsets which are classified as *best* for PCA using correlation matrices and *third best* for PCA using covariance matrices (see also Tables 7.1 and 7.3). To provide a plausible explanation for this, we shall use an analytical description of model 2. Consider the correlation structure within this model (see Table 4.1) and let ρ_{ij} represent the correlation between the variables X_i and X_j. Then, as noted in the previous section, for this model, $\rho_{14} = 0.894$, $\rho_{25} = 0.819$, $\rho_{26} = 0.707$ and $\rho_{56} = 0.985$ and *all* other $\rho_{ij} = 0$. Suppose that the backward elimination procedure is used to select the variables, and as seen before, the effective dimensionality of the model is *three*. Hence, the aim is to choose a plausible subset of *three* variables to describe the simulated data adequately. Needless to say, *prior* to the deletion of any of the variables, the system contains the entire set {1,2,3,4,5,6}. Among the correlated pairs of variables, X_5 and X_6 form the most correlated pair indicating that one of these variables can be deleted without losing much information. Hence, at the initial stage of the backward elimination procedure, X_5 or X_6 will be deleted. Our variable selection criteria tend to delete X_5 and retain X_6, because X_5 is more closely related than X_6 to X_2, and hence more redundant. Assuming that X_5 is deleted from the system, the original set of variables is reduced to {1,2,3,4,6}. At this stage, among the variables in the reduced set, the pair X_1 and X_4 is the most correlated pair of variables. As a result, either variable is a candidate for deletion. When covariance matrices are used to perform PCA, the criteria tend to retain X_4 more frequently because it has a larger variance. However, when correlation matrices are used for PCA, the criteria tend to retain both variables equally often, though X_1 is retained slightly more frequently. This leads to the subset {2,3,4,6} or {1,2,3,6} if X_1 or X_4 is deleted, respectively. Among the variables in each of these subsets, X_2 and X_6 are more closely related than any other pair. The final deletion, however, will be of X_2 since X_6 has a larger variance than X_2. This gives the subset {3,4,6} or {1,3,6}.

As noted earlier, the first two stages of the stepwise selection procedure are exactly the same as the corresponding stages of backward elimination. For model 2, these stages leave two variables deleted from the system. At this stage, the system contains the variables in the set {2,3,4,6} or the set {1,2,3,6} if X_1 or X_4 has been deleted, respectively. The variables X_5 and X_6 are both X_2 plus (different) random disturbances. This suggests that *two* variables from the set {2,5,6} are redundant. Intuitively, it seems preferable to retain X_5 because it is more closely related to both X_2 and X_6 than the latter variables are to each other. In other words, X_5 can represent both X_2 and X_6 simultaneously. In fact, it can be shown that the *optimal* variable to retain, in the sense of method B1 described in section 5.2.1, is X_5. Hence, the next stage of the stepwise selection procedure involves substitution of X_2 or X_6 in the set {2,3,4,6} or the set {1,2,3,6} by X_5. This proceeds in the manner described below. The variables X_2 and X_6 form the most correlated pair in the sets {2,3,4,6} and {1,2,3,6}, and hence either variable is redundant in the presence of the other and should be deleted. The deletion of X_2 gives rise to the set {3,4,6} or the set {1,3,6}, while the deletion of X_6 yields the set {2,3,4} or the set {1,2,3}. At this stage, through the *forward selection* step of the stepwise procedure, the variable X_5 returns because only one variable which is highly correlated with X_5 (i.e., X_2 or X_6) is still present in the system. This gives rise to one of the sets

{3,4,5,6}, {1,3,5,6}, {2,3,4,5} and {1,2,3,5}. In the first two sets, the variables X_5 and X_6 are highly correlated, suggesting that one of these variables can be deleted without losing much information. For the reason described above, X_5 is retained and X_6 is deleted so that the retained subset of variables is {3,4,5} or {1,3,5}. Similarly, in the last two sets {2,3,4,5} and {1,2,3,5}, X_2 and X_5 are highly correlated. Once again, X_5 is retained and X_2 deleted to give the subset {3,4,5} or {1,3,5}.

Notice that the *analytical stepwise selection* of variables described above is based on intuition and the *correlation structure* of model 2. Furthermore, by using correlation matrices for PCA to compute the criteria M^2, $\hat{\gamma}_5$ and $\hat{\gamma}_6$, the *correlation structure* of this model is analysed. Hence, the choices of variables using these criteria are comparable with the choice according to the former technique. However, the computation of the γ-measures takes into account the correlations between the PCs of the complete data and of the subset data, and hence the intercorrelations among the original variables both when correlation and covariance matrices are used to perform the PCA. Hence, in either case the subset selections made using these criteria are comparable with the choices according to the *analytical stepwise procedure* described above. With respect to the classification of subsets of variables for PCA using covariance matrices this shows a slight deterioration in the performance of the γ-measures. However, it can be argued that, as seen before, the *analytical procedure* attempts to reject the most redundant variables, and since the γ-measures mainly choose the same subsets of variables chosen by the *analytical stepwise procedure* regardless of the type of *variation* matrix used (covariance or correlation), they should gain favour over the rest of the criteria.

On the other hand, the variables in models 1, 3 and 4 are not as highly intercorrelated as the variables in the group {2,5,6} in model 2. For example, in model 1, $\rho_{14} = 0.894$, $\rho_{25} = 0.819$, $\rho_{36} = 0.707$ and *all* other $\rho_{ij} = 0$. Similarly, in model 3, $\rho_{14} = 0.707$, $\rho_{24} = 0.566$, $\rho_{25} = 0.819$, $\rho_{36} = 0.894$, $\rho_{45} = 0.463$ and *all* other $\rho_{ij} = 0$. Hence, at a given stage of the backward elimination procedure, the importance of a particular variable is not heavily dependent on the presence of other variables. Thus, when the backward elimination procedure is used with the selection criteria, essential variables for models 1, 3 and 4 can be identified satisfactorily.

It seems that, when *three* or more variables in the system are highly correlated among each other, subset selection using the criteria M^2, $\hat{\gamma}_5$ and $\hat{\gamma}_6$ in conjunction with correlation matrices yields better results for the stepwise selection procedure than for the backward elimination procedure. On the other hand, when covariance matrices are used to perform the PCA, the behaviour of the M^2 criterion remains virtually unaltered when the stepwise selection procedure is used instead of the backward elimination procedure, while the performance of the criteria $\hat{\gamma}_5$ and $\hat{\gamma}_6$ slightly deteriorates. However, in cases where the variables are not highly intercorrelated, the performance of *all* the criteria, including δ and δ^*, is virtually the same for the stepwise selection procedure as for the backward elimination procedure.

7.2.3 A remark on computational time

Finally, we compare our selection criteria among themselves as well as with Krzanowski's (1987) *procrustes*-M^2 criterion on the basis of the amount of computational time needed to perform a single analysis under a given set of sampling considerations. First consider the computational time needed to implement the criteria when variable selection is performed using the backward elimination procedure. Let $C_{(t)}$ denote computer time needed to complete a single analysis for criterion C. Then using Table 7.4, the computer times for the various criteria when $p = 6$ can be ordered as

$$M^2_{(t)} < \hat{\gamma}_{6(t)} < \hat{\gamma}_{5(t)} < \delta_{(t)} < \delta^*_{(t)}, \tag{7.4}$$

for all sample sizes. As noted in section 6.2, the criteria δ and δ^* are similar except that δ uses all the possible distance pairs of the configurations Y and $\tilde{Y}$, while δ^* utilizes *only* the distance pairs defined by the KNN graph based on the set Y. Intuitively, it seems reasonable to expect $\delta^*_{(t)}$ to be greater than $\delta_{(t)}$ because of the 'extra' computational time required to construct the KNN graph. In fact, this is reflected when $p = 6$ in Table 7.4. However, only *one* KNN graph is constructed to be used in the entire analysis. On the other hand, as the number of variables p increases, the number of variables to be deleted also increases. This follows from the fact that the aim of the analysis is *dimensionality reduction*. Consequently, the number of computational operations involving the calculation of the values of δ^* or δ also increases. This, in turn, increases the amount of computational time needed to implement both the criteria δ and δ^*. However, this time increases faster for δ than it does for δ^* because the computation of the values of δ involves more distance pairs. Hence, for a sufficiently large choice of the value of p, the 'extra' time needed to implement δ will exceed the time needed to construct the KNN graph for δ^*. This means that $\delta_{(t)}$ will be greater than $\delta^*_{(t)}$ as opposed to $\delta^*_{(t)}$ being greater than $\delta_{(t)}$ when p is small. This is evident in Table 7.7, where p = 10.

Next consider the computational time needed to implement the criteria when stepwise selection is used to choose subsets of variables. Using Table 7.12, the computer times for the various criteria when $p = 6$ can be ordered as

$$M^2_{(t)} < \hat{\gamma}_{6(t)} < \hat{\gamma}_{5(t)} < \delta^*_{(t)} < \delta_{(t)}, \tag{7.5}$$

for all sample sizes. In the stepwise selection procedure, a larger number of computational operations involving the calculation of the values of the various criteria is required than in backward elimination. Hence, more computational time is needed for the stepwise selection procedure and using a similar argument about δ and δ^* as the one above, $\delta_{(t)}$ is greater than $\delta^*_{(t)}$.

The criteria M^2, $\hat{\gamma}_5$ and $\hat{\gamma}_6$ utilize the corresponding individual data points in the configurations $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$ while δ and δ^* make use of the corresponding interpoint distances. These interpoint distances take more time to compute. Furthermore, while there are only n points in each of the configurations $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$, there are $1/2n(n-1)$ interpoint distances. Hence, in both the backward elimination and the stepwise selection procedures, the computational time needed to implement each of the criteria M^2, $\hat{\gamma}_5$ and $\hat{\gamma}_6$ is less than the time needed for δ and δ^*.

7.3 Conclusions

The results of the Monte Carlo simulation study show, in general, that all five methods perform satisfactorily when samples are of large size. However, the performance deteriorates as the sample size decreases, especially M^2 becoming the worst method for selecting variables in a PCA with small samples. δ becomes the best method in this case, followed very closely by $\hat{\gamma}_5$ and $\hat{\gamma}_6$. In fact, the study shows that, while in some cases $\hat{\gamma}_6$ and $\hat{\gamma}_5$ behave identically, there is a consistent similarity between their behaviour and that of the criterion δ. While in general, the criterion δ^* exhibits the poorest performance in this study, there is reason to believe (from its derivation) that, like M^2, it could successfully identify *structure-bearing* variables for *grouped* data as it attempts to preserve 'local structure'. This however, obviously needs further investigation. It appears that $\hat{\gamma}_5$, $\hat{\gamma}_6$ and δ are the most suitable methods for our aim to retain subsets of variables which will preserve the overall *multivariate structure* of the data in a PCA. Although none of these three criteria is uniformly best, δ seems to be the overall 'best' choice.

Regarding the *optimal sequential* approach in selecting subsets of variables, substitution of the *backward elimination* procedure with *stepwise selection* generally leaves the performance of *all* the criteria virtually unaltered. However, if the data are simulated in such a way that *three* or more variables are highly intercorrelated, use of the stepwise selection procedure shows a considerable improvement in the performance of the criteria M^2, $\hat{\gamma}_5$ and $\hat{\gamma}_6$ when correlation matrices are used to perform the PCA. Hence, in this case, these criteria are better choices for preserving the overall multivariate data structure than δ and δ^*; with $\hat{\gamma}_5$ and $\hat{\gamma}_6$ performing almost equivalently but slightly better than M^2. On the other hand, the performance of the γ-measures slightly deteriorates when covariance matrices are used for PCA, the behaviour of the M^2 criterion is virtually unaltered by using the stepwise elimination procedure.

In terms of the amount of computational time needed to implement the various criteria, M^2, $\hat{\gamma}_5$ and $\hat{\gamma}_6$ seem more practicable than δ and δ^*; M^2 being the fastest. However, δ and δ^* can also be used in practice, particularly for small data sets.

CHAPTER 8

EVALUATION OF THE VARIABLE SELECTION METHODS ON REAL DATA

8.1 Introduction

The aim of this Chapter is to study the behaviours of the proposed criteria ($\hat{\gamma}_5$, $\hat{\gamma}_6$, δ and δ^*) and the M^2-*Procrustes* criterion for selecting variables in PCA using real data sets. In a Monte Carlo simulation study in Chapter 7, the proposed criteria above were considered and compared extensively among each other as well as with the M^2 criterion. The reason for this was that, the currently available methods for selecting variables in PCA are solely based on the *eigen_analysis* of either the covariance or the correlation matrix, and may therefore lead to unsatisfactory subsets. The Monte Carlo comparison of these five selection methods was mainly based on the performances of the subsets they chose. This was possible because the simulated data was constructed in such a way that the constructed variables fell into groups and within which variables were linear combinations of others, with variables from different groups being independent. Therefore, not only the 'true' *dimensionality* of the data and hence the number of variables to be retained were known *prior* to the variable selection, but also the best subsets were known. We argued in Chapter 2, however, that in real practice the 'true' dimensionality of the data is unknown and hence the need for a good estimator of this dimensionality is inevitable. We proposed a technique based on *bootstrapping* for estimating *k*, the number of components in PCA (or dimensionality) and compared it with the cross-validatory scheme of Eastment and Krzanowski (1982). We have seen from our simulation study and from the information available in the literature that, although the latter technique chooses about the right number of components, the tendency is for k to be chosen too small. However, the former technique, based on bootstrapping was shown to be the best choice for estimating the 'true' dimensionality of a given data set.

In real practice, although the dimensionality of the data can be estimated using either our proposed bootstrap technique or the cross-validatory technique of Eastment and Krzanowski (1982) in our investigation, knowledge of the best subsets of retained variables is absent. On the other hand, our aim is to choose the subsets of variables for which the general features of the data from the entire set of variables are 'reproduced' as closely as possible. Hence, in order to assess the performance of the various selection criteria, the *2-dimensional* as well as *3-dimensional* plots (where applicable) of the *principal component scores* from the subset data configuration are compared with the corresponding plots of the complete data configuration. The variable subsets for which these plots are similar can be considered to be the satisfactory subsets. The results corresponding to this assessment of the retained subsets of variables on various data sets considered are presented in section 8.3.1 below. Notice that in this section the *k-dimensional* plots for which the subset size is k explain *all* of the variation in the subset data, hence the axes (i.e., PC1 and PC2 in the case of the *2-dimensional* plots and PC1, PC2 and PC3 in the case of the *3-dimensional* plots) are merely a geometrical rotation of the original subset variables.

Now, using the assessment of the retained variable subsets described above corresponds to the requirement that the discarded variables should be only the redundant ones. There is, however, an alternative approach to identifying subsets of retained variables so that the discarded variables are the most redundant. Recall that in

Multiple Regression with *p* independent (descriptor) variables there is an obvious way of choosing a subset of *k* of these descriptors, and this is to choose the subset which maximizes the *multiple correlation* of the dependent variable with the *k* independent variables (or, equivalently, minimizes the residual mean square). An extension of this to the analysis of interdependent variables, of which *Principal Component Analysis* is a special case, is to retain the set of *k* variables which *maximizes* the *minimum multiple correlation* (say, R_m) between the *k* selected variables and any of the *p-k* discarded ones. Hence, in addition to comparing the plots of the PC scores of the subset data with the corresponding plots from the complete data, we also use this approach of maximizing the minimum multiple correlation to assess the subsets of retained variables. This study is covered in section 8.3.2. Notice that we have used a similar approach in the Monte Carlo study of Chapter 7 to assess the subsets of variables chosen by the various selection criteria for PCA performed using correlation matrices.

Although the performances of our proposed bootstrap technique for determining the dimensionality and the corresponding cross-validatory technique of Eastment and Krzanowski (1982) have been compared extensively in a Monte Carlo fashion in Chapter 4, it was decided to compare them again here using real data. This is achieved by comparing the performances of the various variable selection criteria when used with the choice of the dimensionality according to Eastment and Krzanowski's technique, and the corresponding performances of these criteria when used with the choice according to the proposed bootstrap technique. Three sets of real data were obtained from *two* sources and these are analysed below. Each data set is described separately first, and the results are discussed later.

8.2 Descriptions of data sets

In order to assess the general usefulness of our proposed criteria and that of the M^2-*Procrustes* criterion when used to select variables in PCA, real data which fall into three *categories* are used. Each category consists of a single data set. The *first* category consists of a *small-sized ungrouped* data, henceforth referred to as *data 1*. This data set contains the monthly employment figures from 1948-1981 of 16-19 year old male Americans (see page 392 of Andrews and Herzberg (1985)). Here, the twelve months in a year are treated as variables. The *second* category consists of *small-sized grouped* data, referred to as *data 2* while the *third* consists of *large-sized grouped* data, termed *data 3*. These two categories of data are based on Venezuelan students who were financed under a *British Council* scheme to study *English Language* for a year at colleges in north of *England*. These students were distributed among ten colleges (Oldham, Accrington, Burnley, Lancaster, Cheshire, Reading, Blackpool, Carlett and Lytham) and their progress was monitored by means of tests administered at the start (November 1985), in the middle (February 1986) and at the end (June 1986) of their courses. Altogether, eight subject units in English and Spanish were examined. In this case, the students' scores corresponding to each of these subjects are treated as variables. Furthermore, in our investigation we consider the dates on which the tests were administered as the groupings among the students' scores. The data set *data 2* consists of information on students from the college Carlett only, while *data 3* contains the information from three colleges, namely, Oldham, Cheshire and

Carlett. Descriptions of the variables for *data 1*, *data 2* and *data 3* together with the corresponding correlation matrices can be found in appendix A of this Thesis. The category of *Large-sized ungrouped* data is not used in this investigation due to the difficulty encountered in representing the data using *3-dimensional* and *2-dimensional* plots of principal component scores with labels to identify individual data points. In this case the labels may overlap considerably leading to a totally unreadable plot. In fact, owing to this difficulty, a few observations have been omitted from some of the plots in section 8.3.1.

8.3 Results

8.3.1 Evaluation using 3-D and 2-D plots of PC scores

The results of applying the various criteria on the data sets *data 1*, *data 2* and *data 3* are summarized in Figures 8.1 - 8.14. Each of figures 8.1 - 8.11 consists of two plots of principal component scores, and in each case the first plot (i.e., Figure a) is the *3-dimensional* plot of PC scores for a given set of variables, while the second plot (Figure b) is the corresponding *2-dimensional* plot. Figures 8.12 - 8.14 consist of only the *2-dimensional* plots, and this is because the the corresponding data set has a 'true' *dimensionality* of 2 according to W, the cross-validatory scheme of Eastment and Krzanowski (1982) and $\tilde{W}$, the proposed bootstrap technique. Note that, in order to determine the 'true' dimensionality of the data (*k* or *q*) these techniques (W and $\tilde{W}$) are used throughout the rest of this investigation. For each data set, the number of bootstrap samples used to compute the values of the statistic $\tilde{W}$ is fixed at 1,000. For simplicity, the results presented here are *only* those for which the *backward elimination* procedure is used as the *sequential procedure* to select the variables. Note that, since some of the variables have large variances for all three data sets, it is necessary to standardize all the variates in each case before PCA and variable selection could be performed. Hence, the results in Figures 8.1 - 8.14 are *all* based on using the *correlation matrices* to perform PCA. These figures show the plots of the various data sets on the *first two* and *first three* PCs *before* and *after* variable selection. The aim is to assess, how well the subsets selected by the various methods capture the 'structure' of the complete data. In order to do this, the orientation of each individual data point relative to the orientations of the rest of the points in the space described by the *first two* or *three* PCs of the subset data is compared with the orientation of the same point in the space described by the corresponding PCs of the complete data. If the orientation of the data points in the former space is similar to their orientation in the latter space, then it can be argued that the subset data successfully 'preserves' the multivariate structure present in the complete data. Also shown below each plot is the percentage of variance explained by the plot.

First consider the results based on *data 1* which is *small-sized* and *ungrouped*. For this data set, W suggests that 3 components should be retained for adequate representation of the data (i.e., k = 3) and hence, the optimum number of variables to be retained is also set equal to 3. The results of applying the various variable selection criteria on the basis of this choice of k are summarized in Figures 8.2 - 8.5. On the other hand $\tilde{W}$'s choice for the value of k is 4, suggesting that the first 4 PCs should be used to compute each criterion, and hence 4 of the original set of variables should be retained. We begin by describing the behaviour of the variable selection criteria for the former choice of the value of *k*. The plots in Figures 8.1a and 8.1b respectively show the *3-dimensional* and *2-dimensional* representations of the complete data with *all* 12 variables, while the plots in Figures 8.2a and 8.2b are respectively the *3-dimensional* and *2-dimensional* representations for the subset of variables {3,6,10} selected by the M^2 criterion. Since the plot in Figure 8.2a resembles the one in Figure 8.1a in the manner described above, and similarly the plot in figure 8.2b resembles the one in Figure 8.1b, it can be argued that the M^2 criterion selects a subset that satisfactorily preserves the structure of the complete data. The noticeable discrepancy between the two plots in each case is that the points on the plots from the subset data are more 'compressed' than those from the full set data. The plots in Figures 8.3a and 8.3b which are respectively the *3-dimensional* and *2-dimensional* representations for the subset {2,5,6} selected by $\hat{\gamma}_5$ show that while this criterion 'preserves' the 'original' structure (Figures 8.1a and 8.1b) it seems to have *rotated* the original structure through an angle of 180° followed by a *reflexion*. Notice that, in order to show that data structure clearly the PC1 axis in the plot of Figure 8.1a has been reversed. The criteria $\hat{\gamma}_6$ and δ both chose the subset {2,5,12} whose *3-dimensional* and *2-dimensional* representations are given in Figures 8.4a and 8.4b, respectively. Here, in addition to original structure being preserved, a wave-like pattern of the data points is revealed on the first and fourth quadrants of the *2-dimensional* plot. The sequence of points that defines this wave-like pattern is given by 23, 26, 24, 25, 27, 31, 32, 30, 33, 28, 29, 34. The subset {1,5,11} chosen by δ^* produces a similar plot (Figure 8.5b) except that the wave-like pattern is more prominent. It is apparent from the introductory information on the data set at hand (*data 1*) on page 391 of Andrews and Herzberg (1985) that, unemployment of young persons typically increases in the summer months, when schools are not in session, and falls when schools re-open. This is an annual seasonal pattern. Recalling that the plots above arise from the subsets {2,5,12} and {1,5,11} each containing the summer month May (i.e., variable 5) and two winter months, the wave-like pattern in these plots seems to confirm this seasonal pattern. Hence, the criteria $\hat{\gamma}_6$, δ and δ^* are shown to be excellent choices for detecting such interesting multivariate patterns.

A feature common to *all* the plots in Figures 8.1 - 8.5 is the presence of two *clusters* of observations. This feature shows more clearly in the *3-dimensional* plots. Thus, while there is no *prior* knowledge of the existence of *groupings* among the observations for this data, these plots seem to suggest that at least two groups are present in the data. The *first group* of observations (data points) lies mainly on the *second* and *third quadrants* of each *2-dimensional* plot and consists of the observations 1-22 corresponding to the years 1948-1969, while the *second group* is mostly on the *first* and *fourth quadrants* of each plot and consists of the observations 23-34 from the years 1970-1981. Hence, it seems reasonable to conclude that, the unemployment rate of 16-19 year old males in the *United States of America* (*U.S.A.*) was at one phase during the years 1948-1969 and at another phase during the years 1970-1981. Note that this change of pattern is more prominent in Figures 8.4a and 8.4b, which are respectively the *3-dimensional* and *2-dimensional* representations of the subset chosen by $\hat{\gamma}_6$ and δ, and in Figures 8.5a and 8.5b which result from δ^*'s choice of the 'signal' variables.

Figure 8.1a *Scatter diagram for data 1 plotted on the first three PCs computed from all twelve variables. The diagram accounts for 98.8% of the total variation in the data.*

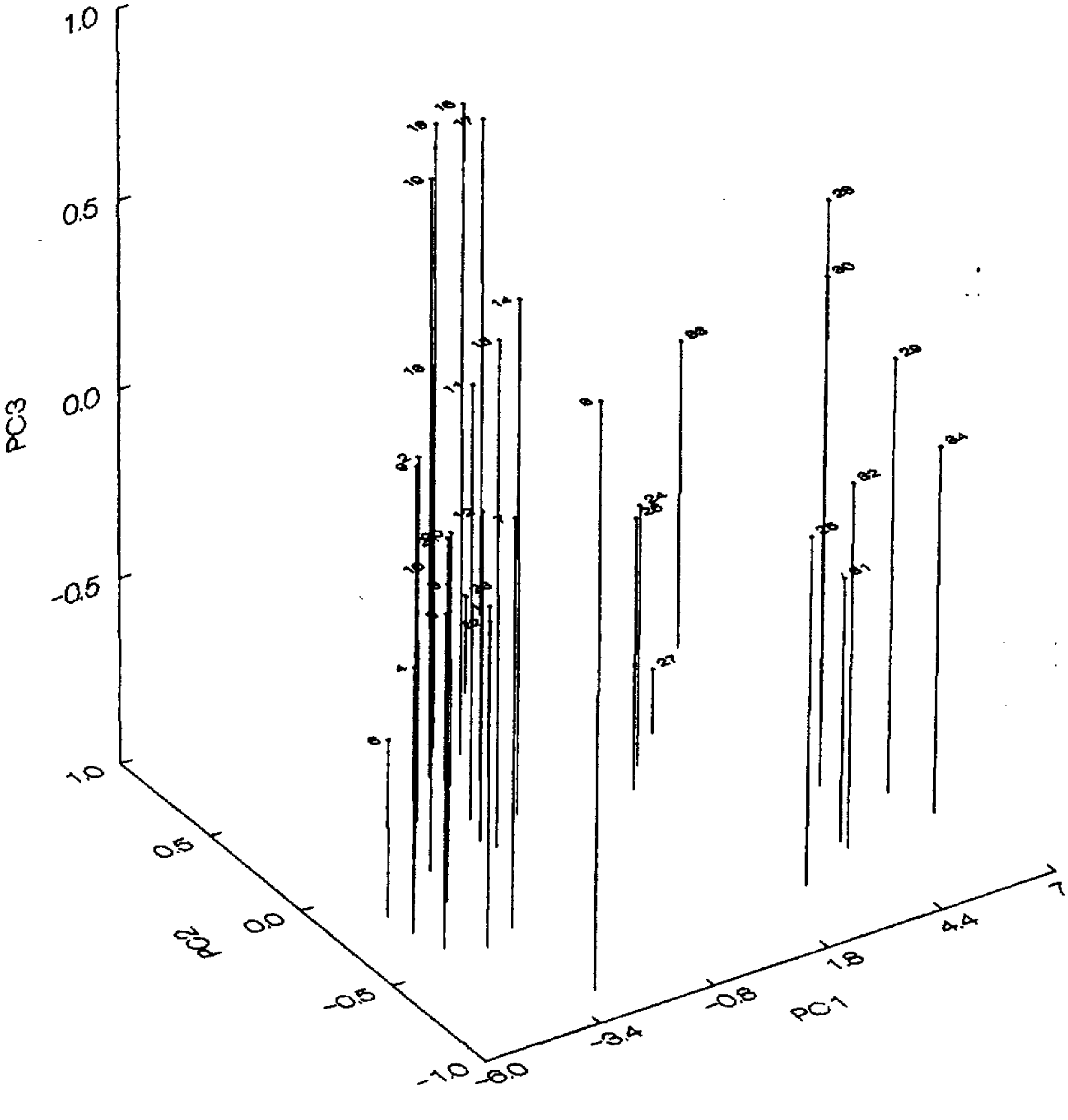


Figure 8.1b *Scatter diagram for data 1 plotted on the first two PCs computed from all twelve variables. The diagram accounts for 97.5% of the total variation in the data.*

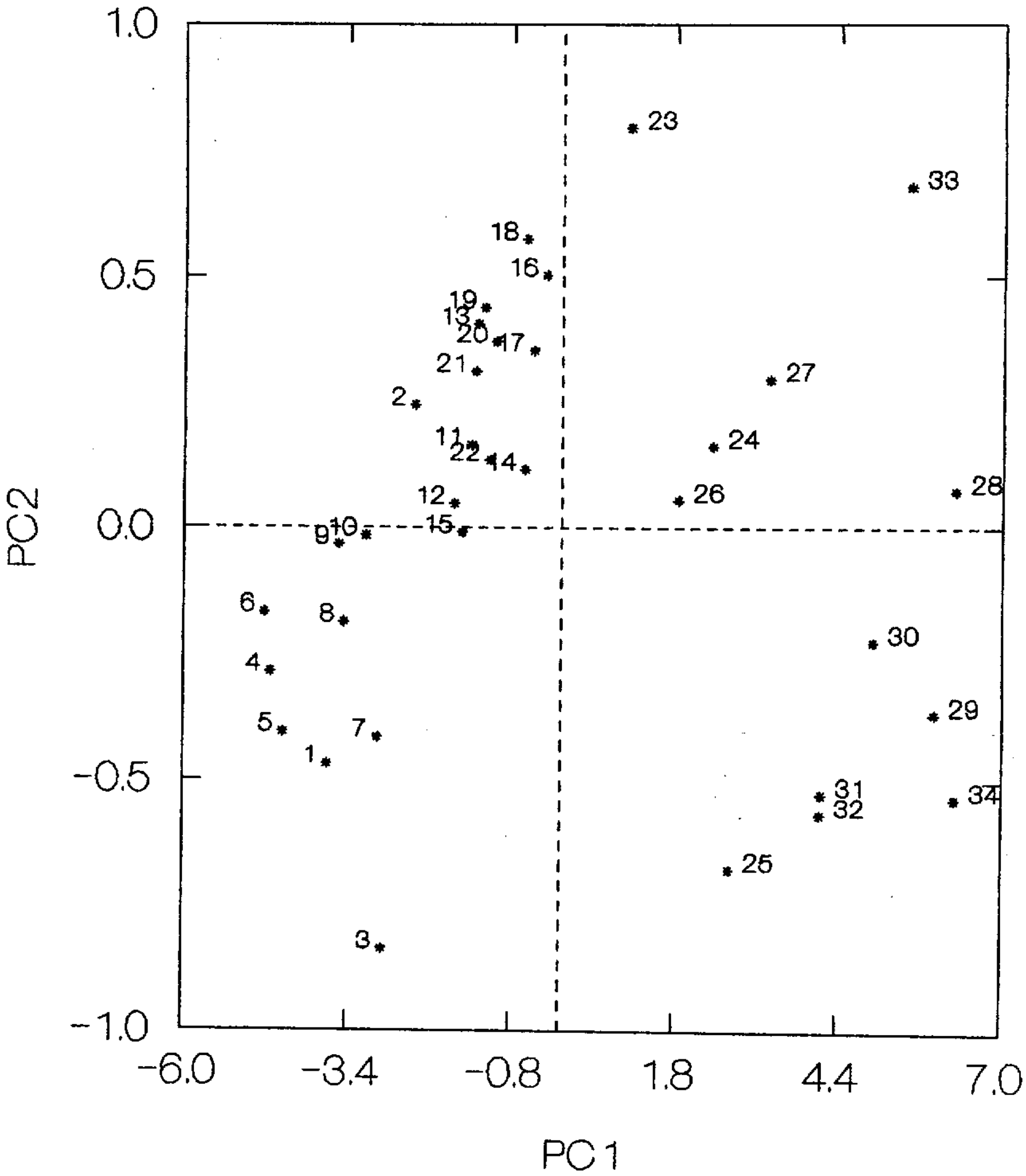


Figure 8.2a *Scatter diagram for data 1 plotted on the first three PCs computed from variables 3,6 and 10 only. The diagram accounts for all of the total variation in the subset data. Note: 5 observations are missing.*

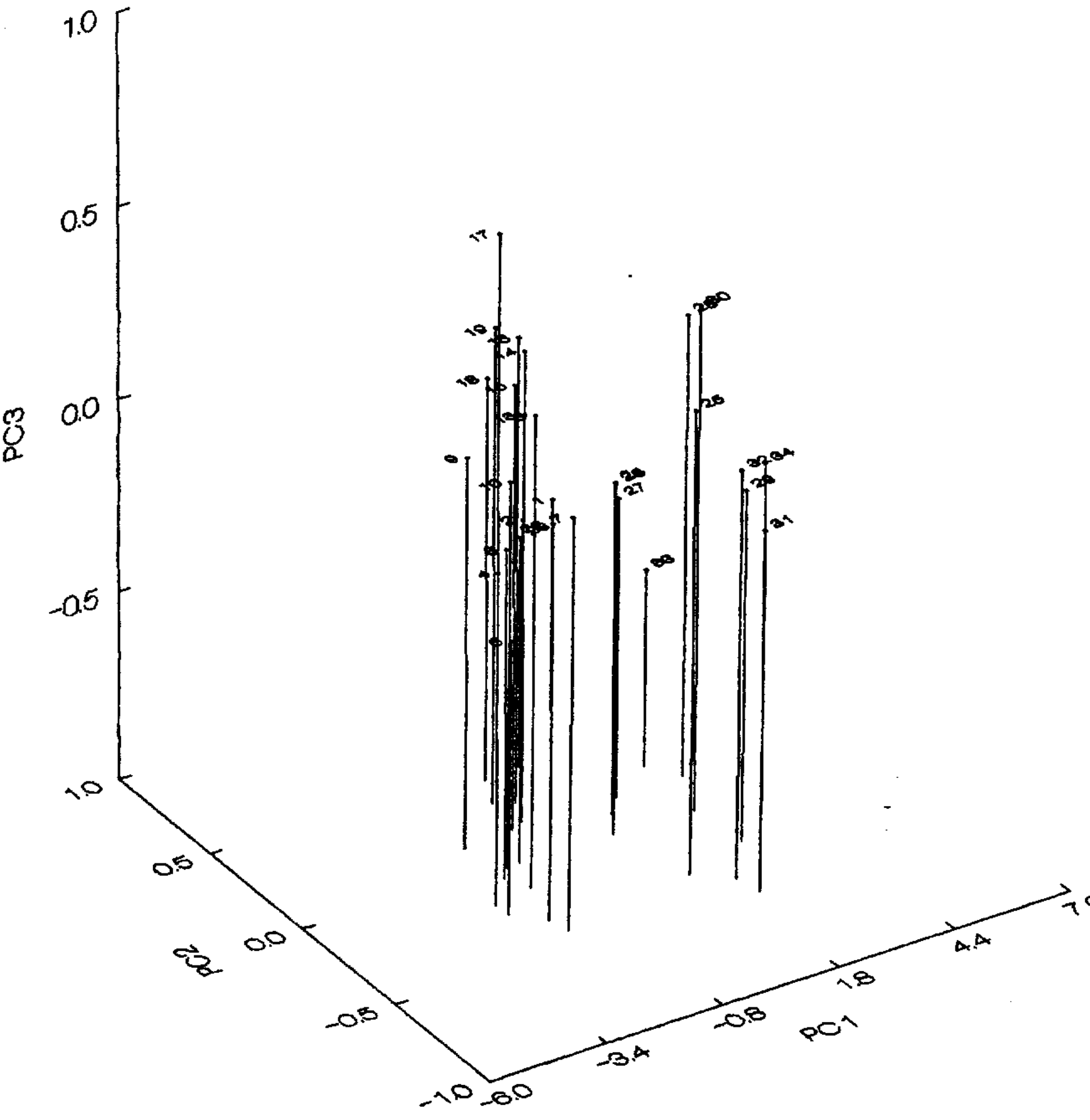


Figure 8.2b *Scatter diagram for data 1 plotted on the first two PCs computed from variables 3,6 and 10 only. The diagram accounts for 98.6% of the total variation in the subset data. Note: 5 observations are missing.*

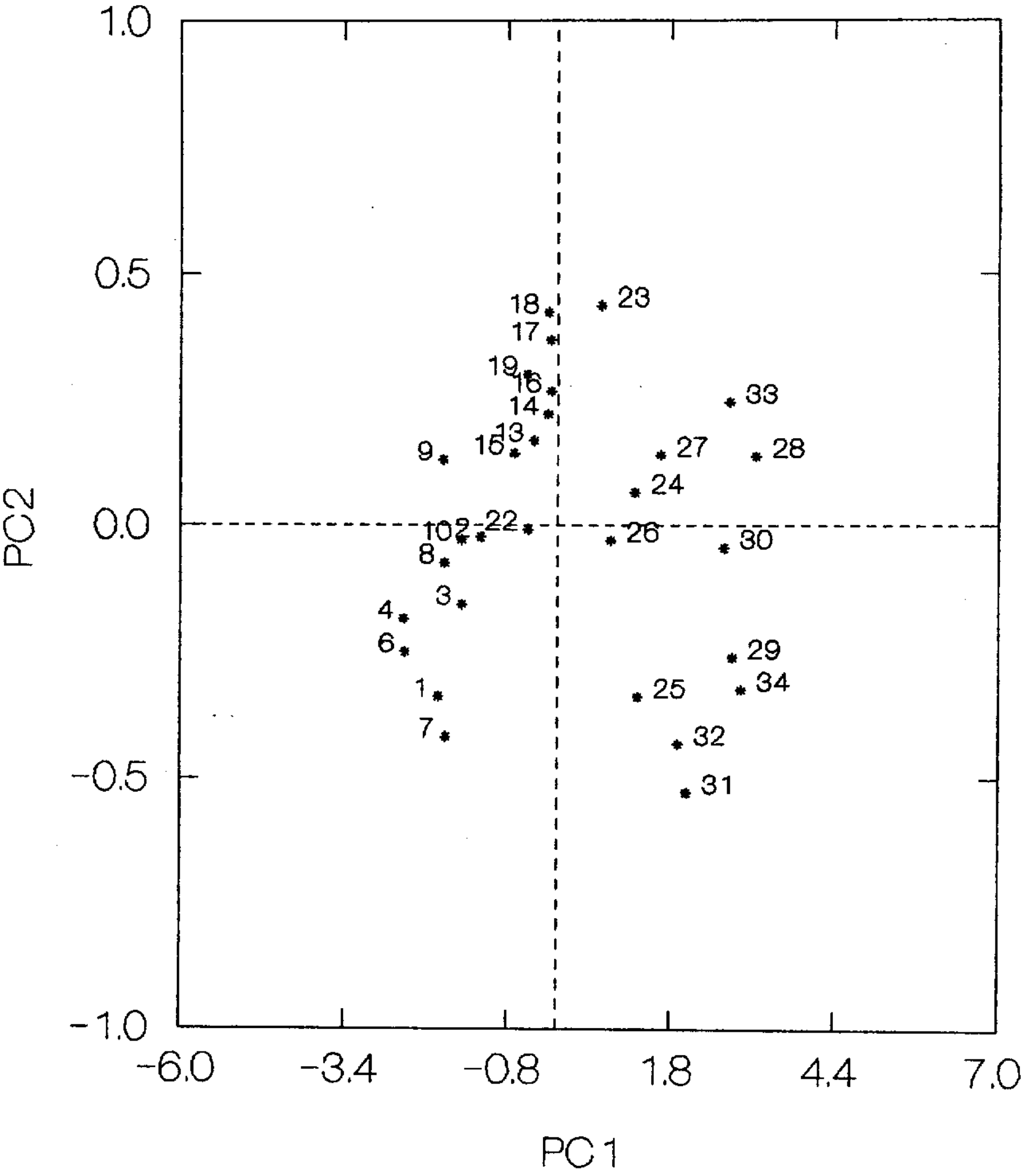


Figure 8.3a *Scatter diagram for data 1 plotted on the first three PCs computed from variables 2,5 and 6 only. The diagram accounts for all of the total variation in the subset data. Note: 4 observations are missing.*

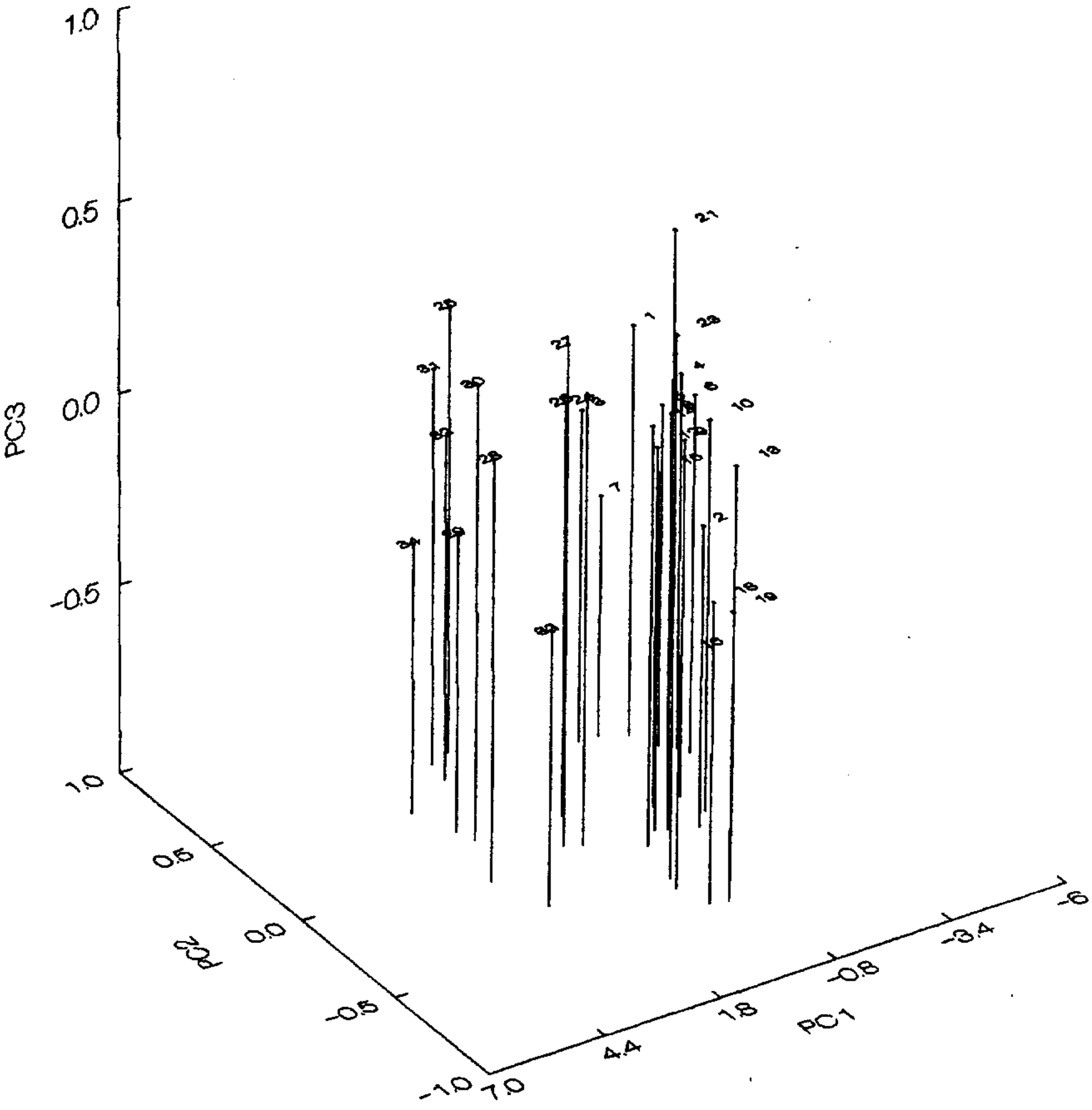


Figure 8.3b *Scatter diagram for data 1 plotted on the first two PCs computed from variables 2,5 and 6 only. The diagram accounts for 98.3 of the total variation in the subset data. Note: 4 observations are missing.*

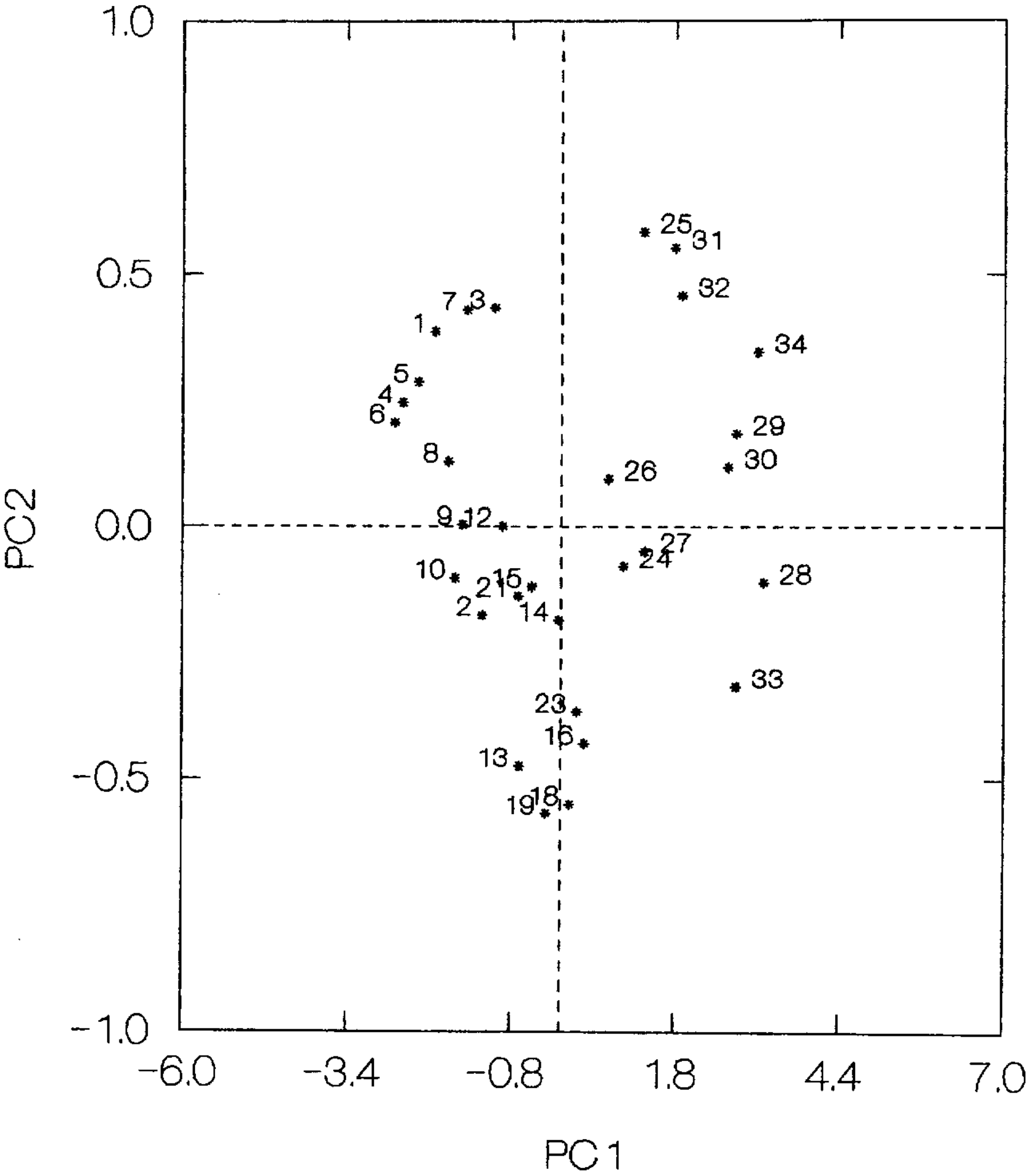


Figure 8.4a *Scatter diagram for data 1 plotted on the first three PCs computed from variables 2,5 and 12 only. The diagram accounts for all of the total variation in the subset data. Note: 2 observations are missing.*

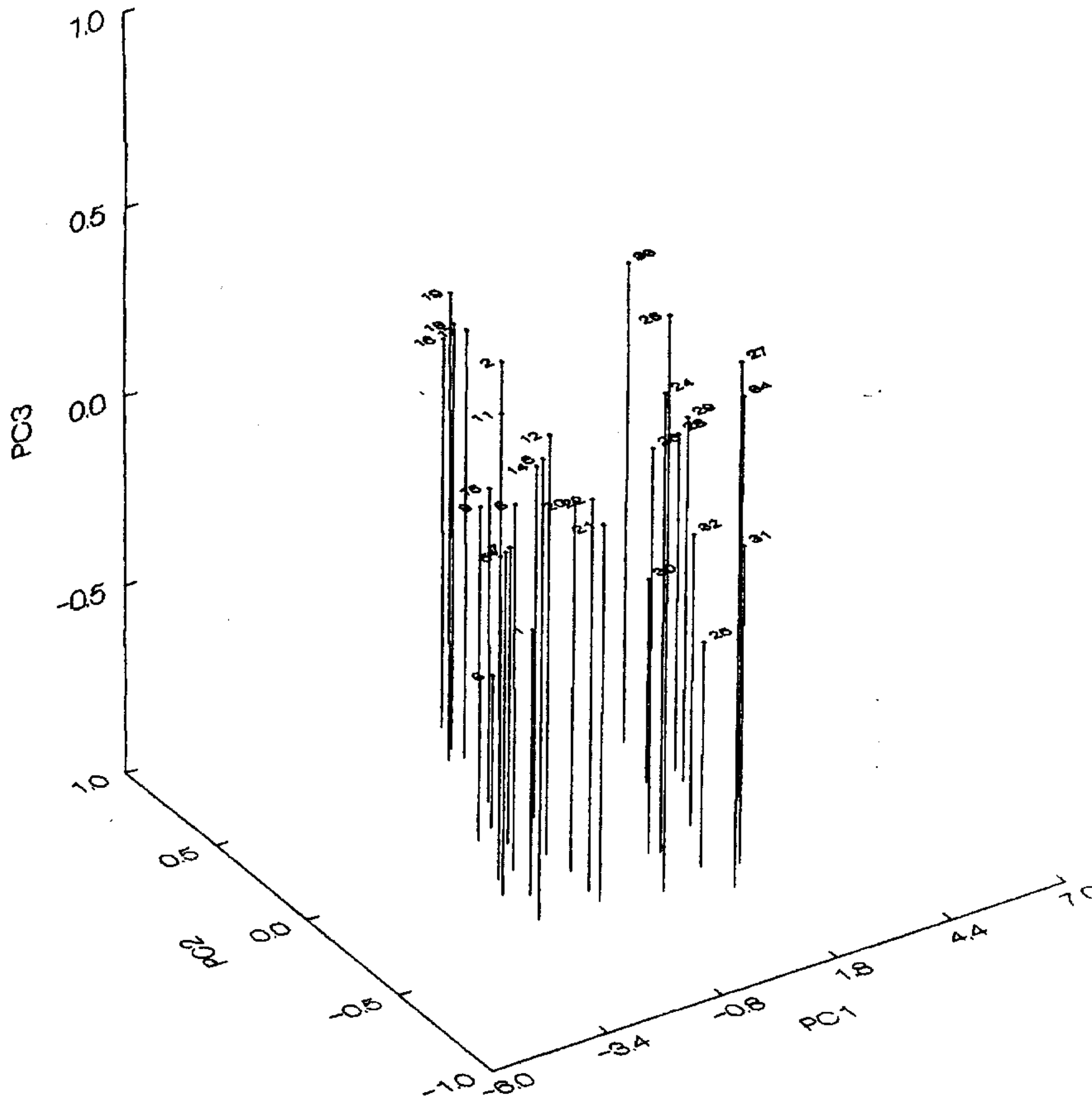


Figure 8.4b *Scatter diagram for data 1 plotted on the first two PCs computed from variables 2,5 and 12 only. The diagram accounts for 98.1% of the total variation in the subset data. Note: 2 observations are missing.*

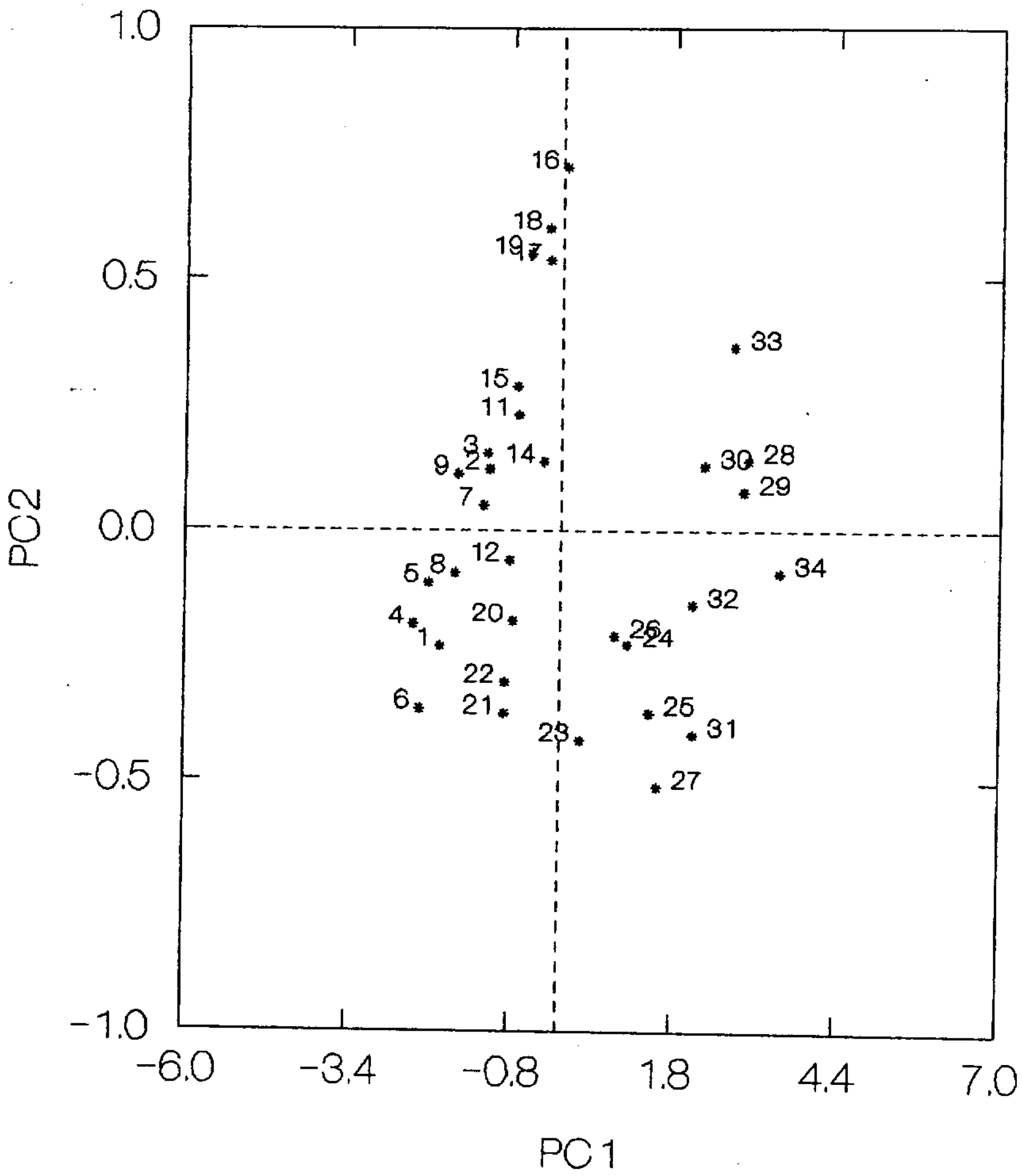


Figure 8.5a *Scatter diagram for data 1 plotted on the first three PCs computed from variables 1,5 and 11 only. The diagram accounts for all of the total variation in the subset data.*

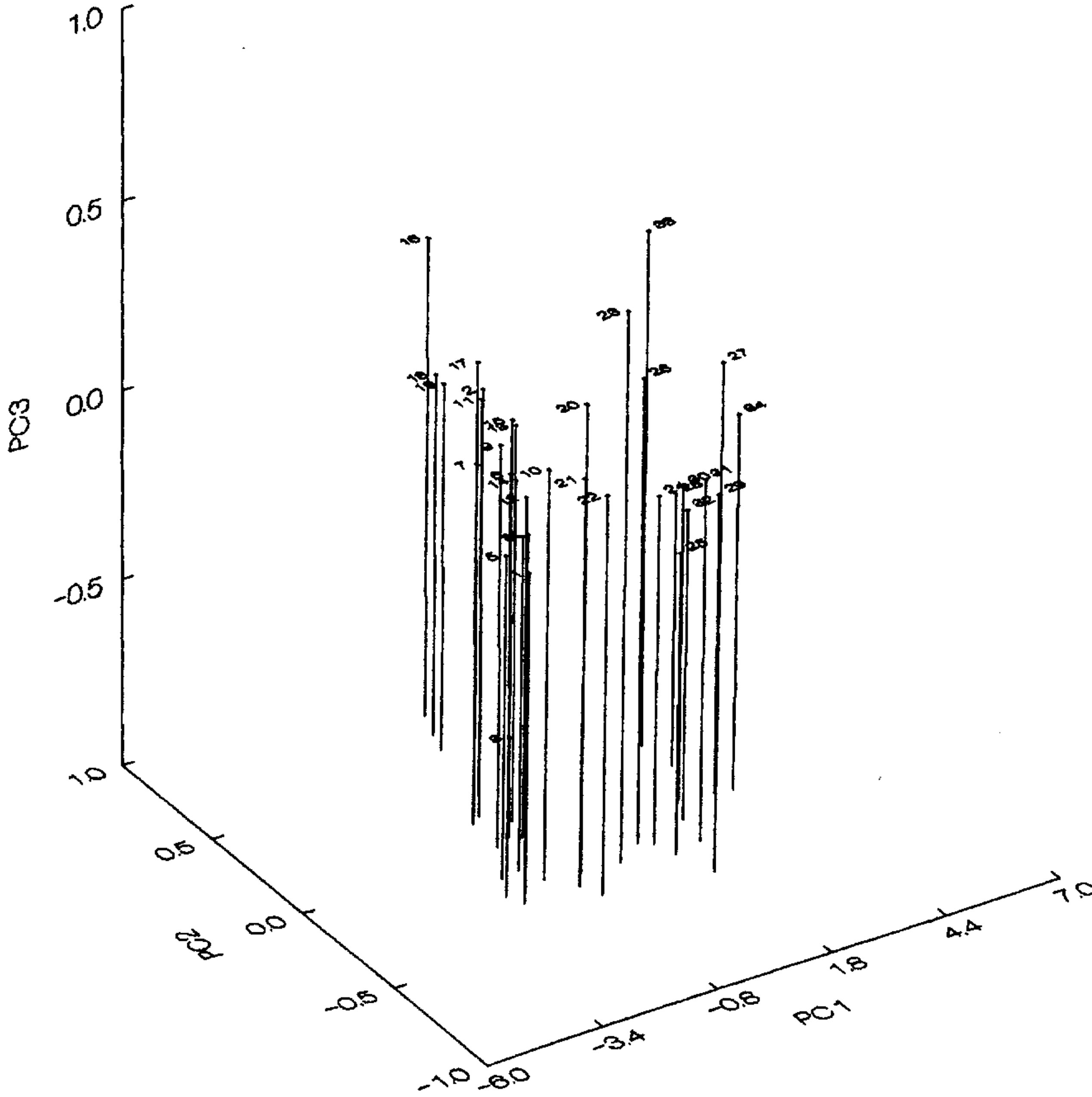
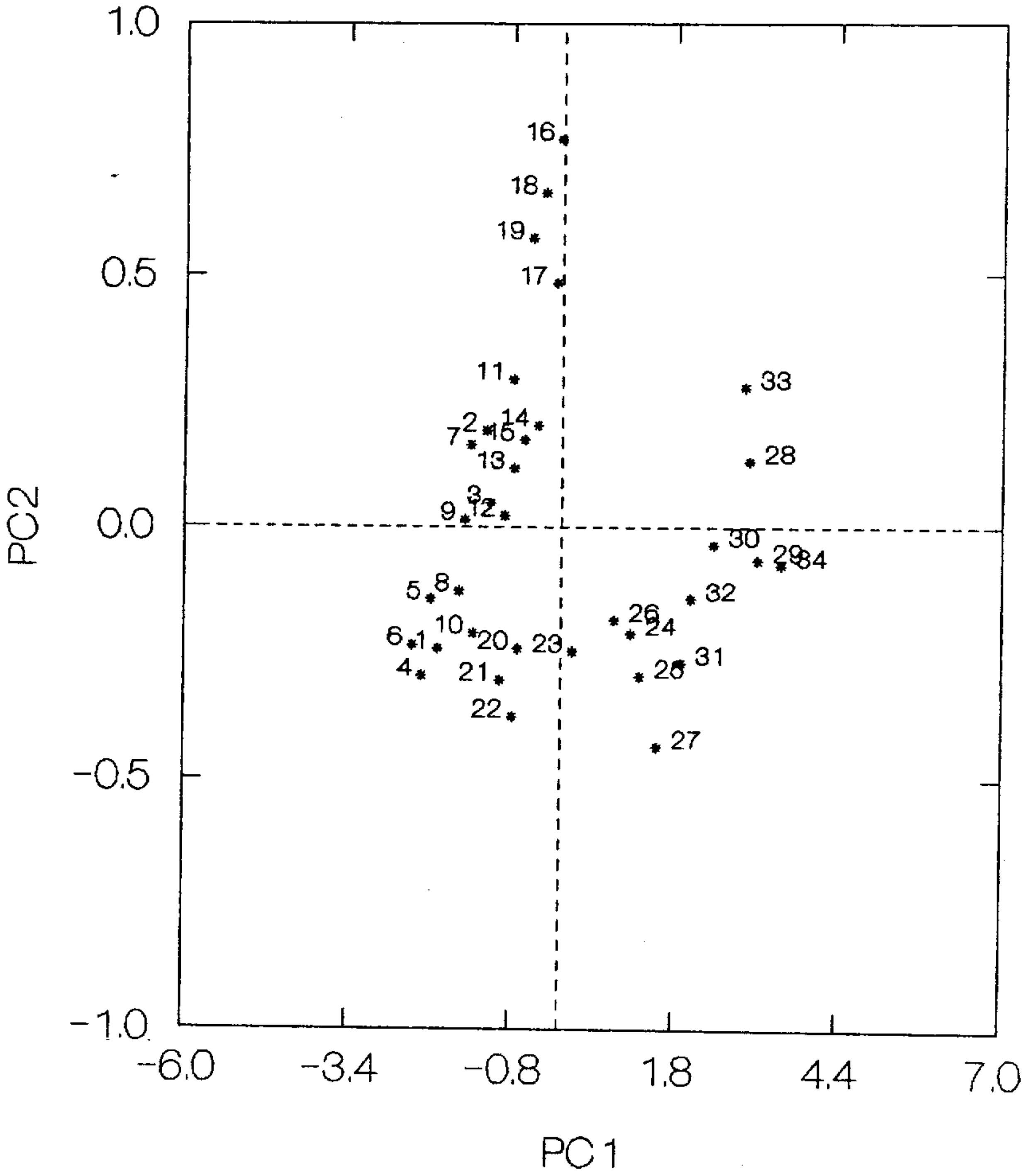


Figure 8.5b *Scatter diagram for data 1 plotted on the first two PCs computed from variables 1,5 and 11 only. The diagram accounts for 98.4% of the total variation in the subset data.*



Turning to the case where the choice of the number of components is made using the proposed bootstrap technique $\tilde{W}$ (i.e., k = 4), first consider the plots in Figures 8.6a and 8.6b. These are respectively the *3-dimensional* and *2-dimensional* representations for the variable subset {2,5,7,12} chosen by the M^2-procrustes criterion. Comparison of these plots with the ones in Figures 8.1a and 8.1b, respectively, seems to suggest that the criterion M^2 preserves the overall structure present among *all* 12 variables. However, in addition to the structure being preserved, the two groups of observations identified when the W criterion is used to choose the value of k are more prominent here. Figures 8.7a and 8.7b respectively show the best *3-dimensional* and *2-dimensional* representations for the subset of variables {3,5,6,12} selected by the criteria $\hat{\gamma}_5$, $\hat{\gamma}_6$ and δ^*, while the plots in Figures 8.8a and 8.8b are respectively the best *3-dimensional* and *2-dimensional* representations for the subset {1,5,6,12} chosen by the criterion δ. By comparing each of the *3-dimensional* plots above with the plot in Figure 8.1a (which arises from the entire set of variables), and each of the *2-dimensional* plots with the corresponding plot in Figure 8.1b, the criteria $\hat{\gamma}_5$, $\hat{\gamma}_6$, δ and δ^* are shown to recover the overall features of the complete data satisfactorily. These criteria are also shown to identify the two major groups of the data more prominently here than when W is used to determine the number of components. It seems that the M^2 criterion combined with the choice of the number of components using the W method yields the most straight forward representation of the original data (compare Figure 8.2a with Figure 8.1a and Figure 8.2b with Figure 8.1b). $\hat{\gamma}_5$ behaves similarly except that the plots which arise from the original data in Figures 8.1a and 8.2b seem to have been rotated through an angle

Figure 8.6a *Scatter diagram for data 1 plotted on the first three PCs computed from variables 2,5, 7 and 12 only. The diagram accounts for 99.3% of the total variation in the subset data. Note: 3 observations are missing.*

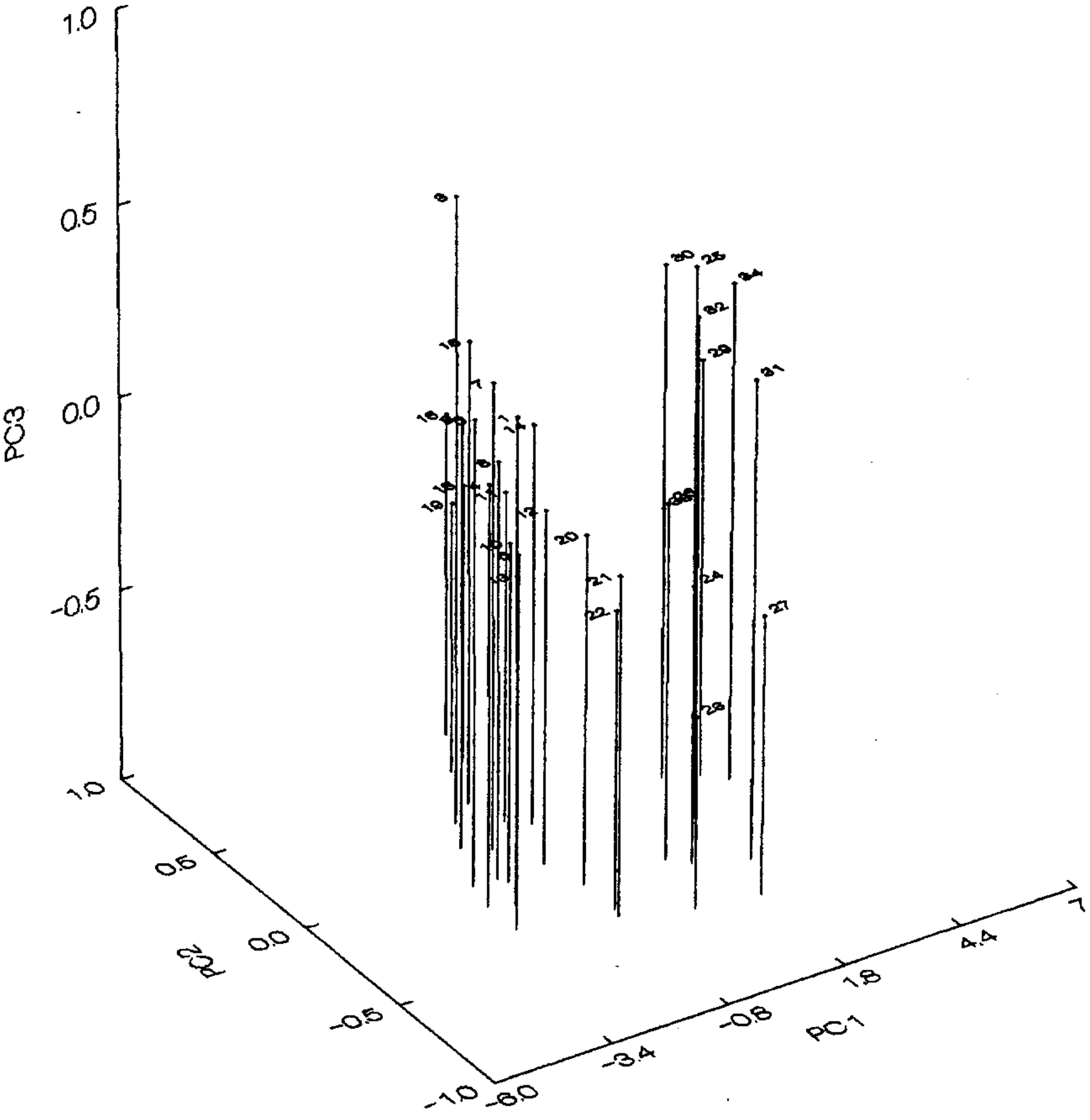


Figure 8.6b *Scatter diagram for data 1 plotted on the first two PCs computed from variables 2,5, 7 and 12 only. The diagram accounts for 97.6% of the total variation in the subset data. Note: 3 observations are missing.*

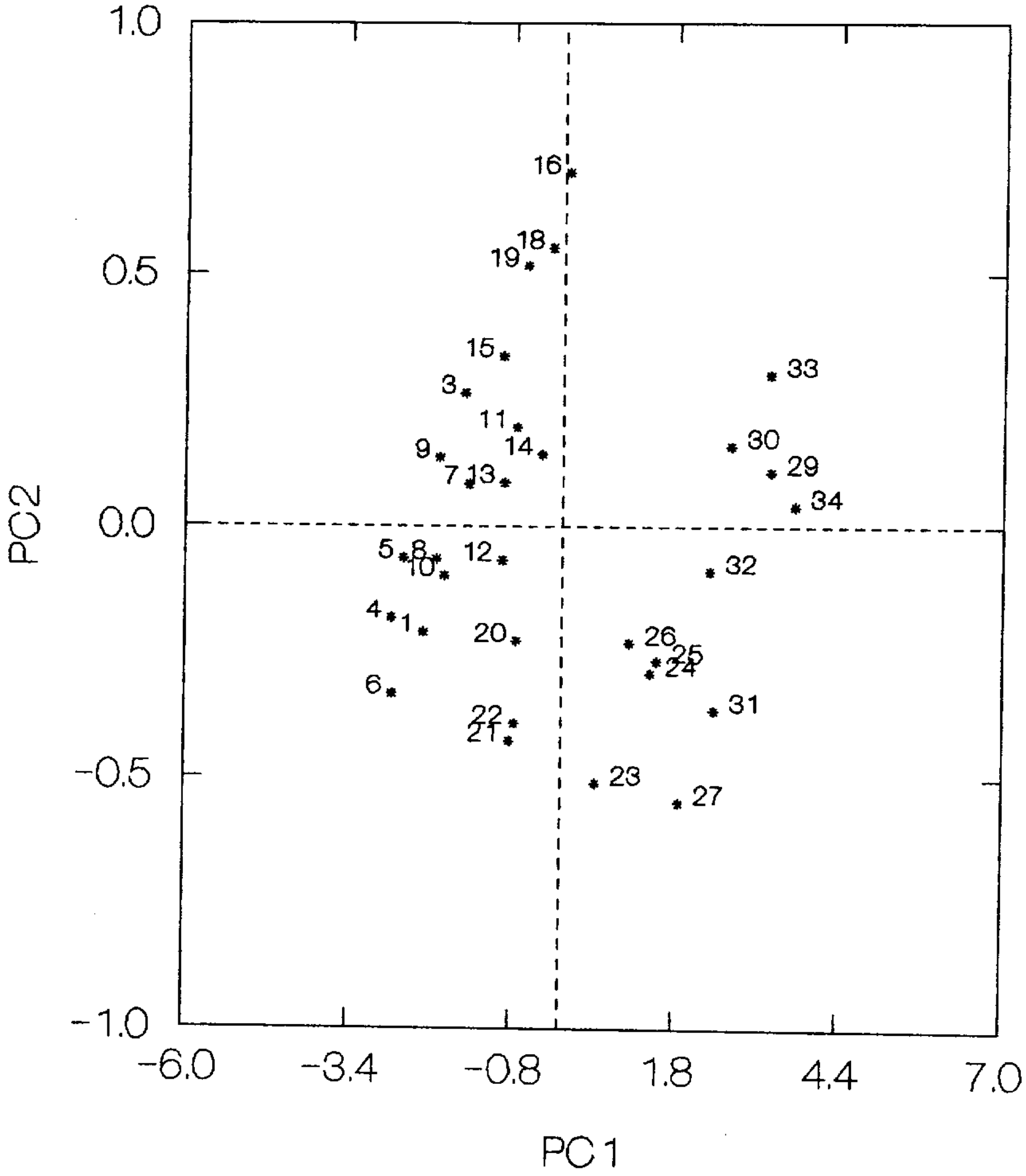


Figure 8.7a *Scatter diagram for data 1 plotted on the first three PCs computed from variables 3,5, 6 and 12 only. The diagram accounts for 99.1% of the total variation in the subset data.*

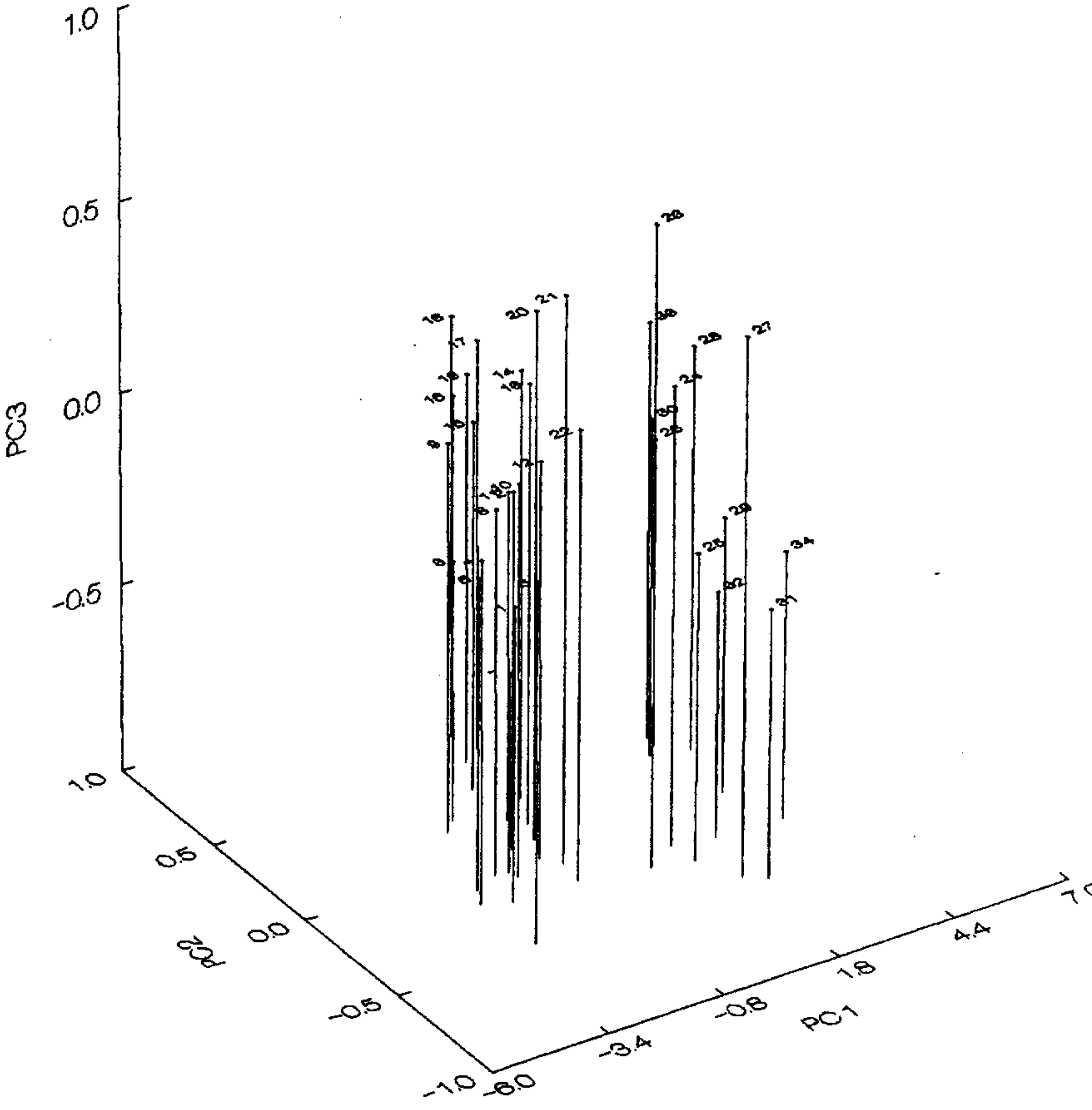


Figure 8.7b *Scatter diagram for data 1 plotted on the first two PCs computed from variables 3,5, 6 and 12 only. The diagram accounts for 97.6% of the total variation in the subset data.*

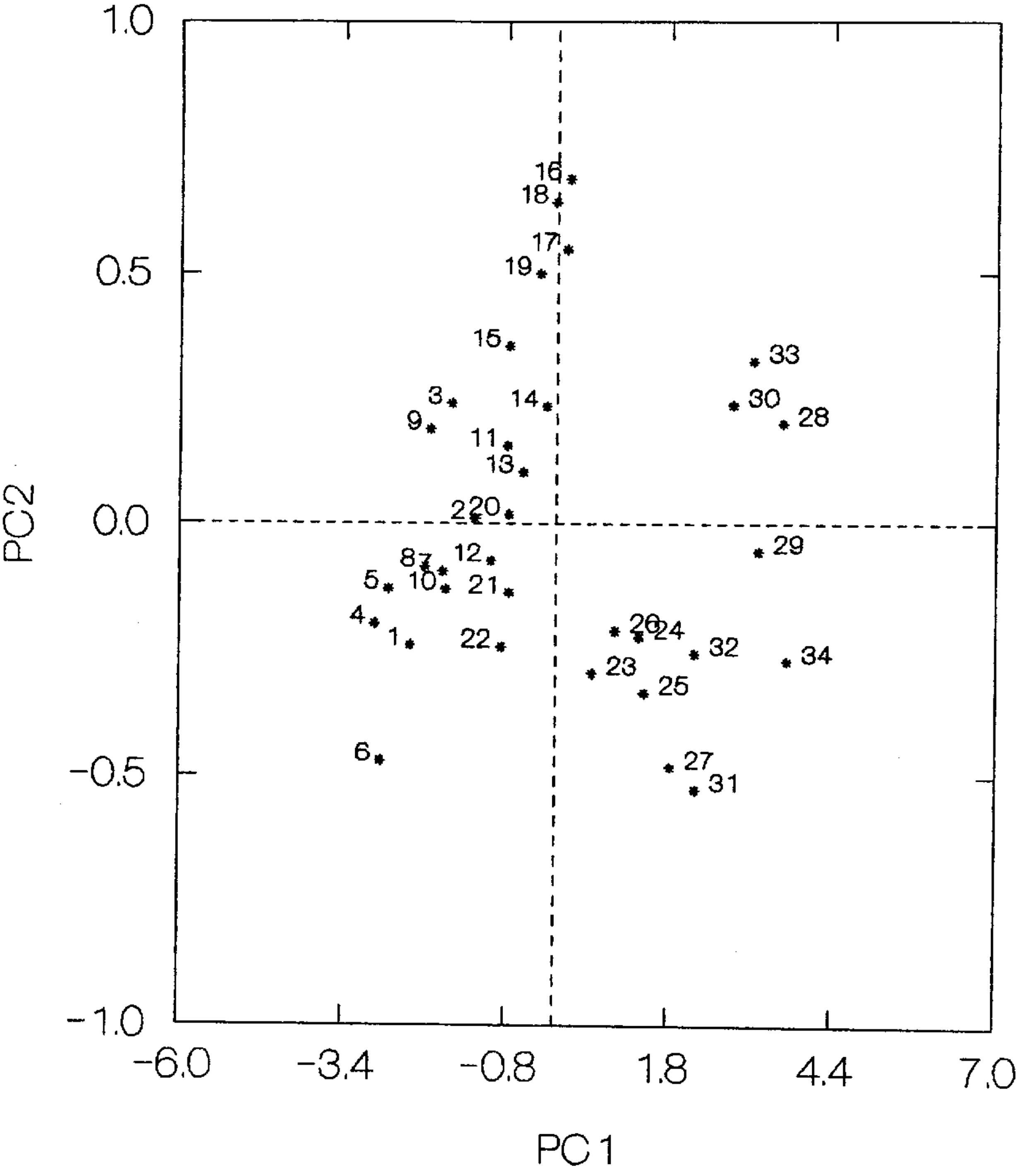


Figure 8.8a *Scatter diagram for data 1 plotted on the first three PCs computed from variables 1,5, 6 and 12 only. The diagram accounts for 99.1% of the total variation in the subset data.*

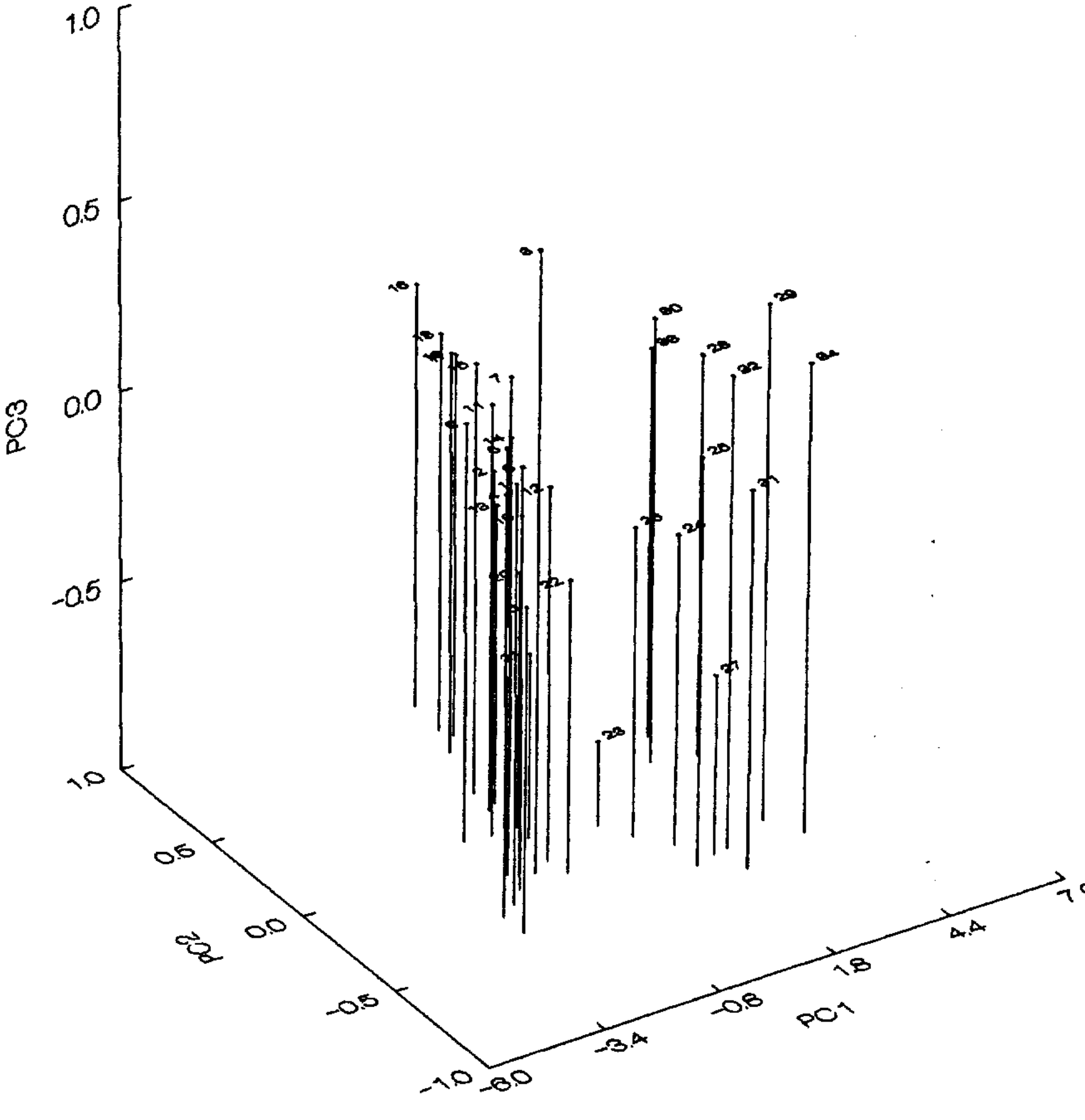
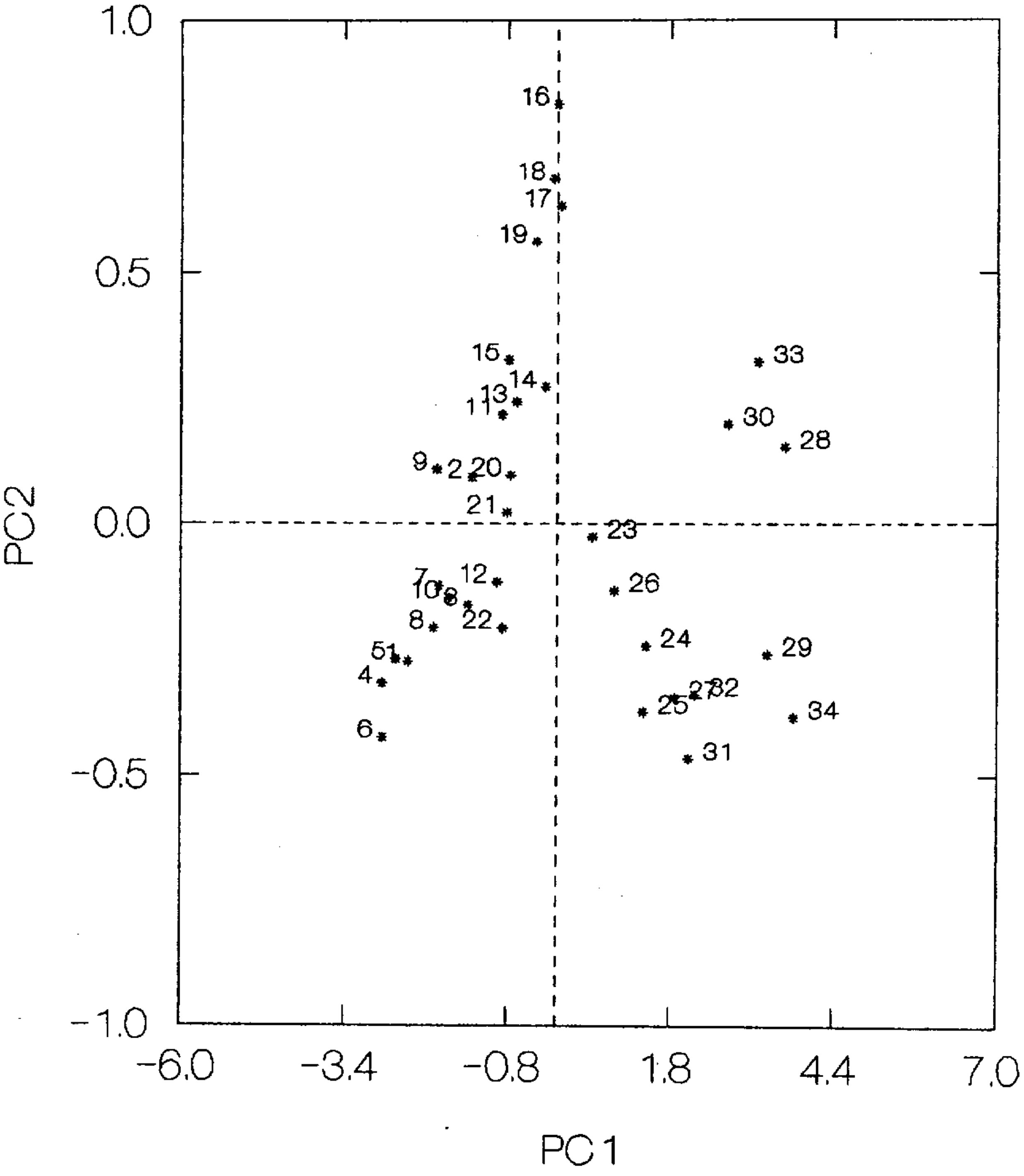


Figure 8.8b *Scatter diagram for data 1 plotted on the first two PCs computed from variables 1,5, 6 and 12 only. The diagram accounts for 97.3% of the total variation in the subset data.*



of 180° and reflected as described before. When used with the choice of the number of components according to W, the other criteria $\hat{\gamma}_6$, δ and δ^* almost exactly reproduce the data. However, the observation 23 seems to have been shifted from its 'original' position. Furthermore, other multivariate patterns are revealed here, including the enhancement of the two groupings of observations identifiable in the *3-dimensional* and *2-dimensional* plots of the complete data. The latter behaviour is also apparent in *all* the criteria when used with the choice of the number of components according to the $\tilde{W}$ method. If we consider the two groupings of observations described above; *group 1*, i.e., observations 1-22 is virtually unaltered by *all* combinations of the various variable selection criteria with the choices of the number of components according to the W and the $\tilde{W}$ methods. On the other hand, *group 2*, i.e., observations 23-34 keeps changing with respect to the choice of W, $\tilde{W}$ and the selection criteria.

Next we consider the results obtained from *data 2* which falls within the *small-sized grouped* data category. Recall from section 8.2 that the dates November 1985, February 1986 and June 1986 during which the progress tests were administered are considered as the 'groups' for this data set in our investigation. Both the techniques W and $\tilde{W}$ suggest that the dimensionality of this data set is 3 (i.e., $k = 3$). Figures 8.9a and 8.9b respectively show the best *3-dimensional* and *2-dimensional* representations of the complete data, while Figures 8.10a and 8.10b respectively show the *3-dimensional* and *2-dimensional* representations for the subset {1,3,8} chosen by the criteria M^2, $\hat{\gamma}_6$, δ and δ^*. *Rotation* of the plot in Figure 8.10a through an angle of 180° followed by a *reflexion* yields a picture similar to the plot in Figure 8.9a, while

Figure 8.9a *Scatter diagram for data 2 plotted on the first three PCs computed from all eight variables. The diagram accounts for 90.0% of the total variation in the data. Legend: * = November 1985, + = February 1986, o = June 1986.*

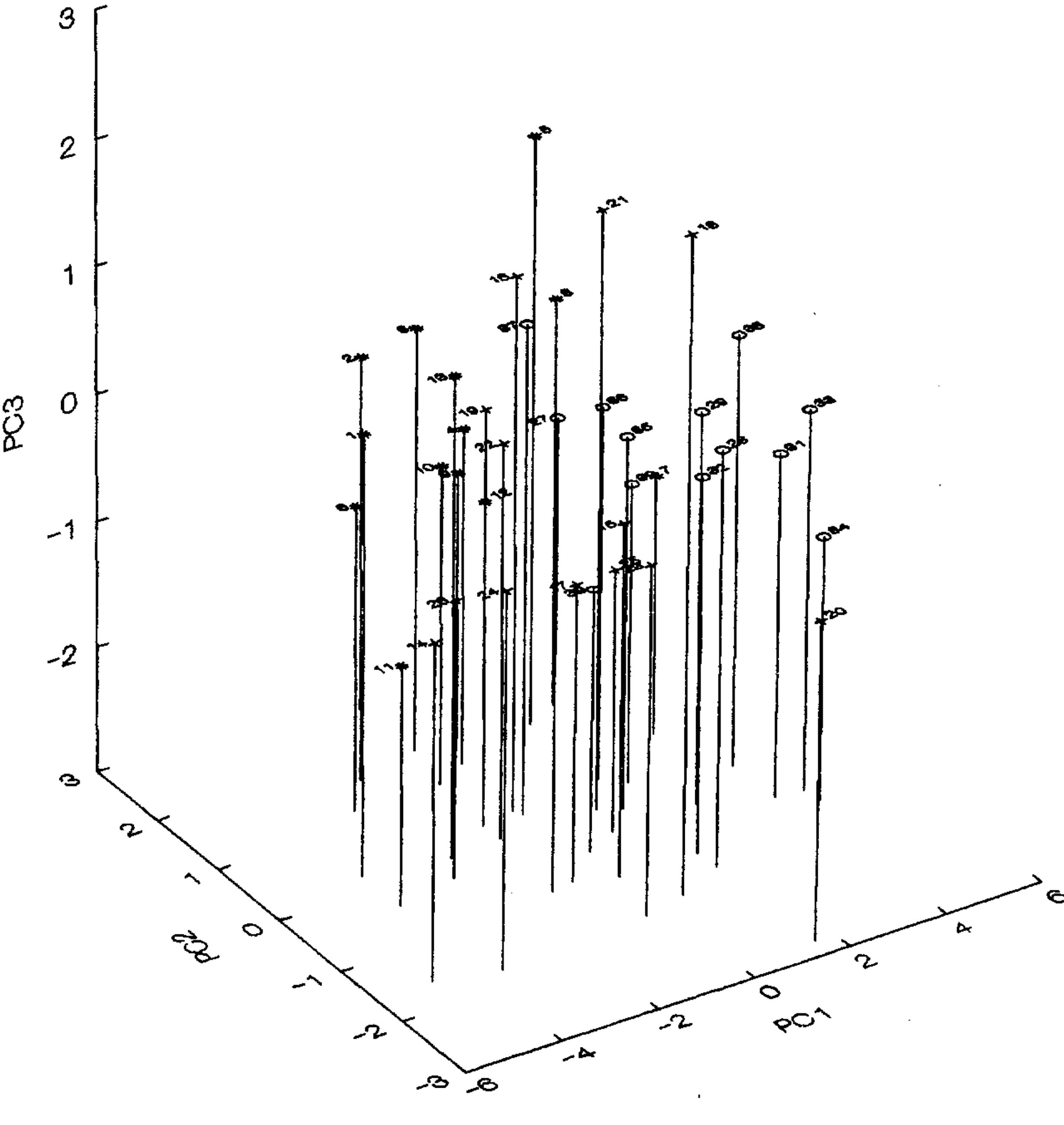


Figure 8.9b *Scatter diagram for data 2 plotted on the first two PCs computed from all eight variables. The diagram accounts for 82.1% of the total variation in the data. Legend: * = November 1985, + = February 1986, o = June 1986.*

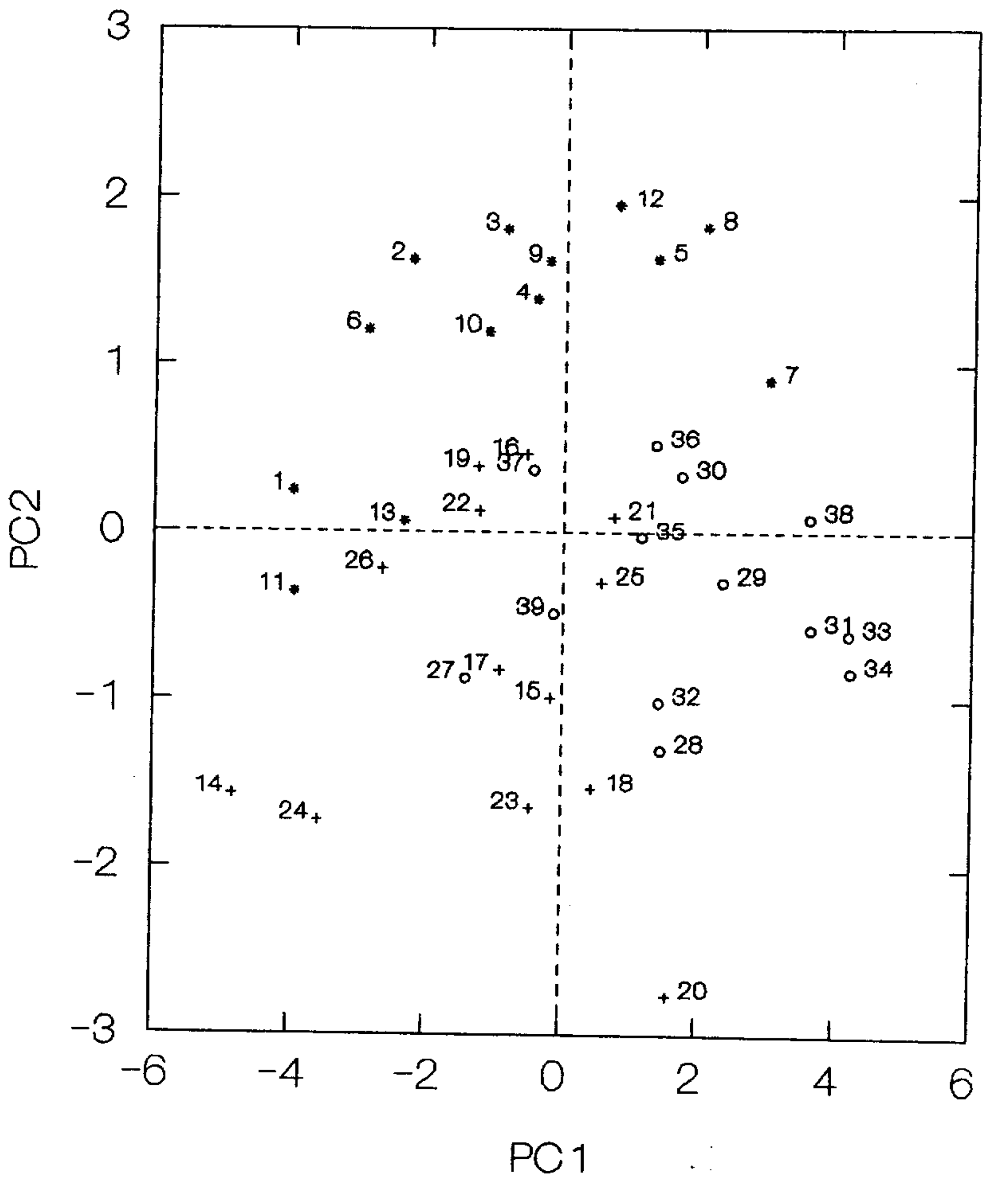


Figure 8.10a *Scatter diagram for data 2 plotted on the first three PCs computed from variables 1,3 and 8 only. The diagram accounts for all of the total variation in the subset data. Legend: * = November 1985, + = February 1986, o = June 1986. Note: 7 observations are missing.*

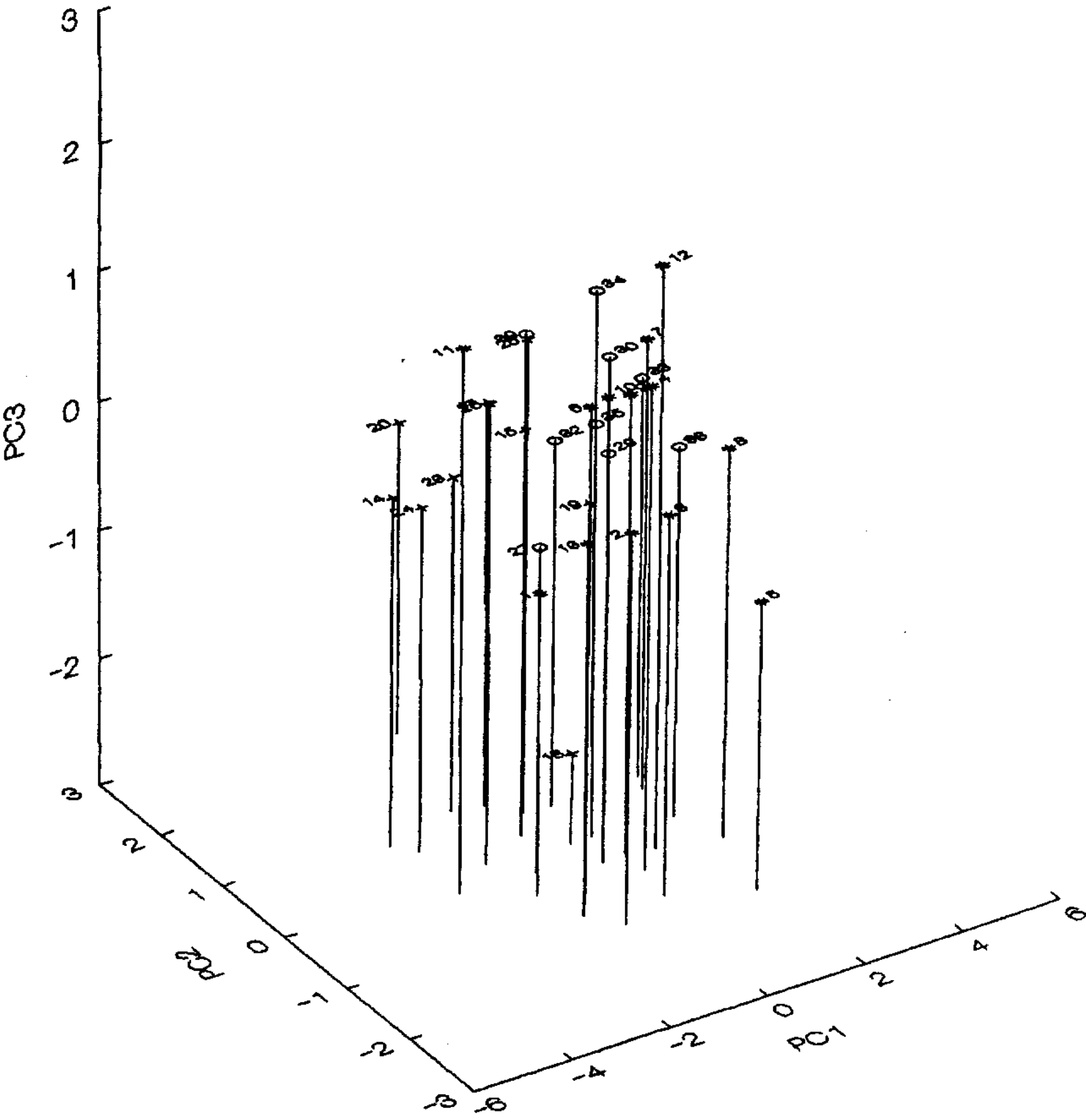


Figure 8.10b *Scatter diagram for data 2 plotted on the first two PCs computed from variables 1,3 and 8 only. The diagram accounts for 83.0% of the total variation in the subset data. Legend:* * = *November 1985,* + = *February 1986,* o = *June 1986. Note: 7 observations are missing.*

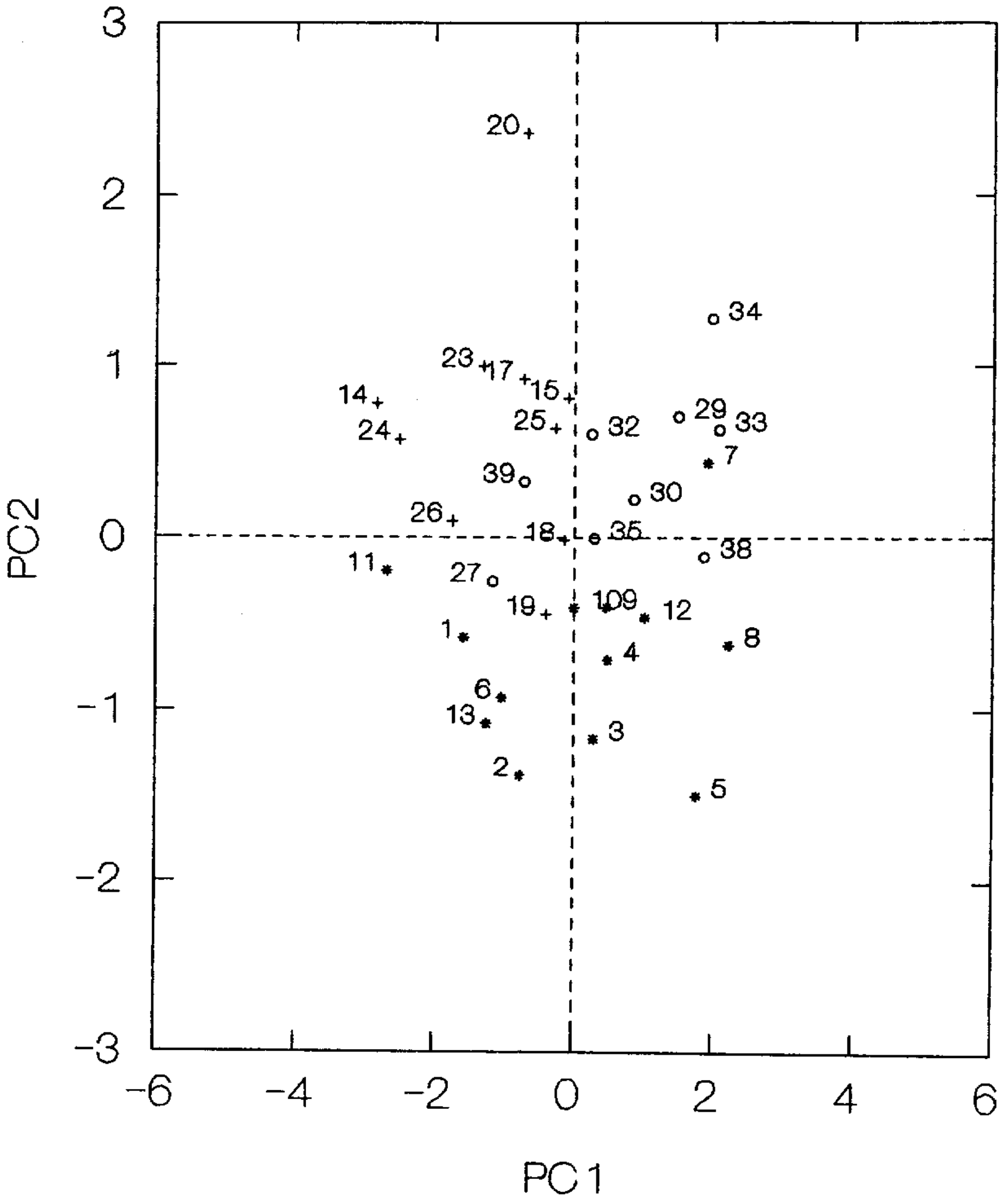


Figure 8.11a *Scatter diagram for data 2 plotted on the first three PCs computed from variables 1,7 and 8 only. The diagram accounts for all of the total variation in the subset data. Legend: * = November 1986, + = February 1986, o = June 1986. Note: 5 observations are missing.*

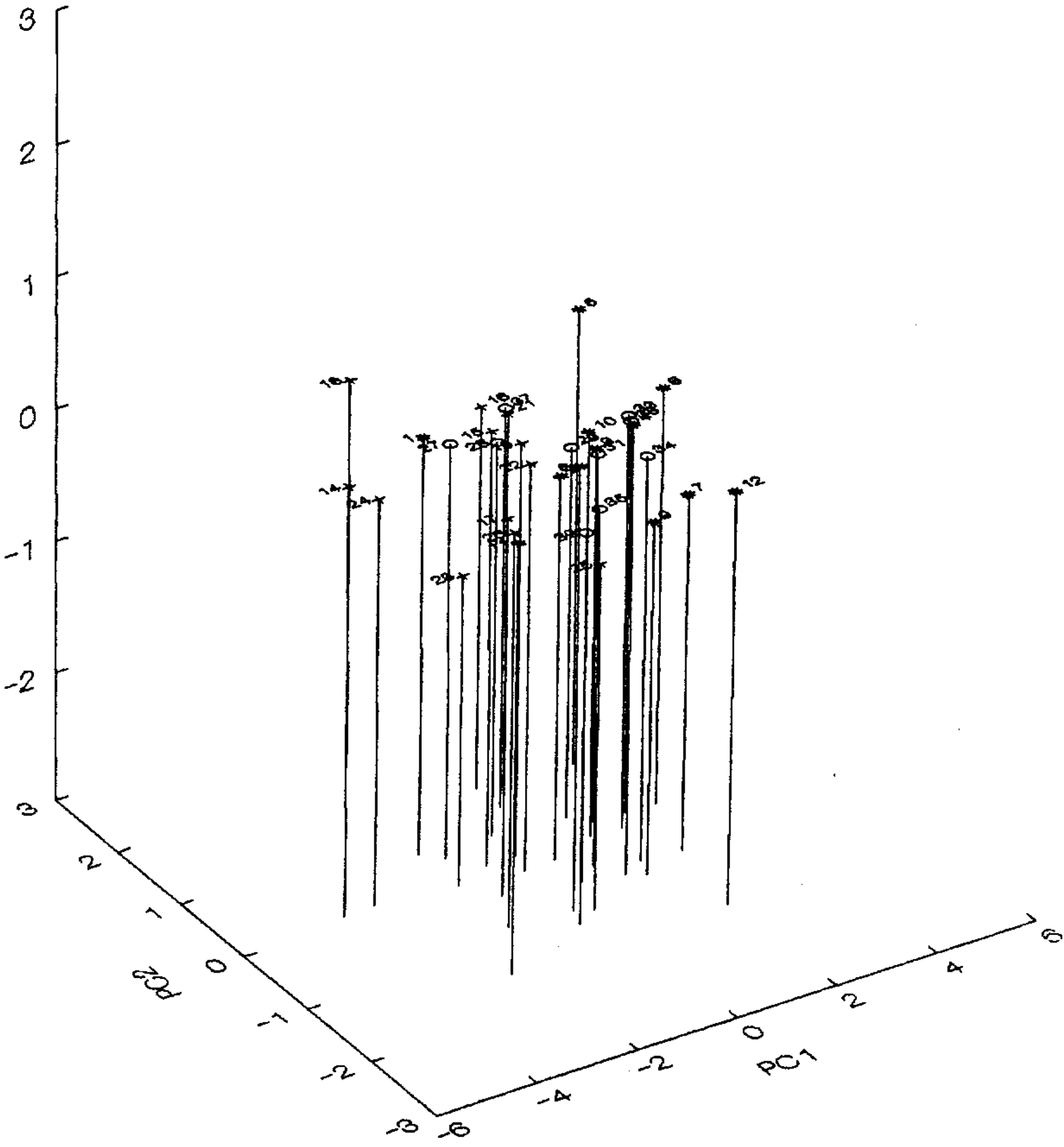
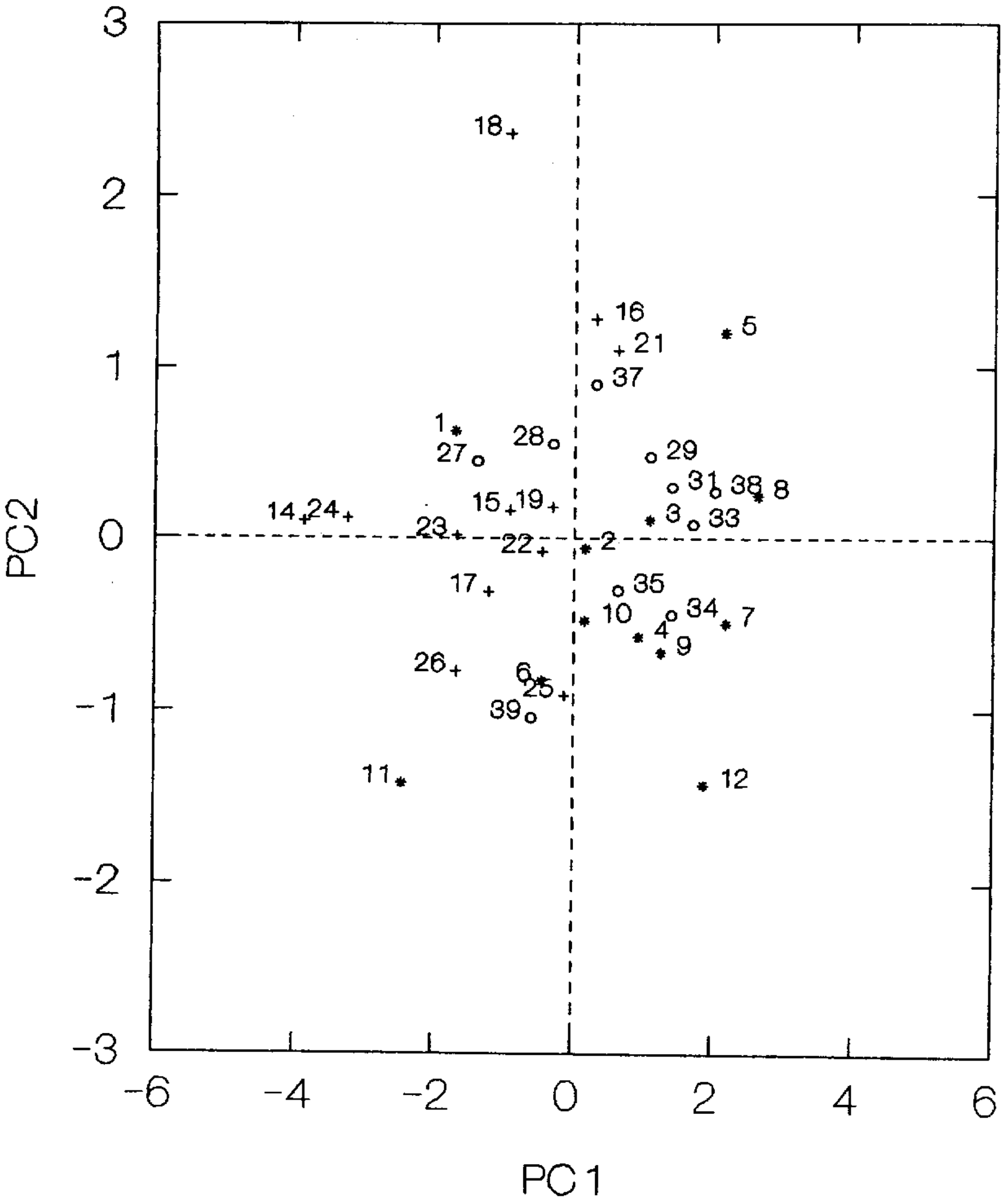


Figure 8.11b *Scatter diagram for data 2 plotted on the first two PCs computed from variables 1,7 and 8 only. The diagram accounts for 96.8 of the total variation in the subset data. Legend: • = November 1986, + = February 1986, o = June 1986. Note: 5 observations are missing.*



the same transformation performed on the plot of Figure 8.10b yields a plot comparable to the one in Figure 8.9b. In other words, the *three* groupings of observations indicating the different dates during which the tests were administered are reproduced almost exactly. In the plots corresponding to the subset data the groups defined by the dates February 1986 and June 1986 overlap - A feature present in the plots corresponding to the complete data. This is an indication that the (*group-*) *structure* is recovered very well, in this regard, by using M^2, $\hat{\gamma}_6$, δ or δ^* as criteria for variable selection. Although the general features of the complete data can be identified in the plots of Figures 8.11a and 8.11b which arise from the subset {1,7,8} chosen by the the $\hat{\gamma}_5$ criterion (after an appropriate *rotation* and *reflexion*), not only the groups defined by the dates February 1986 and June 1986 overlap, but also the groups defined by November 1985 and June 1986 show a considerable overlap. Hence, for a small sample of the 'Venezuela data' , $\hat{\gamma}_5$ seems to be a poor candidate for preserving (*group-*) *structure*.

Finally, we consider the results of applying the various criteria on *data 3* which is *large* and *grouped*. Once again, both the techniques W and $\tilde{W}$ suggest the same dimensionality for this data set and this is 2 (i.e., k = 2). Here, M^2 chooses the subset {3,8}, while $\hat{\gamma}_5$, $\hat{\gamma}_6$, δ and δ^* all choose the subset {2,8}. The best *2-dimensional* representation of the complete data is shown in Figure 3.12, while Figures 8.13 and 8.14 show the best *2-dimensional* representations for the subsets {3,8} and {2,8}, respectively. Note that the plots in Figures 8.13 and 8.14 each explain 100% of the variation in the subset data, and hence the axes (i.e., PC1 and PC2) are merely a geometrical rotation of the original variables. Plots similar to the one in Figure 8.12 can be obtained from a *rotation*

Figure 8.12 *Scatter diagram for data 3 plotted on the first two PCs computed from all eight variables. The diagram accounts for 79.0% of the total variation in the data. Legend: * = November 1985, + = February 1986, ∘ = June 1986.*

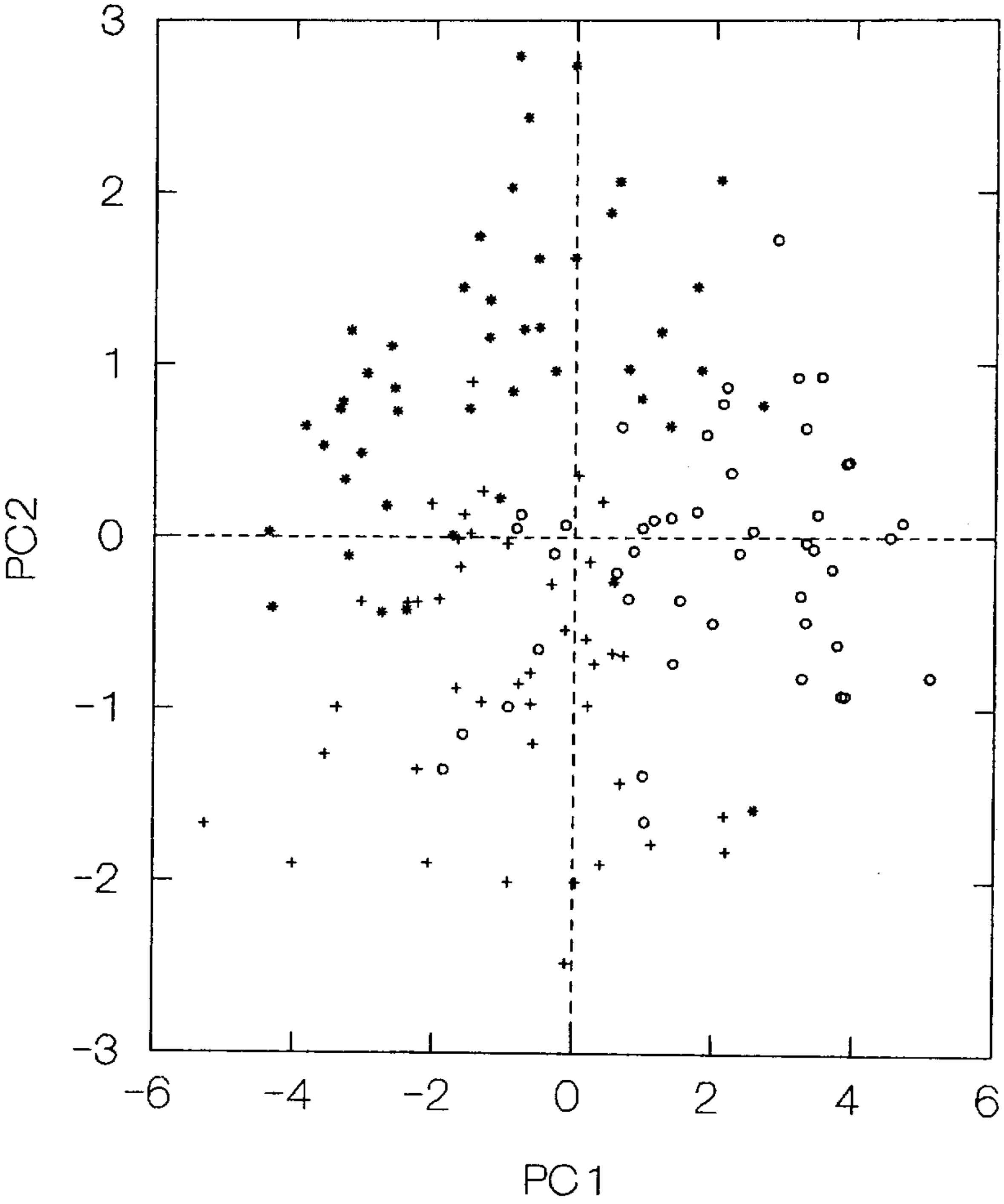


Figure 8.13 *Scatter diagram for data 3 plotted on the first two PCs computed from variables 3 and 8 only. The diagram accounts for all of the variation in the subset data. Legend: * = November 1985, + = February 1986, ○ = June 1986.*

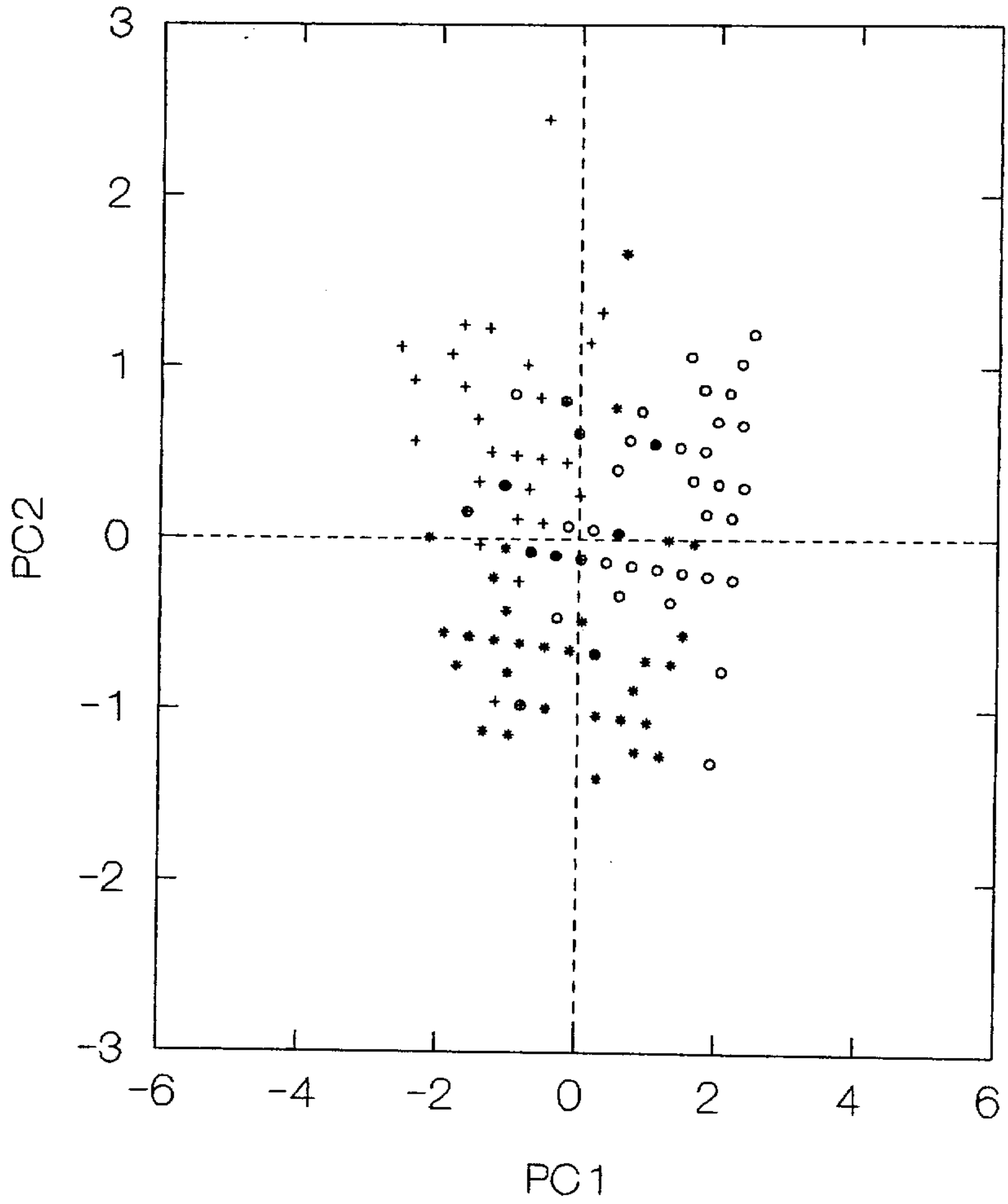
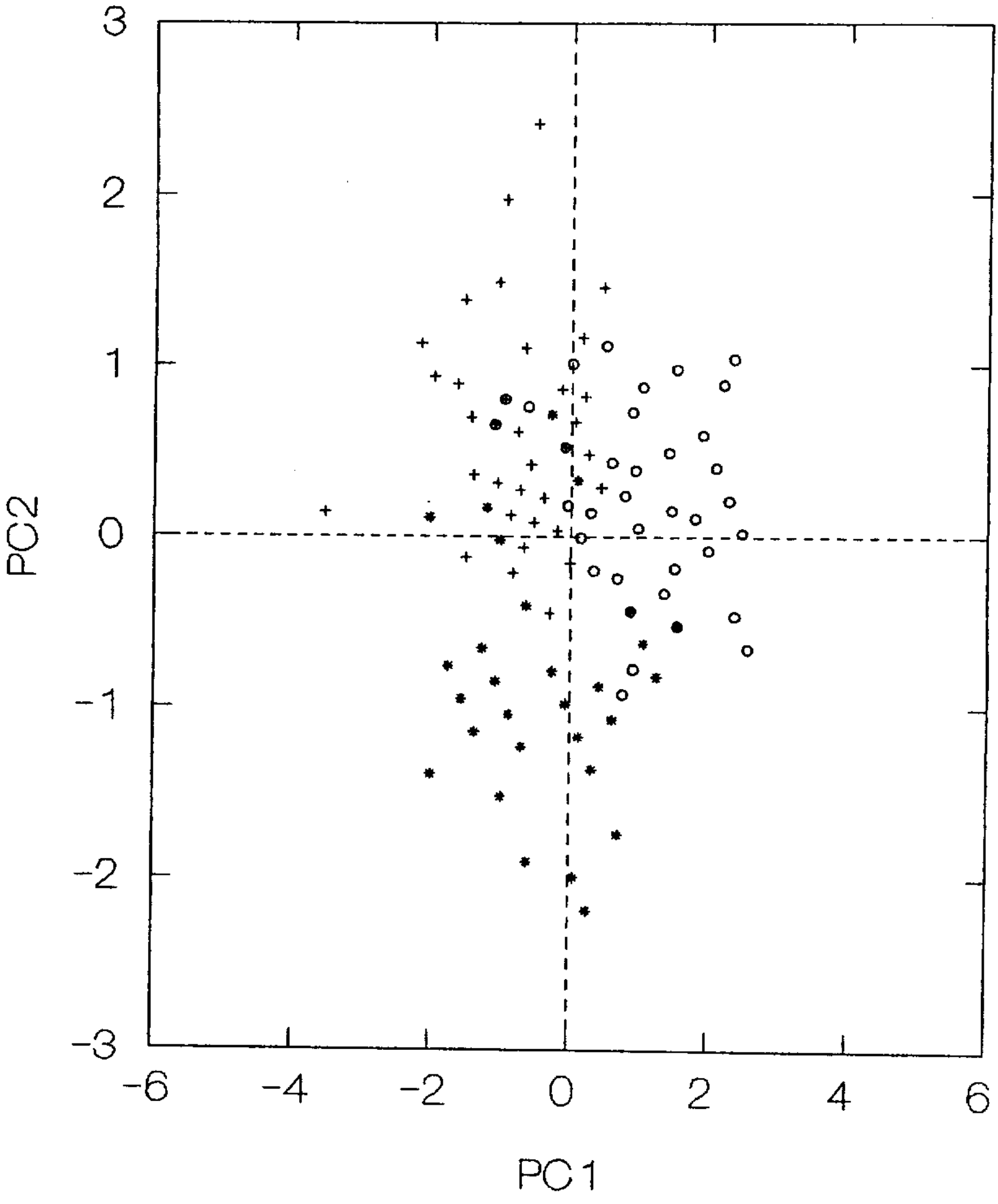


Figure 8.14 *Scatter diagram for data 3 plotted on the first two PCs computed from variables 2 and 8 only. The diagram accounts for all of the variation in the subset data. Legend: * = November 1985, + = February 1986, o = June 1986.*



through an angle of 180° followed by a *reflexion* of each of the plots in Figures 8.13 and 8.14. Hence, all five criteria (M^2, $\hat{\gamma}_5$, $\hat{\gamma}_6$, δ and δ^*) choose subsets of variables which leave the (*group-*) *structure* intact. The plot resulting from appropriate *rotation* and *reflexion* of Figure 8.14 resembles the plot in Figure 8.12 more closely than the corresponding plot of Figure 8.13. Hence, it seems that any one of the criteria $\hat{\gamma}_5$, $\hat{\gamma}_6$, δ and δ^* is more appropriate than M^2 for preserving (*group-*) *structure*, for *large-grouped* data.

8.3.2 Evaluation using the minimum multiple correlation

In this section we evaluate the performance of the various variable selection criteria on real data when choice regarding the best subsets of retained variables is based on maximizing R_m, the minimum multiple correlation between the k selected variables and any of the p-k discarded ones. Computer programs written in GAUSS (1988) which are similar to programs 11 and 12 in appendix B are used to compute the multiple correlations for all possible subsets. These programs are applied to the data sets *data 1*, *data 2* and *data 3*, and for each data set, R_m^*, the maximized value of R_m and the R_m values corresponding to the subsets chosen by the various selection criteria are identified in the program output. The aim is to compare the variable selection criteria among themselves on the basis of the extent to which the subsets they retain maximize the minimum multiple correlation R_m. The results of this operation are summarized in Tables 8.1a, 8.1b, 8.2 and 8.3. Table 8.1a gives the results of applying the variable selection criteria on *data 1* with k, the number of components chosen using the W method, while presented in Table 8.1b are the results based on the same data set except that here, the choice of k is made using the $\tilde{W}$ method. Tables

8.2 and 8.3 give the results of the analyses of the data sets *data 2* and *data 3*, respectively. As noted before, for the last two data sets, use of the W method and the $\tilde{W}$ method lead to the same choice of k.

First we consider the results based on *data 1* which falls within the *small-sized ungrouped* data category. As noted before, W chooses $k = 3$, while $\tilde{W}$ chooses $k = 4$ for this data set. Hence, the number of retained variables is set equal to 3 and 4, respectively. Comparing the values of the minimum multiple correlations in Table 8.1a shows that δ^* chooses the subset {1,5,11} with the largest R_m value, and hence ranks 'best' among the selection criteria. The criteria $\hat{\gamma}_6$ and δ follow very closely, choosing the subset {2,5,12} whose R_m value is the second largest. On the other hand, while the criteria M^2 and $\hat{\gamma}_5$ choose satisfactory subsets {3,6,10} and {2,5,6}, respectively, with a reasonably large value of R_m, their performance is slightly worse than the performances of the other criteria.

When the dimensionality is 4 in Table 8.1b, the criterion δ becomes the 'best' choice for selecting variables for *data 1*; the choice being of the subset {1,5,6,12} with the largest R_m value. The criteria $\hat{\gamma}_5$, $\hat{\gamma}_6$ and δ^* are slightly worse in this case, choosing the subset {3,5,6,12} whose R_m value is the second largest. Although the subset {2,5,7,12} chosen by the M^2 criterion is satisfactory, with an R_m value larger than 0.9, this criterion is the worst among the variable selection criteria. While the R_m values corresponding to the choice of k according to the $\tilde{W}$ method are larger than those based on the choice of k using W, this is not necessarily an indication that $\tilde{W}$ is a better method for choosing k than W. This follows from the fact that the R_m values for $k = t+1$ will generally be larger than those for $k = t$, $t = 1, 2, \ldots$, etc. Hence, in our

Table 8.1a *Subsets retained by the various methods for data 1 and the corresponding values of the minimum multiple correlation when the dimensionality is determined using* W (*i.e.,* $k = 3$).

Method	Retained subset	Minimum multiple correlation (R_m)
M^2	{3,6,10}	0.960
$\hat{\gamma}_5$	{2,5,6}	0.960
$\hat{\gamma}_6$	{2,5,12}	0.961
δ	{2,5,12}	0.961
δ^*	{1,5,11}	0.962
Maximized minimum multiple correlation (R^*_m)		0.974

Table 8.1b *Subsets retained by the various methods for data 1 and the corresponding values of the minimum multiple correlation when the dimensionality is determined using* $\tilde{W}$ (*i.e.,* $k = 4$).

Method	Retained subset	Minimum multiple correlation (R_m)
M^2	{2,5,7,12}	0.978
$\hat{\gamma}_5$	{3,5,6,12}	0.985
$\hat{\gamma}_6$	{3,5,6,12}	0.985
δ	{1,5,6,12}	0.986
δ^*	{3,5,6,12}	0.985
Maximized minimum multiple correlation (R^*_m)		0.988

Table 8.2 *Subsets retained by the various methods for data 2 and the corresponding values of the minimum multiple correlation. k, the dimensionality is 3.*

Method	Retained subset	Minimum multiple correlation (R_m)
M^2	{1,3,8}	0.749
$\hat{\gamma}_5$	{1,7,8}	0.602
$\hat{\gamma}_6$	{1,3,8}	0.749
δ	{1,3,8}	0.749
δ^*	{1,3,8}	0.749
Maximized minimum multiple correlation (R^*_m)		0.802

Table 8.3 *Subsets retained by the various methods for data 3 and the corresponding values of the minimum multiple correlation. k, the dimensionality is 2.*

Method	Retained subset	Minimum multiple correlation (R_m)
M^2	{3,8}	0.592
$\hat{\gamma}_5$	{2,8}	0.542
$\hat{\gamma}_6$	{2,8}	0.542
δ	{2,8}	0.542
δ^*	{2,8}	0.542
Maximized minimum multiple correlation (R^*_m)		0.597

investigation we consider only the earlier comparisons of the methods $\tilde{W}$ and W in section 8.3.1 based on the *3-dimensional* and *2-dimensional* plots of principal component scores.

Next we consider the results obtained from *data 2* which is grouped and small-sized. As noted before, both W and $\tilde{W}$ choose k = 3 for this data set. Hence, the number of retained variables is set equal to 3. For this data set (see Table 8.2), the criteria M^2, $\hat{\gamma}_6$, δ and δ^* all perform equivalently, choosing the same subset {1,3,8} whose R_m value is the closest to the maximized minimum multiple correlation R^*_m. While $\hat{\gamma}_5$'s choice of retained variables {1,7,8} is reasonable, with $R_m > 0.7R^*_m$, this criterion becomes the worst method in this case. Notice that we have noted a similar behaviour in section 8.3.1 above for the assessment of the performance of the criteria using the *3-dimensional* and *2-dimensional* plots of principal component scores.

Finally, we consider the results of applying the various criteria on *data 3* which is *large* and ungrouped. As mentioned earlier, both the methods W and $\tilde{W}$ lead to the same dimensionality for this data set and this is 2 (i.e., k = 2). By comparing the R_m values in Table 8.3, it is shown that, for this data, M^2 performs better than the other methods; its choice being of the subset {3,8} whose R_m value is the closest to the maximized minimum multiple correlation R^*_m. The other criteria all choose the subset {2,8} whose R_m value is also satisfactory, i.e., $R_m > 0.7R^*_m$.

8.4 Conclusions

It has been shown in section 8.3 above that, *three* previously published sets of data contain more variables than necessary in the 'Principal Component' sense. PCA using fewer variables yields a set of results similar to those obtained from a PCA using the entire set of variables. This is an indication that the variables in the retained subsets are sufficient for an adequate representation of the 'signal' in the data; the rest being a reflection of the 'noise'. Also, in the absence of the 'noise' variables, interesting patterns or relationships are enhanced. Hence, *only* the 'signal' variables need to be measured for future experiments.

Furthermore, the proposed criteria $\hat{\gamma}_5$, $\hat{\gamma}_6$, δ and δ^*, and the existing M^2-*procrustes* criterion are shown to be suitable for deciding which variables to retain in order to preserve the general features of the complete data. The M^2 criterion identifies the *structure-bearing* variables particularly when there are *groupings* among the observations in the data. The proposed criteria $\hat{\gamma}_5$, $\hat{\gamma}_6$, δ and δ^* are also shown to be *optimal* in this regard. However, in addition the variables identified by the criteria $\hat{\gamma}_6$, δ and δ^* also enhance other multivariate patterns present in the complete data. Since our objective is to choose those variables which carry whatever (*unknown*) multivariate structure that may be present in the data, these criteria ($\hat{\gamma}_6$, δ and δ^*) are the most suitable choices for this purpose. Hence, they gain favour over the M^2 criterion when used for variable selection in PCA. The criterion δ^* seems to be the best choice for selecting variables which reveal *group-structure* as well as other interesting patterns or relationships.

A comparison between W, the cross-validatory technique of Eastment and Krzanowski (1982) and $\tilde{W}$, the proposed bootstrap technique shows that the two techniques choose the same number of components for *two* of the data sets (i.e., *data 2* and *data 3*). When this choice of the number of components is used with the various criteria to select variables, *all* the criteria are shown to choose variables which recover *group-structure* satisfactorily. For *data 1*, however, the techniques W and $\tilde{W}$ lead to different choices of the number of components. All the criteria are shown to choose variables which preserve the general features of the complete data when used with the choice according to the W technique. However, the variables chosen also reveal ‘extra’ multivariate structure such as *group-structure* and other interesting patterns which are hidden in the presence of the ‘noise’ variables in the complete data. Using the criteria with the choice according to the technique $\tilde{W}$ yields similar results except that the *group-structure* identifiable in the complete data seems more prominent here than when W is used to choose the number of components. Hence, $\tilde{W}$ seems more suitable as the criteria for choosing the number of components in this regard. Notice that the Monte Carlo simulation study in Chapter 4 led to the same conclusion.

Finally, regarding the performance of the criteria on the basis of the extent to which the subsets they select maximize the minimum multiple correlation, all the criteria are shown to perform satisfactorily. However, the results also show that the M^2 criterion performs slightly worse than the proposed criteria in most cases, although $\hat{\gamma}_5$ also exhibits this behaviour in a few cases.

CHAPTER 9

SUMMARY, CONCLUSIONS AND FUTURE PROSPECTS

The need for methods of selecting important variables for a *Principal Component Analysis* (PCA) is largely self-evident. Firstly, one of the most popular uses of PCA is *dimensionality reduction*. Since the *Principal Components* (PCs) are linear combinations of the original variables under the constraint that they are uncorrelated and ordered in such a way that the *first few* retain most of the variation or information in the data, *dimensionality reduction* can be easily achieved by using *only* the *first few* of these PCs. If the original data set consists of p variables, and *only* the *first few* of the corresponding PCs are needed for adequate representation of the data, then it can be argued that the *true dimensionality* of the data is much less than p; the remaining dimensions being a reflection of the 'noise'. This is because it often happens that the investigator has measured more variables on each sample member than strictly necessary in an attempt to avoid ignoring essential variables. In other words, if some of the original variables are highly intercorrelated, then these variables may be effectively 'saying the same thing' and hence the *dimensionality* of the data needs to be reduced. While using *only* the *first few* PCs may successfully reduce the *dimensionality* of the data without disturbing the overall sample features, interpreting these PCs in terms of *all* the original p variables seems inevitable. Furthermore, inclusion of *all* the original variables in the analysis is one of the major causes for lack of interpretability of the PCs. Moreover, in the presence of the 'noise' variables, interesting patterns or relationships which would otherwise be revealed, are less prominent if not completely hidden. Hence, the selection of 'important' variables in PCA becomes one of the major aims of the analysis. Secondly, while an extensive amount of literature is available in the area of variable selection in the context of *Regression Analysis* and *Discriminant Analysis*, very little seems to have been done in the context of variable selection with reference to PCA.

Although the major aim of the current research project is concerned with the selection of variables in PCA, as a lead-in to this it was decided to carry out a study on the choice of the *number of* PCs (say, k) to be used in the selection process. This is the minimum number of PCs necessary for adequate representation of the data. As part of this study, we have proposed a new technique for choosing k based on *bootstrapping*. There were three major reasons for this. Firstly, the only techniques which seem to have a formal statistical justification available in the literature are the cross-validatory techniques and the technique based on Bartlett's test. However, not only the latter technique is based on distributional assumptions which are often unrealistic, but also it is based on the eigenvalues of the covariance or correlation matrix. Hence, the choice of k is unique to the covariance or correlation matrix used rather than the data at hand. Furthermore, it tends to retain more PCs than necessary in practice, and hence leaves us with the choice of only the cross-validatory techniques. Secondly, although the cross-validatory techniques are less *ad hoc* than those based on eigenvalues, *cross-validation* involves deletion of part of the data which can lead to choices of k which are sensitive to sampling error and therefore unreliable, particularly in cases where sample size is small. Thirdly, not only *bootstrapping* overcomes the problems above, but also it has been shown to out-perform *cross-validation* and other *resampling* plans, for example when used with multivariate methods such as *Discriminant Analysis*, where it is desired to find the best estimator for the *error rate* of *misclassification*.

The proposed method involved finding the *bootstrap* estimators (as opposed to the *cross-validatory* estimators in the case of Eastment and Krzanowski's technique) for the error sums of squares due to *not fitting* the last $p-k$ PCs, k = 1, 2, ..., etc. A suitable function of these estimators was then used to find the *optimum* value of k.

Given in Chapter 2 of this Thesis was a detailed description of the various techniques based on *eigenvalues* and those based on *cross-validation* for choosing the number of components in PCA and a review of some comparative studies of these methods that are available in the literature. Chapter 3 then described the proposed *bootstrap* based technique for choosing the number of components, while presented in Chapter 4 were the details of the Monte Carlo comparison of this new technique with the *cross-validatory* scheme of Eastment and Krzanowski (1982). These two techniques are referred to as the $\tilde{W}$ technique and the W technique, respectively. The data for this study were simulated using models 1 through 4 of Jolliffe (1972), who constructed these models in such a way that within each model, some variables are linear combinations of others except for random disturbances, and hence are redundant. The constructed variables make use of randomly generated variates which are *identically* and *independently* (*iid*) distributed as N(0,1). These new variables fall into groups and within which the variables are linear combinations of each other (plus random disturbances), while variables from different groups are independent. Hence, for each model, the *dimensionality* or the number of components needed to represent the data adequately were *known prior* to application of the $\tilde{W}$ technique and the W technique. These two techniques were compared on the basis of the percentage of the number of times each technique retains the correct (expected) number of components. Two sample sizes, large and small, were used for the training samples analysed in the simulation study. The aim was to see whether or not the size of the sample affects the performance of the two techniques. Furthermore, in the case of the W technique the number of bootstrap samples used were fixed at two levels, large and small. Here, the aim was to see whether or not the number of bootstrap samples is a factor in the performance of the $\tilde{W}$ technique. For each of the sampling considerations above, both *covariance* matrices and *correlation* matrices were used for PCA in conjuction with the $\tilde{W}$ technique and the W technique. The aim was to see whether or not the type of *variation* matrix used affects the performance of the two techniques.

The Monte Carlo study showed that, for choosing the number of components in PCA, $\tilde{W}$ is on average superior to W, providing the correct number of components with a very high probability. Although W provided the correct number of components satisfactorily for large samples, there was a tendency for the choice of k to be too small. This feature was more prominent when data sets were of small sample size. These results were consistent with the earlier studies by Eastment and Krzanowski (1982) and Krzanowski (1983). In fact, a similar experience has been reported by Wold (1978), in a limited Monte Carlo simulation study with the choice of the number of components according to the statistic R. On the other hand, it was noted from the simulation study in Chapter 4 that, $\tilde{W}$ generally chooses the correct number of components with a very high probability, although there is a slight tendency (much smaller than in the case of W) for the choice of k to be one component smaller than necessary for small samples and one component larger than necessary for large samples. The tendency for the choice of *k* to be too large was seen to be an advantage over its tendency to be too small, because with the former, all the essential components are retained, whereas with the latter some of the essential components may be deleted. The simulation study also showed that for a small number of bootstrap samples, $\tilde{W}$ takes slightly longer to be computed than W when samples are of small size; and approximately 1.3 times quicker than W to complete the analysis for large samples. For a large number of bootstrap samples, $\tilde{W}$ is generally slower than W. However, the study also showed that there is very little and in some cases no improvement in the performance of $\tilde{W}$ due to an increase in the number of bootstrap samples. Hence, the computational time for $\tilde{W}$ can be kept at a minimum by using a considerably small number of bootstrap samples without effectively decreasing its performance. The study also showed that, there is a very little change in the performance of the $\tilde{W}$ technique and the W technique due to the type of *variation* matrix used for PCA. The advantages of $\tilde{W}$ above, over W (and R) led us to strongly recommend it as the criteria for choosing the number of components in PCA.

Chapters 5 - 7 dealt with variable selection in *Principal Component Analysis*. As can be seen in the literature, the *only* publications which seem to be available in the area of variable selection with reference to PCA are Jolliffe (1972, 1973, 1986), McCabe (1984) and Krzanowski (1987). The methods for variable selection in PCA suggested by Jolliffe (1972, 1973) and McCabe (1984) are mainly based on *Eigen_analysis* and they satisfy various optimality criteria, but they consistently fail to retain the suitable subsets of the original variables which preserve the overall features of the complete data. One possible reason for the failure of Jolliffe's and McCabe's methods to recover the *multivariate structure* of the complete data is that, they are concerned with the overall features, either of the complete data (in the case of Jolliffe (1972, 1973)) or of the subset data (in the case of McCabe (1984)). Thus, their criteria are based exclusively on covariance or correlation matrices and their eigenvalues or eigenvectors. A suitable criterion for preserving the overall structure present in the entire set of variables would involve comparisons between the configuration which arises from the PCA of the complete data and that which arises from the PCA of the subset data. We termed such criteria as measures of *Multivariate Association* (MVA). Krzanowski (1987) proposed such a criterion (termed M^2-*procrustes* criterion) based on *Procrustes Analysis* which attempts to preserve *multivariate data structure* by preserving structure among individual data points. While this criterion successfully identifies the *structure-bearing* variables, the results are optimal particularly when groups are present in the data. Hence, our major aim in the current research project was to utilize the idea of *multivariate association* to select subsets of variables which preserve *any* (*unknown*) multivariate structure that may be present in the data. Having reviewed the literature on variable selection in PCA, we proposed *four* methods for selecting variable in Chapter 6 and compared these among themselves as well as with the M^2-*procrustes* criterion in Chapter 7.

Two of the proposed methods, termed $\hat{\gamma}_5$ and $\hat{\gamma}_6$ were based on *canonical correlations*, while the other two, termed δ and δ^* were based on *Euclidean* distances between individual data points and some ideas from *Graph Theory*. The criteria $\hat{\gamma}_5$ and $\hat{\gamma}_6$ chose those subsets of variables for which functions of the *canonical correlations* between the subset configuration and the complete data configuration are *maximal*. The criteria δ and δ^*, on the other hand, chose those subsets of variables for which the discrepancy between corresponding *interpoint-distances* in the two configurations is minimized under *Graph-Theoretic* constraints.

In order to perform the variable selection, each of the criteria above was incorporated into the *backward elimination* procedure and the variables whose omission *optimized* the criterion under investigation were deleted from the system in a *sequential* manner until the required number of variables was reached. It was noted that, since k components may be sufficient to model the 'signal' in the data, the remaining $p-k$ dimensions are a reflection of the 'noise'. Hence, q, the number of variables to retain is set equal to k. Although the *backward elimination* procedure was used throughout the Thesis, it was decided to compare the Monte Carlo results based on this procedure with those based on the *stepwise selection* procedure in Chapter 7. The aim for this was to see whether or not using the *stepwise selection* procedure (which is computationally slower) improves the performance of the various criteria when used for variable selection in PCA.

The data for the Monte Carlo study in Chapter 7 were simulated using models 1 - 4 of Jolliffe (1972). Note that these were exactly the same models used to asses the performance of the $\tilde{W}$ technique and the W technique for choosing the number of components in PCA. As noted earlier, Jolliffe constructed these models in such a way that the variables fall into groups within which variables are linear combinations of others (except for random disturbances) and hence are redundant, while variables from different groups are uncorrelated. This way, not only the *dimensionality* or the number of components to use with each criterion, and hence the number of variables to retain were *known prior* to variable selection, but also the subsets of variables which satisfactorily define this dimensionality were known. In order to conduct the Monte Carlo study, the size of the sample was fixed at three levels, large, moderate and small. The aim was to discover any change of pattern in the behaviour of the various criteria due to sample size. Both the *covariance* matrices and the *correlation* matrices were used for PCA in conjuction with all the five variable selection criteria M^2, $\hat{\gamma}_5$, $\hat{\gamma}_6$, δ and δ^* to see whether or not the type of *variation* matrix used affects their performance.

The Monte Carlo study found that, in general, all the five variable selection methods perform satisfactorily when samples are of large size. Furthermore, the methods generally perform better for correlation matrices used to perform the PCA than for covariance matrices. However, the performance deteriorates as sample size decreases, with M^2 being the worst method for selecting variables in PCA when samples are of small size. δ becomes the best method in this case, followed very closely by $\hat{\gamma}_5$ and $\hat{\gamma}_6$. In fact, the study showed that, while in some cases $\hat{\gamma}_5$ and $\hat{\gamma}_6$ behave identically, there is a consistent similarity between their behaviour and that of the criterion δ. While in general, the criterion δ^* showed the poorest performance, there is reason to believe (from its derivation) that, like M^2, it could successfully identify *structure-bearing* variables for *grouped* data as it attempts to preserve 'local structure'. This however, obviously needs further investigation. It seemed that $\hat{\gamma}_5$, $\hat{\gamma}_6$ and δ are the most suitable methods for our aim to select subsets of variables which will preserve the overall *multivariate structure* of the data in a PCA. Although none of these three methods was uniformly the best, δ may be recommended as the overall 'best' choice.

Regarding the *optimal sequential* approach in selecting subsets of variables, the study showed that substitution of the *backward* *elimination* procedure with *stepwise selection* generally leaves the performance of *all* the criteria unaltered. However, if the data are simulated in such a way that *three* or more variables are highly intercorrelated, use of the stepwise selection procedure showed a considerable improvement in the performance of $\hat{\gamma}_5$, $\hat{\gamma}_6$ and M^2 when correlation matrices were used to perform the PCA. Hence, in this case, these criteria were better choices for preserving the overall multivariate data structure than δ and δ^*; with $\hat{\gamma}_5$ and $\hat{\gamma}_6$ performing almost equivalently but better than M^2. On the other hand, when covariance matrices were used to perform the PCA, the performance of M^2 was virtually the same as when the backward elimination procedure was used to select the variables, while the performance of the γ-measures was slightly worse.

In terms of the amount of computational time needed to implement the various criteria, M^2, $\hat{\gamma}_5$, and $\hat{\gamma}_6$ were shown to be more practicable than δ and δ^*, and M^2 was the fastest. However, it was also seen that δ and δ^* can also be used in practice, particularly for small data sets.

Although the conclusions from our simulation studies seem to be prominent and clear, the behaviour of the $\tilde{W}$ technique for choosing *k*, the number of components and the various variable selection methods should be confirmed by further wide-spread studies. A further study on the behaviour of $\tilde{W}$ could consider different types of structure for the *covariance* and *correlation* matrices, for example, with variables having identical or different variances and correlation coefficients. Incorporating such structure into simulated data would cast some light on the sampling distribution of $\tilde{W}$ and possibly lead to setting up a formal significance test to decide on *k*. There are two sources of variability to be considered in constructing such a distribution, namely the variability due to different sample covariance matrices for a fixed population covariance matrix, and the variability due to the fact that a fixed sample covariance matrix can result from different data matrices. For each source of variability, there is a large number of parameters that can be varied, such as the sample size, the number of variables, and particularly, the structure of the covariance matrix.

In the current research project, only the 'ordinary' bootstrap methodology was considered in computing the statistic $\tilde{W}$ for choosing k. Future investigations could make use of one of the recent developments, the *balanced bootstrap*. Suppose we wish to draw b bootstrap samples. In the 'ordinary' bootstrap, each datum $\underline{x}_i$ (i = 1, 2, ..., n) may not appear equally often in the aggregate of all b bootstrap samples. The 'balanced bootstrapping' attempts to correct this deficiency, and has been shown to have the effect of reducing the probable error in the variance due to bootstrapping. Furthermore, the balanced bootstrap has been shown to perform better than the 'ordinary' bootstrap when used with multivariate methods such as *Discriminant Analysis*, where it is desired to find the best estimator for the *error rate* of *misclassification*. Further discussions of the balanced bootstrap can be found in Gleason (1988), Graham *et al.* (1990) and Hall (1990a, 1990b).

A further study on the various criteria for variable selection in PCA could involve incorporating 'extra' multivariate structure into models similar to those of Jolliffe (1972), for example, *curve patterns*, *group-structure*, and so on. This would be an attempt to further confirm that the criteria, particularly the proposed criteria, choose subsets of the original variables which will preserve whatever *unknown* multivariate structure that may be present in the data. Certainly, further investigation on the criterion δ^* is needed to confirm our suspicion that, like M^2, this criterion could be robust towards preserving *group-structure*.

Finally, the application of the techniques $\tilde{W}$ and W for choosing the number of components and the various criteria for variable selection to several different real data sets was made in Chapter 8. These analyses confirmed the conclusions made from the Monte Carlo simulation study, that the various variable selection criteria choose subsets of variables to preserve the overall features of the complete data. When used with the choice of k according to W, the criteria $\hat{\gamma}_5$, $\hat{\gamma}_6$, δ and δ^* selected subsets for which interesting multivariate patterns other than *group-structure* were revealed for one of the data sets. When used with the choice of k according to $\tilde{W}$, *all* the criteria including M^2 chose variable subsets for which *group-structure* was enhanced. This has implications for other multivariate techniques such as *Discriminant Analysis*, *Cluster Analysis* and *Projection Pursuit*. The objective in *Discriminant Analysis* is to classify future individuals into *known groups*, while in *Cluster Analysis* and *Projection Pursuit* the objective is to find *groups* or *outliers* in the data. In *Discriminant Analysis*, the practitioner is often interested in choosing subsets of the original variables for which the *error rate* of *misclassification* is *minimal*, while in *Cluster Analysis*, interest is often placed in those variables which can be used to distinguish between clusters. Hence, the approach of the present study may be contrasted with these selection exercises. However, our objective in the current research project was to identify subsets of variables which carry whatever *unknown* structure that may be present in the data, and in an attempt to address this problem we have proposed several highly-promising criteria for variable selection in PCA which seem to be excellent alternatives to Krzanowski's (1987)

M^2-*procrustes* criterion. Furthermore, as indicated in this Chapter, there is an overwhelming scope for further investigation of the proposed methods and criteria together with some new ideas in the area of *variable selection* in *Principal Component Analysis*.

APPENDIX A

Listed in this Appendix are the lists of variables of the data sets described in Chapter 8, together with their covariance and correlation matrices.

Data 1: U.S.A. 16-19 year old male unemployment data

Measurements of monthly unemployment (1948-1981)

X1	:	January
X2	:	February
X3	:	March
X4	:	April
X5	:	May
X6	:	June
X7	:	July
X8	:	August
X9	:	September
X10	:	October
X11	:	November
X12	:	December

Covariance matrix

x 10^6

	X1	X2	X3	X4
X1	62277.0588	62742.9964	58452.9002	53604.3262
X2	62742.9964	64672.4323	59810.9519	54538.0160
X3	58452.9002	59810.9519	56719.3191	51983.4492
X4	53604.3262	54538.0160	51983.4492	49043.0018
X5	47156.8485	47921.0758	45897.3030	44007.3030
X6	66157.1587	66941.3476	64892.7629	61649.3013
X7	65597.8324	66838.2861	63800.2531	59795.1480
X8	56048.2014	57095.2968	54305.7950	50668.5936
X9	56433.8841	57635.2077	54435.4510	50480.9608
X10	57155.7398	57947.5553	54858.7897	51089.2317
X11	61569.1979	62633.1533	59093.6257	54790.5187
X12	57656.8467	58467.2540	55378.8699	51416.6595

	X5	X6	X7	X8
X1	47156.8485	66157.1587	65597.8324	56048.2014
X2	47921.0758	66941.3476	66838.2861	57095.2968
X3	45897.3030	64892.7629	63800.2531	54305.7950
X4	44007.3030	61649.3013	59795.1480	50668.5936
X5	42119.2273	56827.5152	53617.3485	45635.8636
X6	56827.5152	85620.1533	79376.4278	65552.9519
X7	53617.3485	79376.4278	78573.4768	64484.2219
X8	45635.8636	65552.9519	64484.2219	54752.9207
X9	44976.5606	65062.7665	64750.5481	54136.0169
X10	45479.9242	66936.8592	66647.2184	55408.2112
X11	48785.0000	70950.6328	70364.5686	58850.2781
X12	45428.9845	66249.3102	66141.6961	55215.7620

	X9	X10	X11	X12
X1	56433.8841	57155.7398	61569.1979	57656.8467
X2	57635.2077	57947.5553	62633.1533	58467.2540
X3	54435.4510	54858.7897	59093.6257	55378.8699
X4	50480.9608	51089.2317	54790.5187	51416.6595
X5	44976.5606	45479.9242	48785.0000	45428.9849
X6	65062.7665	66936.8592	70950.6328	66249.3102
X7	64750.5481	66647.2184	70364.5686	66141.6961
X8	54136.0169	55408.2112	58850.2781	55215.7620
X9	55187.3628	55989.0989	60066.9216	56387.1114
X10	55989.0989	58191.7763	61754.0998	58109.9064
X11	60066.9216	61754.0998	67080.4920	62485.8039
X12	56387.1114	58109.9064	62485.8039	59539.6800

Correlation matrix

	X1	X2	X3	X4	X5	X6
X1	1.0000	0.9886	0.9835	0.9699	0.9207	0.9060
X2	0.9886	1.0000	0.9875	0.9684	0.9182	0.8996
X3	0.9835	0.9875	1.0000	0.9856	0.9390	0.9312
X4	0.9699	0.9684	0.9856	1.0000	0.9683	0.9514
X5	0.9207	0.9182	0.9390	0.9683	1.0000	0.9463
X6	0.9060	0.8996	0.9312	0.9514	0.9463	1.0000
X7	0.9378	0.9376	0.9557	0.9633	0.9320	0.9678
X8	0.9598	0.9595	0.9745	0.9778	0.9503	0.9574
X9	0.9626	0.9647	0.9730	0.9703	0.9329	0.9465
X10	0.9494	0.9446	0.9549	0.9563	0.9186	0.9483
X11	0.9526	0.9509	0.9580	0.9553	0.9178	0.9362
X12	0.9469	0.9422	0.9530	0.9515	0.9072	0.9279

	X7	X8	X9	X10	X11	X12
X1	0.9378	0.9598	0.9626	0.9494	0.9526	0.9469
X2	0.9376	0.9595	0.9647	0.9446	0.9509	0.9422
X3	0.9557	0.9745	0.9730	0.9549	0.9580	0.9530
X4	0.9633	0.9778	0.9703	0.9563	0.9553	0.9515
X5	0.9320	0.9503	0.9329	0.9186	0.9178	0.9072
X6	0.9678	0.9574	0.9465	0.9483	0.9483	0.9279
X7	1.0000	0.9831	0.9833	0.9856	0.9692	0.9670
X8	0.9831	1.0000	0.9848	0.9816	0.9711	0.9671
X9	0.9833	0.9848	1.0000	0.9880	0.9872	0.9837
X10	0.9856	0.9816	0.9880	1.0000	0.9884	0.9872
X11	0.9692	0.9711	0.9872	0.9884	1.0000	0.9887
X12	0.9670	0.9671	0.9837	0.9872	0.9887	1.0000

Data 2 and *data 3*: **Venezuela data**

Measurements of English subject units on Venezuela students

X1 : Comprehensive score
X2 : Essay score
X3 : Cloze score, with acceptable alternatives ('acceptable')
X4 : Cloze score, alternatives not accepted ('exact')
X5 : Structure test score
X6 : Dictation test score
X7 : Spanish cloze score ('acceptable')
X8 : Spanish cloze score ('exact')

Data 2: **College Carlett only**

Covariance matrix

	X1	X2	X3	X4	X5	X6	X7	X8
X1	10.8097	8.2713	4.6883	4.5324	8.5243	11.2409	7.5526	6.3968
X2	8.2713	24.6208	16.1700	13.1147	23.5553	24.8104	9.9649	6.0061
X3	4.6883	16.1700	16.4008	12.3381	16.4575	19.7955	8.9474	5.6215
X4	4.5324	13.1147	12.3381	10.1957	14.4055	16.0310	8.0439	5.6296
X5	8.5243	23.5553	16.4575	14.4055	32.6055	27.1856	10.2982	5.9393
X6	11.2409	24.8104	19.7955	16.0310	27.1856	43.9447	16.0614	10.7794
X7	7.5526	9.9649	8.9474	8.0439	10.2982	16.0614	15.3860	13.9211
X8	6.3968	6.0061	5.6215	5.6296	5.9393	10.7794	13.9211	15.6923

Correlation matrix

	X1	X2	X3	X4	X5	X6	X7	X8
X1	1.0000	0.5070	0.3521	0.4317	0.4541	0.5158	0.5856	0.4911
X2	0.5070	1.0000	0.8047	0.8278	0.8314	0.7543	0.5120	0.3056
X3	0.3521	0.8047	1.0000	0.9541	0.7117	0.7374	0.5632	0.3504
X4	0.4317	0.8278	0.9541	1.0000	0.7901	0.7574	0.6422	0.4451
X5	0.4541	0.8314	0.7117	0.7901	1.0000	0.7182	0.4598	0.2626
X6	0.5158	0.7543	0.7374	0.7574	0.7182	1.0000	0.6177	0.4105
X7	0.5856	0.5120	0.5632	0.6422	0.4598	0.6177	1.0000	0.8959
X8	0.4911	0.3056	0.3504	0.4451	0.2626	0.4105	0.8959	1.0000

Data 3: **Colleges Oldham, Cheshire and Carlett**

Covariance matrix

	X1	X2	X3	X4	X5	X6	X7	X8
X1	13.2517	6.9566	8.0842	6.2434	9.9524	11.6967	6.6549	6.4451
X2	6.9566	23.0721	14.0500	10.1826	22.0160	23.0024	8.9755	5.8757
X3	8.0842	14.0500	17.0439	12.2595	17.5306	19.2268	9.5208	7.4921
X4	6.2434	10.1826	12.2592	9.7835	12.9875	14.3727	7.1668	5.7790
X5	9.9524	22.0160	17.5306	12.9875	33.5198	27.4200	10.9230	7.3064
X6	11.6967	23.0024	19.2268	14.3727	27.4200	45.4077	12.8615	7.4007
X7	6.6549	8.9755	9.5208	7.1668	10.9230	12.8615	13.9749	12.2903
X8	6.4451	5.8757	7.4921	5.7790	7.3064	7.4007	12.2903	13.6416

Correlation matrix

	X1	X2	X3	X4	X5	X6	X7	X8
X1	1.0000	0.3978	0.5379	0.5483	0.4722	0.4768	0.4890	0.4794
X2	0.3978	1.0000	0.7085	0.6777	0.7917	0.7107	0.4999	0.3312
X3	0.5379	0.7085	1.0000	0.9494	0.7334	0.6911	0.6169	0.4913
X4	0.5483	0.6777	0.9494	1.0000	0.7172	0.6819	0.6129	0.5002
X5	0.4722	0.7917	0.7334	0.7172	1.0000	0.7028	0.5047	0.3417
X6	0.4768	0.7107	0.6911	0.6819	0.7028	1.0000	0.5106	0.2974
X7	0.4890	0.4999	0.6169	0.6129	0.5047	0.5106	1.0000	0.8901
X8	0.4794	0.3312	0.4913	0.5002	0.3417	0.2974	0.8901	1.0000

APPENDIX B

The various computer programs corresponding to the techniques discussed in this thesis were written in the matrix language programming software GAUSS (1988), and the lists of command statements for these programs are located in the pocket at the back of the thesis. Also presented are the lists of command statements for the GAUSS programs used to conduct various other studies essential to the thesis, for example, the program used to correct Jolliffe's (1972) classification of variable subsets for model 1 - 4 described in Table 4.1.

REFERENCES

Anderson, T.W. (1958). *An Introduction to Multivariate Statistical Analysis*. John Wiley and Sons, Inc., New York.

Andrews, D.F., and Herzberg, A.M. (1985). *Data: A Collection of Problems from Many Fields for the Student and Research Worker.* Springer-Verlag, New York.

Bartlett, M.S. (1950). Tests of Significance in Factor Analysis. *British Journal of Psychology, Statistics Section*, **3**, 77-85.

Beale, E.M.L., Kendall, M.G. and Mann, D.W. (1967). Discarding Variables in Multivarite Analysis. *Biometrika*, **54**, 357-366.

Bolch, B.W. and Huang, C.J. (1974). *Multivariate Statistical Methods for Business and Economics*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Bose, A. (1988). Edgeworth Correction by Bootstrap in Autoregressions. *The Annals of Statistics*, **16**, 1709-1722.

Brook, R.J. and Arnold, G.C. (1985). *Applied Regression Analysis and Experimental Design*. New York: Marcel Dekker, Inc.

Bunch, J.R. and Nielsen, C.P. (1978). Updating the Singular Value Decomposition. *Numerische Mathematik*, **31**, 111-129.

Bunch, J.R., Nielsen, C.P. and Sorensen, D.C. (1978). Rank One Modification of the Eigenproblem. *Numerische Mathematik*, **31**, 31-48.

Bunke, O., and Droge, B. (1984). Bootstrap and Cross-Validation Estimates of the Prediction Error for Linear Regression Models. *The Annals of Statistics*, **12**, 1400-1424.

Campbell, N.A. and Atchley, W.R. (1981). The Geometry of Canonical Variate Analysis. *Systimatic Zoology*, **30**, 268-280.

Cattell, R.B. (1966). The Scree Test for the Number of Factors. *Journal of Multivariate Behavioural Research*, **1**, 245-276.

Chatfield, C., and Collins, A.J. (1986). *Introduction to Multivariate Analysis*. Chapman and Hall, London.

Cohen, J. (1982). Set Correlation as a General Multivariate Data-Analytic Method. *Multivariate Behavioural Research*, **17**, 301-341.

Costanza, M.C., and Afifi, A.A. (1979). Comparison of Stopping Rules in Forward Stepwise Discriminant Analysis. *Journal of the American Statistical Association*, **74**, 777-785.

Coxhead, P. (1974). Measuring the Relationship Between two Sets of Variables. *British Journal of Mathematical and Statistical Psychology*, **27**, 205-212.

Cramer, E.M. and Nicewander, W.A. (1979). Some Symmetric, Invariant Measures of Multivariate Association. *Psychometrika*, **44**, 43-54.

Daudin, J.J. (1986). Selection of Variables in Mixed-Variable Discriminant Analysis. *Biometrika*, **42**, 473-481.

De Wet, T., and Van Wyk, J.W.J. (1986). Bootstrap Confidence Intervals for Regression Coefficients When the Residuals Are Dependent. *Journal of Statistical Computation and Simulation*, **23**, 317-327.

Dillon, W.R., and Goldstein, M. (1984). *Multivariate Analysis: Methods and Applications*. John Wiley and Sons, Inc., New York.

Dolker, M., Halperin, S. and Divgi, D.R. (1982). Problem With Bootstrapping Pearson Correlations in Very Small Bivariate Samples. *Psychometrika*, **47**, 529-530.

Draper, N.R. and Smith, H. (1966). *Applied Regression Analysis*. New York: Wiley.

Dunteman, G.H. (1989). *Principal Component Analysis*. Sage Publications, Inc., California.

Eastment, H.T. and Krzanowski, W.J. (1982). Cross-Validatory Choice of the Number of Principal Components from a Principal Component Analysis. *Technometrics*, **24**, 73-77.

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, **7**, 1-26.

Efron, B. (1981). Nonparametric Estimates of Standard Error: the Jackknife, the Bootstrap and Other Methods. *Biometrika*, **68**, 589-599.

Efron, B. (1982). *The Jackknife, The Bootstrap and Other Resampling Plans*, SIAM-CBMS Monograph **38**. Philadelphia: S.I.A.M.

Efron, B. (1985). Bootstrap Confidence Intervals for a Class of Parametric Problems. *Biometrika*, **72**, 45-58.

Efron, B. (1990). More Efficient Bootstrap Computations. *Journal of the American Statistical Association*, **85**, 79-89.

Efron, B. and Gong, G. (1983). A Leisurely Look at the Bootstrap, the Jackknife and Cross-Validation. *The American Statistician*, **37**, 36-48.

Efron, B. and Tibshirani, R. (1986). Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy. *Statistical Science*, **1**, 54-77.

Fisher, N. I., and Hall, P. (1989). Bootstrap Confidence Regions for Directional Data. *Journal of the American Statistical Association*, **84**, 996-1002.

Fisher, N.I., and Hall, P. (1990). Bootstrap Algorithms for Small Samples. Unpublished Manuscript.

Fisher, R.A. (1936). The Use of Multiple Measurements in Taxonomic Problems. *The Annals of Eugenics*, **7**, 179-188.

Freedman, D. (1981). Bootstrapping Regression Models. *The Annals of Statistics*, **9**, 1218-1228.

Freedman, D.A. and Peters, S.C. (1984). Bootstrapping a Regression Equation: Some Empirical Results. *Journal of the American Statistical Association*, **79**, 97-106.

Friedman, J.H. and Rafsky, L.C. (1979). Multivariate Generalizations of the Wald-Wolfowitz and Smirnov Two-Sample Tests. *The Annals of Statistics*, **7**, 697-717.

Friedman, J.H. and Rafsky, L.C. (1983). Graph-Theoretic Measures of Multivariate Association and Prediction. *The Annals of Statistics*, **11**, 377-391.

Furnival, G. and Wilson, R. (1974). Regression by Leaps and Bounds. *Technometrics*, **16**, 499-511.

Gabriel, K.R. (1978). Least Squares Approximation of Matrices by Additive and Multiplicative Models. *Journal of the Royal Statistical Society B*, **40**, 186-196.

Ganeshanandam, S. (1987). *Variable Selection in Two-Group Discriminant Analysis Using The Linear Discriminant Function.* Unpublish PhD. thesis, University of Reading, UK.

Ganeshanandam, S. and Krzanowski, W.J. (1989). On Selecting Variables and Assessing Their Performance in Discriminant Analysis. *Australian Journal of Statistics vol 3*, **31**, 433-447.

GAUSS (1988). *The GAUSS System.* Version 2.0. Aptech Systems, Inc. Kent, Washington.

Gittins, R. (1985). *Canonical Analysis.* Springer-Verlag Berlin Heidelberg, Tokyo.

Gleason, J. R. (1988). Algorithm for Balanced Bootstrap Simulation. *The American Statistician*, **42**, 263-265.

Gong, G. (1986). Cross-Validation, the Jackknife, and the Bootstrap: Excess Error Estimation in Forward Logistic Regression. *Journal of the American Statistical Association*, **81**, 108-113.

Gower, J.C. (1966). Some Distance Properties of Latent Root and Vector Methods Used in Multivariate Analysis. *Biometrika*, **53**, 325-338.

Gower, J.C. (1971). Statistical Methods of Comparing Different Multivariate Analyses of the Same Data. In *Mathematics in the Archaeological and Historical Sciences* (F.R. Hodson, D.G. Kendall and P. Tautu, eds), pp. 138-149. Edinbugh: University Press.

Gower, J.C. (1975). Generalized Procrustes Analysis. *Psychometrika*, **40**, 33-51.

Good, I.J. (1969). Some applications of the Singular Value Decomposition of a matrix. *Technometrics*, **11**, 823-831.

Graham, R.L., Hinkley, D.V., John P.W.M. and Shi, S. (1990). Balanced Design of Bootstrap Simulations. *Journal of the Royal Statistical Society, Series B*, **52**, 185-202.

Green, B.F. (1952). The Orthogonal Approximation of an Oblique Structure in Factor Analysis. *Psychometrika*, **17**, 429-440.

Hall, P. (1990a). Performance of Bootstrap Balanced Resampling in Distribution Function and Quantile Problems. *Probability Theory and Related Fields*.

Hall, P. (1990b). Balanced Importance Resampling for the Bootstrap. *Canberra Statistics Technical Report*, CSTR-022-90, SMS-054-90.

Hall, P., Martin, M.A. and Schucany, W.R. (1989). Better Nonparametric Bootstrap Confidence Intervals for the Correlation Coefficient. *Journal of Statistical Compution and Simulation*, **33**, 161-172.

Hall, P. and Hart, J.D. (1990). Bootstrap Test for Difference Between Means in Nonparametric Regression. *Journal of the American Statistical Association*, **85**, 1039-1049.

Hardle, W. and Bowman, A. (1988). Bootstrapping in Nonparametric Regression: Local Adaptive Smoothing and Confidence Bands. *Journal of American Statistical Association*, **83**, 102-110.

Hardle, W. and Marron, J.S. (1990). Semiparametric Comparison of Regression Curves. *The Annals of Statistics*, **18**, 63-89.

Hocking, R.R. (1976). The Analysis and Selection of Variables in Linear Regression. *Biometrics*, **32**, 1-50.

Hotelling, H. (1936). Relations Between Two Sets of Variates. *Biometrika*, **28**, 57-71.

Hotelling, H. (1957). The relations of the Newer Multivariate Statistical Methods to Factor Analysis. *British Journal of Statistical Psychology*, **10**, 69-79.

Hurley, J.R. and Cattell, R.B. (1962). Producing Direct Rotation to Test a Hypothesized Factor Structure. *Behavioural Science*, **7**, 258-262.

Jeffers, J.N.R. (1967). Two Case Studies in the Application of Principal Component Analysis. *Applied Statistics*, **16**, 225-236.

Jolliffe, I.T. (1970). Redundant Variables in Multivariate Analysis. Unpublished PhD. thesis, University of Sussex.

Jolliffe, I.T. (1972). Discarding Variables in a Principal Component Analysis, I: Artificial Data. *Applied Statistics*, **21**, 160-173.

Jolliffe, I.T. (1973). Discarding Variables in a Principal Component Analysis, II: Real Data. *Applied Statistics*, **22**, 21-31.

Jolliffe, I.T. (1986). *Principal Component Analysis*. Springer-Verlag, New York.

Knapp, T.R. (1978). Canonical Correlation Analysis: A General Parametric Significance-Testing System. *Psychological Bulletin*, **85**, 410-416.

Krishnaiah, P.R. (1982). Selection of Variables Under Regression Models. *Handbook of Statistics, vol. 2* (P.R. Krishnaiah and I.N. Kanal, eds.) pp. 805-820. North Holland Publishing Company.

Krzanowski, W.J. (1971). A Comparison of Some Distance Measures Applicable to Multinomial Data, Using a Rotational Fit Technique. *Biometrics*, **27**, 1062-1068.

Krzanowski, W.J. (1979). Between-Group Comparison of Principal Components. *Journal of the American Statistical Association*, **74**, 703-707.

Krzanowski, W.J. (1982). Between-Group Comparison of Principal Components - Some Sampling Results. *Journal of Statistical Computation and Simulation*, **15**, 141-154.

Krzanowski, W.J. (1983). Cross-Validatory Choice in Principal Component Analysis: Some Sampling Results. *Journal of Statistical Computation and Simulation*, **18**, 299-314.

Krzanowski, W.J. (1987). Variable Selection to Preserve Multivariate Data Structure, Using Principal Components. *Applied Statistics*, **36**, 22-33.

Krzanowski, W.J. (1988). *Principles of Multivariate Analysis: A User's Perspective*. Oxford University Press, New York.

Levine, M.S. (1977). *Canonical Analysis and Factor Comparison*. Sage Publications, Inc., Carlifornia.

Lunneborg, C.E. (1985). Estimating the Correlation Coefficient: The Bootstrap Approach. *Psychological Bulletin*, **98**, 209-215.

Mammen, E. (1989). Asymptotic With Increasing Dimension for Robust Regression With Application to the Bootstrap. *Annals of Statistics*, **17**, 382-400.

Mandel, J. (1972). Principal Components, Analysis of Variance and Data structure. *Statistica Neerlandica*, **26**, 119-129.

Mandel, J. (1982). Use of the Singular Value Decomposition in Regression Analysis. *The American Statistician*, **36**, 15-24.

Mardia, K.V., Kent, J.T. and Bibby, J.M. (1979). *Multivariate Analysis*. Academic Press, London.

McCabe, G.P. (1984). Principal Variables. *Technometrics*, **26**, 137-144.

McKay, R.J. (1976). Simultaneous Procedures in Discriminant Analysis Involving Two Groups. *Technometrics*, **18**, 47-53.

McKay, R.J. (1977). Variable Selection in Multivariate Regression: An Application of Simultaneous Test Procedures. *Journal of the Royal Statistical Society, Ser. B*, **39**, 371-380.

McKay, R.J. (1979). The Adequacy of Variable Subsets in Multivariate Regression. *Technometrics*, **21**, 475-479.

McKay, R.J. and Campbell, N.A. (1982a). Variable Selection Techniques in Discriminant Analysis. I. Description. *British Journal of Mathematical and Statistical Psychology*, **35**, 1-29.

McKay, R.J. and Campbell, N.A. (1982b). Variable Selection Techniques in Discriminant Analysis. II. Allocation. *British Journal of Mathematical and Statistical Psychology*, **35**, 30-41.

McLachlan, G.J. (1979). A Criterion for Selecting Variables for the Linear Discriminant Function. *Biometrics*, **32**, 529-534.

McLachlan, G.J. (1980). On the Relationship Between the F Test and the Overall Error Rate for Variable Selection in Two-Group Discriminant Analysis. *Biometrics*, **36**, 501-510.

McLachlan, G.J. (1987). On Bootstrapping the Likelihood Ratio Test Statistic for the Number of Components in a Normal Mixture. *Applied Statistics*, **36**, 318-324.

McReynolds, W.O. (1970). Characterization of Some Liquid Phases. *Journal of Chromatographical Sciences*, **8**, 685-691.

Mosier, C.I. (1939). Determining a Simple Structure When Loadings for Certain Tests are Known. *Psychometrika*, **4**, 149-162.

Muller, K.E. (1981). Relationships Between Redundancy Analysis, Canonical Correlation, and Multivariate Regression. *Pychometrika*, **46**, 139-142.

Muller, K.E. (1982). Understanding Canonical Correlation Through the General Linear Model and Principal Components. *The American Statistician*, **36**, 342-354.

Murray, G.D. (1977). A Cautionary Note on Selection of Variables in Discriminant Analysis. *Applied Statistics*, **26**, 246-250.

Narayanaswamy, C.R. and Raghavarao, D. (1990). Principal Component Analysis of Large Dispersion Matrices. *Applied Statistics*, *2*, **40**, 309-316.

Navidi, W. (1989). Edgeworth Expansions for Bootstrapping Regression Models. *The Annals of Statistics*, **17**, 1472-1478.

Nesselroade, J.R. and Baltes, P.B. (1970). On a Dilemma of Comparative Factor Analysis: A Study of Factor Matching Based on Random Data. *Educational and Psychological Measurement*, **30**, 935-948.

Orheim, ALV (1981). Personal Communication with McCabe (1984).

Overall, J.E. (1974). Marker Variable Factor Analysis: A Regional Principal Axes Solution. *Multivariate Behavioural Science*, **9**, 149-164.

Peay, E.R. (1988). Multidimensional Rotation and Scaling of Configurations to Optimal Agreement. *Psychometrika*, **53**, 199-208.

Rasmussen, J.L. (1987). Estimating Correlation Coefficients: Bootstrap and Parametric Approach. *Psychological Bulletin*, **101**, 136-139.

Rice, J. (1984). Bandwidth Choice for Nonparametric Regression. *The Annals of Statistics*, **12**, 1215-1230.

Rocke, D.M. (1989). Bootstrap Bartlett Adjustment in Seemingly Unrelated Regression. *Journal of the American Statistical Association*, **84**, 598-601.

Rozeboom, W.W. (1965). Linear Correlation Between Sets of Variables. *Psychometrika*, **30**, 57-71.

SAS (1990). *SAS Users Guide*. Version 6. SAS Institut Inc., Cary NC, USA.

Schönemann, P.H. (1966). A Generalized Solution of the Orthogonal Procrustes Problem. *Psychometrika*, **31**, 1-10.

Schönemann, P.H. (1968). On Two-Sided Orthogonal Procrustes Problems. *Psychometrika*, **33**, 19-33.

Schönemann, P.H. and Carroll, R.M. (1970). Fitting One Matrix to Another Under Choice of Central Dilation and a Rigid Motion. *Psychometrika*, **35**, 245-255.

Seber, G.A.F. (1984). *Multivariate Observations*. John Wiley and Sons, Inc., New York.

Serlin, R.C. (1982). A Multivariate Measure of Association Based on the Pillai-Bartlett Procedure. *Psychological Bulletin*, **91**, 413-417.

Sibson, R. (1978). Studies in the Robustness of Multidimensional Scaling: Procrustes Statistics. *Journal of the Royal Statistical Society*, **40**, 234-238.

Snapinn, S.M. and Knoke, J.D. (1989). Estimation of Error Rates in Discriminant Analysis With Selection of Variables. *Biometrics*, **45**, 289-299.

Stine, R.A. (1985). Bootstrap Prediction Intervals for Regression. *Journal of the American Statistical Association*, **80**, 1026-1031.

Stone, M. (1974). Cross-Validatory Choice and Assessment of Statistical Predictions (with discussion). *Journal of Royal Statistical Society, B*, **36**, 111-148.

Strube, M.J. (1988). Bootstrap Type I Error Rate for the Correlation Coefficient: An Examination of Alternate Procedures. *Psychological Bulletin*, **104**, 290-292.

Tutsuoka, M.M. (1971). *Multivariate Analysis*. John Wiley and Sons, Inc., New York.

Thomas, L.A. and Schucany, W.R. (1990). Bootstrap Prediction Intervals for Autoregression. *Journal of the American Statistical Association*, **85**, 486-492.

van den Burg, W. and Lewis, C. (1988). Some Properties of Two Measures of Multivariate Association. *Psychometrika*, **53**, 109-122.

Van Ness, J.W. (1979). On the Effects of Dimension in Discriminant Analysis for Unequal Covariance Populations. *Technometrics*, **21**, 119-127.

Wold, S. (1976). Pattern Recognition by Means of Disjoint Principal Component Models. *Pattern Recognition*, **8**, 127-139.

Wold, S. (1978). Cross-Validatory Estimation of the Number of Components in Factor and Principal Component Models. *Technometrics*, **20**, 397-405.

Wold, S. and Andersson, K. (1973). Major Components Influencing Retention Indices in Gas Chromatography. *Journal of Chromatography*, **80**, 43-59.

Wold, S. and Lyttkens, E. (1969). Non-Linear Iterative Partial Least Squares (NIPALS) Estimation Procedures. *Bulletin of International Statistical Instituite: Proceedings, 37th Session, London*, 1-15.

Wu, C.F.J. (1986). Jackknife, Bootstrap, and Other Resampling Methods in Regression Analysis. *The Annals of Statistics*, **14**, 1261-1350.

Young, G. A. (1988). A Note on Bootstrapping the Correlation Coefficient. *Biometrika*, **75**, 370-373.

SUPPLEMENT TO APPENDIX B:

Computer Programs written in GAUSS (1988)

```
/* -------------------------------------------------------------- */
/*                            Program 1                           */
/*                            ---------                           */
/*                                                                */
/*                         Simulated data                         */
/*                         --------------                         */
/*                                                                */
/* This program generates data simulated according to models 1-4  */
/* of Jolliffe (1972) which are described in Table 4.1.           */
/*                                                                */
/* Input:                                                         */
/* -----                                                          */
/*                                                                */
/* 1. Number of Training samples                                  */
/* 2. Sample size                                                 */
/* 3. Number of Variables                                         */
/* 4. Model number                                                */
/*                                                                */
/* Output:                                                        */
/* ------                                                         */
/*                                                                */
/* 1. Simulated data                                              */
/*                                                                */
/* -------------------------------------------------------------- */
/* ------------------------------------ */
/* Requests for input.                  */
/* ------------------------------------ */
   PRINT "Program execution in progress...";
   PRINT;
   PRINT "Type in the number of training samples to generate";
   PRINT;
   tns=CON(1,1);
   PRINT;
   PRINT "Type in the sample size";
   PRINT;
   rsize=CON(1,1);
   PRINT;
   PRINT "Type in the number of variables";
   PRINT;
   clsize=CON(1,1);
   PRINT;
   PRINT "Type in the model number";
   PRINT;
   mod=CON(1,1);
   f1=1;
   f2=2;
   IF mod==1;
      IF rsize==100;
         fname="b:\\n100\\model1";
      ELSEIF rsize=50;
         fname="b:\\n50\\model1";
      ELSE;
         fname="b:\\n25\\model1";
      ENDIF;
      CREATE f1=^fname WITH zr,clsize,4;
      f1=CLOSE(f1);
      OPEN f2=^fname FOR UPDATE;
      PRINT "Generating and storing samples according to model 1";
      PRINT "Please wait...";
      PRINT;
   smplnumb=1;
   DO WHILE smplnumb<=tns;
      z=RNDN(rsize,clsize);
      x13=z[.,1:3];
      modl1=x13~(z[.,1]+0.5*z[.,4])~(z[.,2]+0.7*z[.,5])
            ~(z[.,3]+z[.,6]);
      PRINT "Sample [" smplnumb "] has been generated";
      CALL WRITER(f2,modl1);
      smplnumb=smplnumb+1;
   ENDO;
   CALL CLOSE(f2);
ELSEIF mod==2;
   IF rsize==100;
      fname="b:\\p100\\model2";
   ELSEIF rsize=50;
      fname="b:\\p50\\model2";
   ELSE;
      fname="b:\\p25\\model2";
   ENDIF;
   CREATE f1=^fname WITH zr,clsize,4;
   f1=CLOSE(f1);
   OPEN f2=^fname FOR UPDATE;
   PRINT "Generating and storing samples according to model 2";
   PRINT "Please wait...";
   PRINT;
   smplnumb=1;
   DO WHILE smplnumb<=tns;
      z=RNDN(rsize,clsize);
      x13=z[.,1:3];
      modl2=x13~(z[.,1]+0.5*z[.,4])~(z[.,2]+0.7*z[.,6])
            ~(z[.,2]+z[.,6]);
      PRINT "Sample [" smplnumb "] has been generated";
      CALL WRITER(f2,modl2);
      smplnumb=smplnumb+1;
   ENDO;
   CALL CLOSE(f2);
ELSEIF mod==3;
   IF rsize==100;
      fname="b:\\q100\\model3";
   ELSEIF rsize=50;
      fname="b:\\q50\\model3";
   ELSE;
      fname="b:\\q25\\model3";
   ENDIF;
   CREATE f1=^fname WITH zr,clsize,4;
   f1=CLOSE(f1);
   OPEN f2=^fname FOR UPDATE;
   PRINT "Generating and storing samples according to model 3";
   PRINT "Please wait...";
   PRINT;
   smplnumb=1;
   DO WHILE smplnumb<=tns;
      z=RNDN(rsize,clsize);
      x13=z[.,1:3];
      modl3=x13~(z[.,1]+0.8*z[.,2]+0.6*z[.,4])
            ~(z[.,2]+0.7*z[.,5])~(z[.,3]+0.5*z[.,6]);
      PRINT "Sample [" smplnumb "] has been generated";
      CALL WRITER(f2,modl3);
      smplnumb=smplnumb+1;
   ENDO;
   CALL CLOSE(f2);
ELSE;
   IF rsize==100;
      fname="b:\\m100\\model4";
   ELSEIF rsize=50;
      fname="b:\\m50\\model4";
   ELSE;
      fname="b:\\m25\\model4";
   ENDIF;
   CREATE f1=^fname WITH zr,clsize,4;
   f1=CLOSE(f1);
   OPEN f2=^fname FOR UPDATE;
   PRINT "Generating and storing samples according to model 4";
   PRINT "Please wait...";
   PRINT;
   smplnumb=1;
   DO WHILE smplnumb<=tns;
      z=RNDN(rsize,clsize);
      x12=z[.,1:2];
      mod41=x12~(z[.,2]+z[.,3])~(z[.,4])~(z[.,4]+0.75*z[.,5])
            ~(2*z[.,4]+0.75*z[.,5]+1.5*z[.,6]);
      mod42=(z[.,7])~(z[.,7]+0.5*z[.,8])~(2*z[.,7]+0.5*z[.,8]
            +z[.,9])~(3*z[.,7]+z[.,8]+z[.,9]+z[.,10]);
      modl4=mod41~mod42;
      PRINT "Sample [" smplnumb "] has been generated";
      CALL WRITER(f2,modl4);
      smplnumb=smplnumb+1;
   ENDO;
   CALL CLOSE(f2);
ENDIF;
```

```
/* -------------------------------------------------------------- */
/*                            Program 2                           */
/*                            ---------                           */
/*                                                                */
/*       Cross-Validatory choice of the number of components      */
/*       ---------------------------------------------------      */
/*                                                                */
/* This program performs the cross-validatory choice of the       */
/* number of components proposed by Eastment and Krzanowski       */
/* (1982).                                                        */
/*                                                                */
/* Input:                                                         */
/* -----                                                          */
/*                                                                */
/* 1. Number of Training samples                                  */
/* 2. Sample size                                                 */
/* 3. Number of Variables                                         */
/* 4. Data set                                                    */
/*                                                                */
/* Output:                                                        */
/* ------                                                         */
/*                                                                */
/* 1. PRESS values                                                */
/* 2. Cross-validatory W statistic values (equivalent to F ratio) */
/* 3. Computational time (in seconds)                             */
/*                                                                */
/* -------------------------------------------------------------- */
   NEW;                       @ Clearing the computer memory        @
   #INCLUDE SVD.H;            @ Loading the procedure for SVD       @
/* ------------------------------------ */
/* Loading the data into the program    */
/* ------------------------------------ */
   PRINT "Program execution in progress...";
   PRINT;
   PRINT "Type in the number of training samples";
   PRINT;
   tns=CON(1,1);
   PRINT;
   PRINT "Type in the sample size";
   PRINT;
   rsize=CON(1,1);
   PRINT;
smplnumb=1;
DO WHILE smplnumb<=tns;
   time1=TIME;                     @ Initial time  @
   rnumb=((smplnumb-1)*rsize)+1;   @ Row number    @
   f4=4;
   fname1="b:\\n25\\model1";       @ External data file             @
   OPEN f4=^fname1 FOR READ;       @ Opening data file for read     @
   CALL SEEKR(f4,rnumb);
   Z=READR(f4,rsize);
   f4=CLOSE(f4);
   n=ROWS(Z);
   p=COLS(Z);
/* -------------------------------------------------------------- */
/* Mean-centering the data, and standardizing it (if necessary)   */
/* -------------------------------------------------------------- */
   PROC STD(Y);
        LOCAL mean,ctrans,deviat;
        mean=MEANC(Y);
        ctrans=((-1)*mean)+Y';
        deviat=STDC(Y);
        RETP((ctrans./deviat)');
   ENDP;
   X=(((-1)*MEANC(Z))+Z')';
   CLEAR Z;
/* --------------------------------------------------------------- */
/* A procedure used to find the signs of elements in given matrix */
/* --------------------------------------------------------------- */
   PROC SIGN(L);
        LOCAL q;
        q=ROUND(ABS(L)./L);
        RETP(q);
   ENDP;
/* ---------------------------------------------- */
/* Performing the SVD of the complete data matrix */
/* ---------------------------------------------- */
   {Uf,Sf,Vf}=SVD2(X);
/* --------------------------------------------------------------- */
/* Creating a file on disk to store estimated values of xij, the   */
/* elements of the data matrix X.                                  */
/* --------------------------------------------------------------- */
   f1=1;
   f2=2;
   fname="a:\\estmxij";
   CREATE f1=^fname WITH xe,1,4;
   f1=CLOSE(f1);
   OPEN f2=^fname FOR UPDATE;
/* --------------------------------------------------------------- */
/* Estimating the data matrix X using the first M components.      */
/* --------------------------------------------------------------- */
   PRINT;
   PRINT "Estimating the original data matrix using each of the";
   PRINT "first p-1 PCs and storing the results in drive A:...";
   PRINT;
   PRINT "Please wait...";
   i=1;
   ei=ZEROS(n,1)  @ Tracks the deletion of rows of X                @
   ej=ZEROS(P,1)  @ Tracks the deletion of columns of X             @
   DO UNTIL i>n;
      ei[i,1]=1;
      Xdeli=DELIF(X,ei);    @ Deleting the i-th row                 @
      ei[i,1]=0;
      {Udeli,Sdeli,Vdeli}=SVD2(Xdeli);
                                      @ SVD of X without the row i  @
      CLEAR Xdeli,Udeli;
      Sdelip=DIAG(Sdeli);
      CLEAR Sdeli;
      U=Uf[i,1:p-1];
      j=1;
      DO UNTIL j>p;
         ej[j,1]=1;
         Xdelj=(DELIF(X',ej))';    @ Deleting the j-th column       @
         ej[j,1]=0;
         {Udelj,Sdelj,Vdelj}=SVD2(Xdelj);
                                    @ SVD of X without the column j @
         CLEAR Xdelj,Vdelj;
         Sdeljp=DIAG(Sdelj);
         CLEAR Sdelj;
         Vft=Vf';
         V=Vft[1:p-1,j];
         Vit=Vdeli'
         Vi=Vit[1:p-1,j];
         Uj=Udelj[i.,];
         UV=U'.*V;
         UjVi=Uj'.*Vi;
         UjVic=ABS(UjVi).*SIGN(UV);  @ Performing a parity check   @
                                     @ in order to obtain unique   @
                                     @ estimates of xij.           @
         CLEAR UjVi;
         M=1;
         DO UNTIL M>p-1;    @ Setting the number of PCs to M,      @
                            @ M = 1, 2, ..., p-1.                  @
            SdeliM=Sdelip[1:M,1];
            SdeljM=Sdeljp[1:M,1];
            UjVicM=UjVic[1:M,1];
            UVSjM=UjVicM.*SQRT(SdeljM);
            UVSSM=UVSjM.*SQRT(SdeliM);
            xijM=SUMC(UVSSM);
            CALL WRITER(f2,xijM);
            M=M+1;
         ENDO;
         j=j+1;
      ENDO;
      i=i+1;
   ENDO;
   f2=CLOSE(f2);
/* -------------------------------------------------------------- */
/* Computing PRESS values for the M-th PC, M = 0, 1, 2, ..., p-1. */
/* -------------------------------------------------------------- */
   f3=3;
   OPEN f3=^fname FOR READ;
   XM=ONES(n,p);
   PRESS=ONES(p,1);
   PRESS0=(SUMC(DIAG(X'*X)))/(n*p);
   CLS;                              @ Clearing the screen         @
   PRINT "PRESS[" 0 "]=" PRESS0;
   s=1;
   t=1;
   M=1;
   DO UNTIL M>p-1;
      i=1;
      DO UNTIL i>n;
         j=1;
         DO UNTIL j>p;
            CALL SEEKR(f3,t);
            XM[i,j]=READR(f3,1);
            t=t+p-1;
            j=j+1;
         ENDO;
         i=i+1;
      ENDO;
      PRESS[M,1]=(SUMC(DIAG((XM-X)'*(XM-X))))/(n*p);
      PRINT "PRESS[" M "]=" PRESS[M,1];
      M=M+1;
      s=s+1;
      t=s;
   ENDO;
```

```
/* -------------------------------------------------------------- */
/* Rearranging the PRESS values to include PRESS[0] in the same    */
/* variable name PRESS.                                            */
/* -------------------------------------------------------------- */
   M=p-1;
   DO WHILE M>=1;
      PRESS[M+1,1]=PRESS[M,1];
      M=M-1;
   ENDO;
   PRESS[1,1]=PRESS0;
/* -------------------------------------------------------------- */
/* Computing the W statistic for the M-th PC, M = 1, 2, ..., p-1. */
/* -------------------------------------------------------------- */
   Mvalue=ONES(p-1,1);
   Wvalue=ONES(p-1,1);
   M=2;
   DMC=0;
   DO UNTIL M>p;
      DM=n+p-2*(M-1);
      DMC=DMC+DM;
      DR=n*p-p-DMC;
      Mvalue[M-1,1]=M-1;
      Wvalue[M-1,1]=(PRESS[M-1,1]-PRESS[M,1])*DR/(DM*PRESS(M,1);
      M=M+1;
   ENDO;
   f3=CLOSE(f3);
/* --------------------- */
/* Printing the results  */
/* --------------------- */
   LET MWnames[2,1=Mvalue Wvalue;
   LET label[1,2]= M W;
   MW=MERGEVAR(MWnames);
   CLS;                  @ Clearing the screen                     @
   PRINT;
   PRINT $label;
   PRINT;
   PRINT MW;
   PRINT;
   smplnumb=smplnumb+1;
   time2=TIME;                     @ Final time     @
/* -------------------------------------------------------------- */
/* Converting and printing the computational time in seconds      */
/* -------------------------------------------------------------- */
   hr2=(time2[1,1])*3600;
   hr1=(time1[1,1])*3600;
   min2=(time2[2,1])*60;
   min1=(time1[2,1])*60;
   sec2=hr2+min2+(time2[3,1])+((time2[4,1])/100);
   sec1=hr1+min1+(time1[3,1])+((time1[4,1])/100);
   comptime=sec2-sec1;             @ Computational time in seconds @
   PRINT comptime;
   PRINT;
   PRINT "End of Session [" smplnumb "]";
   PRINT;
ENDO;
```

```
/* -------------------------------------------------------------- */
/*                            Program 3                           */
/*                            ---------                           */
/*                                                                */
/*         Bootstrap estimation of the number of components       */
/*         ------------------------------------------------       */
/*                                                                */
/* This program performs the proposed bootstrap choice of the     */
/* number of components in PCA.                                   */
/*                                                                */
/* Input:                                                         */
/* -----                                                          */
/*                                                                */
/* 1. Number of Training samples                                  */
/* 2. Sample size                                                 */
/* 3. Number of Variables                                         */
/* 4. Data set                                                    */
/*                                                                */
/* Output:                                                        */
/* ------                                                         */
/*                                                                */
/* 1. PRESS values                                                */
/* 2. Bootstrap W statistic values (equivalent to the F ratio)    */
/* 3. Computational time (in seconds)                             */
/*                                                                */
/* -------------------------------------------------------------- */
   NEW;                       @ Clearing the computer memory       @
   #INCLUDE SVD.H;            @ Loading the procedure for SVD      @
/* ------------------------------------ */
/* Loading the data into the program    */
/* ------------------------------------ */
   PRINT "Program execution in progress...";
   PRINT;
   PRINT "Type in the number of training samples";
   PRINT;
   tns=CON(1,1);
   PRINT;
   PRINT "Type in the sample size";
   PRINT;
   rsize=CON(1,1);
   PRINT;
smplnumb=1;
DO WHILE smplnumb<=tns;
   time1=TIME;                     @ Initial time  @
   rnumb=((smplnumb-1)*rsize)+1;   @ Row number    @
   f4=4;
   fname1="b:\\n25\\model1";       @ External data file                @
   OPEN f4=^fname1 FOR READ;       @ Opening data file for read        @
   CALL SEEKR(f4,rnumb);
   Z=READR(f4,rsize);
   f4=CLOSE(f4);
   n=ROWS(Z);
   p=COLS(Z);
   PRINT;
   PRINT "Type in the number of bootstrap samples";
   PRINT;
   B=CON(1,1);
   f1=1;
   f2=2;
   f3=3;
```

```
/* -------------------------------------------------------------- */
/* Creating and opening a file to store the PRESS values.          */
/* -------------------------------------------------------------- */
   fname="a:\\PRSFILE";
   CREATE f1=^fname WITH prs,1,4;
   f1=CLOSE(f1);
   OPEN f2=^fname FOR UPDATE;
/* -------------------------------------------------------------- */
/* For each bootstrap sample, predicting the bootstrap sample      */
/* using only the first M components, and computing and storing    */
/* PRESS[M], M = 1, 2, ..., p-2.                                   */
/* -------------------------------------------------------------- */
   PRESS=ZEROS(p,1);
   countB=1;
   DO WHILE countB<=B;
   rndrows=FLOOR(n*RNDU(n,1)+ONES(n,1));
   Y=SUBMAT(Z,rndrows,0);    @ Generating the bootstrap sample       @
   X=(Y'-MEANC(Y))';         @ Mean-centering the bootstrap sample   @
   PRESS[1,1]=(1/(n*p))*SUMC(DIAG(X'*X));     @ PRESS[0]            @
   {U,S,Vt}=SVD2(Y);
   V=Vt';
   M=1;
   DO WHILE M<=p-2;          @ Setting the number of PCs to M,       @
                             @ M = 1, 2, ..., p-2                    @
      XM=(U[.,1:M])*(S[1:M,1:M])*(V[1:M,.]);
      {UM,SM,VMt}=SVD2(XM);
      CLEAR SM, VMt;
      alpha=UM'*X;
      XMp=UM*alpha;          @ Predictor of the bootstrap sample     @
      discr=X-XMp;
      PRESS[M+1,1]=(1/(n*p))*SUMC(DIAG(discr'*discr));
      M=M+1;
   ENDO;
   CALL WRITER(f2,PRESS);
   countB=countB+1;
ENDO;
f2=CLOSE(f2);
/* -------------------------------------------------------------- */
/* Opening the file with PRESS values for reading and declaring    */
/* some variables.                                                 */
/* -------------------------------------------------------------- */
   OPEN f3=^fname FOR READ;  @ Opening the file with PRESS values    @
                             @ for reading                           @
   PRSM=ZEROS(B,1);          @ PRESS[M]                              @
   PRSMSUB1=ZEROS(B,1);      @ PRESS[M-1]                            @
   Wvalue=ZEROS(p,1);        @ Bootstrap W statistic values          @
   DM=ZEROS(p,1);            @ Numerator degrees of freedom          @
   DR=ZEROS(p,1);            @ Denominator degrees of freedom        @
   booterr=ZEROS(p-1,1);     @ Bootstrap error                       @
   avPRESS=ZEROS(p,1);       @ Mean of PRESS values over all         @
                             @ bootstrap samples                     @
/* -------------------------------------------------------------- */
/* Reading the PRESS values for M = 1 and M = 2.                   */
/* -------------------------------------------------------------- */
   countB=1;
   r=1;
   DO WHILE countB<=B;
      CALL SEEKR(f3,r);
      PRSMSUB1[countB,1]=READR(f3,1);
      r=r+1;
      CALL SEEKR(f3,r);
      PRSM[countB,1]=READR(f3,1);
      r=r+p-1;
      countB=countB+1;
   ENDO;
/* --------------------------------------------------------------- */
/* Computing the bootstrap W value, the bootstrap error and the    */
/* mean of the PRESS values for M = 1.                             */
/* --------------------------------------------------------------- */
   DM[1,1]=n;
   DR[1,1]=n*(p-2);
   NUM=(MEANC(PRSMSUB1)-MEANC(PRSM))/DM[1,1];
   DEN=MEANC(PRSM)DR[1,1];
   Wvalue[1,1]=NUM/DEN;
   booterr[1,1]=STDC(PRSMSUB1);
   avPRESS[1,1]=MEANC(PRSM);
/* --------------------------------------------------------------- */
/* Computing the bootstrap W value, the bootstrap error and the    */
/* mean of the PRESS values for M = 2, 3, ..., p-2.                */
/* --------------------------------------------------------------- */
   M=2;
   DO WHILE M<=p-2;
      PRSMSUB1=PRSM;
      r=M+1;
      countB=1;
      DO WHILE countB<=B;
         CALL SEEKR(f3,r);
         r=r+p
         countB=countB+1;
      ENDO;
      DM[M,1]=n;
      DR[M,1]=n*(p-M-1);
      NUM=(MEANC(PRSMSUB1)-MEANC(PRSM))/DM[M,1];
      DEN=MEANC(PRSM)DR[M,1];
      Wvalue[M,1]=NUM/DEN;
      booterr[M,1]=STDC(PRSMSUB1);
      avPRESS[M,1]=MEANC(PRSM);
      M=M+1;
   ENDO;
   f3=CLOSE(f3);
   booterr[p-1,1]=STDC(PRSM);
/* --------------------- */
/* Printing the results  */
/* --------------------- */
   compnent=CUMSUMC(ONES(p,1));
   results=compnent~avPRESS~Wvalue~DM~DR;
   LET title[1,5=k avPRESS Wvalue Ndf Ddf;
   CLS;                  @ Clearing the screen                      @
   PRINT;
   PRINT $title;
   PRINT;
   PRINT results;
   PRINT;
   smplnumb=smplnumb+1;
   time2=TIME;                      @ Final time    @
/* --------------------------------------------------------------- */
/* Converting and printing the computational time in seconds       */
/* --------------------------------------------------------------- */
   hr2=(time2[1,1])*3600;
   hr1=(time1[1,1])*3600;
   min2=(time2[2,1])*60;
   min1=(time1[2,1])*60;
   sec2=hr2+min2+(time2[3,1])+((time2[4,1])/100);
   sec1=hr1+min1+(time1[3,1])+((time1[4,1])/100);
   comptime=sec2-sec1;             @ Computational time in seconds @
   PRINT comptime;
   PRINT;
   PRINT "End of Session [" smplnumb "]";
   PRINT;
ENDO;
```

```
/* ------------------------------------------------------------- */
/*                            Program 4                          */
/*                            ---------                          */
/*                                                               */
/*               The M-squared-Procrustes criterion              */
/*               ----------------------------------              */
/*                      Backward elimination                     */
/*                      --------------------                     */
/*                                                               */
/* This program performs variable selection in PCA using the M-  */
/* squared procrustes criterion of Krzanowski (1987) and the     */
/* bakward elimination procedure for a given data set.           */
/*                                                               */
/* Input:                                                        */
/* -----                                                         */
/*                                                               */
/* 1. Number of Training samples                                 */
/* 2. Sample size                                                */
/* 3. Number of Variables                                        */
/* 4. Data set                                                   */
/* 5. Number of PCs to be used for variable selection            */
/*                                                               */
/* Output:                                                       */
/* ------                                                        */
/*                                                               */
/* 1. Deleted variables and the corresponding M-squared values   */
/* 2. Selected subset of variables                               */
/* 3. Computational time (in seconds)                            */
/*                                                               */
/* ------------------------------------------------------------- */
   NEW;                       @ Clearing the computer memory       @
   #INCLUDE SVD.H;            @ Loading the procedure for SVD      @
/* ------------------------------------ */
/* Loading the data into the program    */
/* ------------------------------------ */
   PRINT "Program execution in progress...";
   PRINT;
   PRINT "Type in the number of training samples";
   PRINT;
   tns=CON(1,1);
   PRINT;
   PRINT "Type in the sample size";
   PRINT;
   rsize=CON(1,1);
   PRINT;
   f11=11;
   fname2="b:\\n25\\model1";       @ External data file            @
   OPEN f11=^fname2 FOR READ;      @ Opening data file for read    @
   PRINT "Type in the number of PCs to be used";
   PRINT;
   M=CON(1,1);
smplnumb=1;
DO WHILE smplnumb<=tns;
   time1=TIME;                     @ Initial time  @
   Z=READR(f11,rsize);
   n=ROWS(Z);
   p=COLS(Z);
/* ------------------------------------------------------------- */
/* Mean-centering the data, and standardizing it (if necessary)  */
/* ------------------------------------------------------------- */
```

```
      leastM2[p-q+1,1]=MINC(trackM2);
      l=1;
      DO WHILE l<=p;
         IF M2all[l,1]==leastM2[p-q+1,1];
            vardel[p-q+1,1]=l;
            trackdel[l,1]=1;
            l=l+1;
         ELSE;
            l=l+1;
         ENDIF;
      ENDO;
      q=q-1;
   ENDO;
/* ------------------------------------ */
/* Identifying the retained variables   */
/* ------------------------------------ */
   slctv=ONES(1,M);    @ Contains the retained variables' indices  @
   k=1;
   l=1;
   DO WHILE l<=p;
      IF trackdel[l,1]==0;
         slctv[1,k]=l;
         k=k+1;
         l=l+1;
      ELSE;
         l=l+1;
      ENDIF;
   ENDO;
/* --------------------- */
/* Printing the results  */
/* --------------------- */
   LET VM2names[2,1]=vardel leastM2;
   VM2=MERGEVAR(VM2names);
   LET label[1,2]= Variable M-squared;
   PRINT $label;
   PRINT;
   PRINT VM2;
   PRINT;
   PRINT "Selected subset of variables:";
   PRINT;
   PRINT slctv;
   smplnumb=smplnumb+1;
   time2=TIME;                     @ Final time    @
/* -------------------------------------------------------------- */
/* Converting and printing the computational time in seconds      */
/* -------------------------------------------------------------- */
   hr2=(time2[1,1])*3600;
   hr1=(time1[1,1])*3600;
   min2=(time2[2,1])*60;
   min1=(time1[2,1])*60;
   sec2=hr2+min2+(time2[3,1])+((time2[4,1])/100);
   sec1=hr1+min1+(time1[3,1])+((time1[4,1])/100);
   comptime=sec2-sec1;             @ Computational time in seconds @
   PRINT;
   PRINT comptime;
   PRINT;
   PRINT "End of Session [" smplnumb "]";
   PRINT;
ENDO;
```

```
/* -------------------------------------------------------------- */
/*                           Program 5                            */
/*                           ---------                            */
/*                                                                */
/*                     The Gamma-5 criterion                      */
/*                     ---------------------                      */
/*                     Backward elimination                       */
/*                     --------------------                       */
/*                                                                */
/* This program performs variable selection in PCA using the     */
/* Gamma-5 criterion and the backward elimination procedure for   */
/* a given data set.                                              */
/*                                                                */
/* Input:                                                         */
/* -----                                                          */
/*                                                                */
/* 1. Number of Training samples                                  */
/* 2. Sample size                                                 */
/* 3. Number of Variables                                         */
/* 4. Data set                                                    */
/* 5. Number of PCs to be used for variable selection             */
/*                                                                */
/* Output:                                                        */
/* ------                                                         */
/*                                                                */
/* 1. Deleted variables and the corresponding  Gamma-5  values    */
/* 2. Selected subset of variables                                */
/* 3. Computational time (in seconds)                             */
/*                                                                */
/* -------------------------------------------------------------- */
   NEW;                       @ Clearing the computer memory       @
   #INCLUDE SVD.H;            @ Loading the procedure for SVD      @
/* ------------------------------------ */
/* Loading the data into the program    */
/* ------------------------------------ */
   PRINT "Program execution in progress...";
   PRINT;
   PRINT "Type in the number of training samples";
   PRINT;
   tns=CON(1,1);
   PRINT;
   PRINT "Type in the sample size";
   PRINT;
   rsize=CON(1,1);
   PRINT;
   f11=11;
   fname2="b:\\n25\\model1";       @ External data file            @
   OPEN f11=^fname2 FOR READ;      @ Opening data file for read    @
   PRINT "Type in the number of PCs to be used";
   PRINT;
   M=CON(1,1);
smplnumb=1;
DO WHILE smplnumb<=tns;
   time1=TIME;                     @ Initial time  @
   Z=READR(f11,rsize);
   n=ROWS(Z);
   p=COLS(Z);
/* -------------------------------------------------------------- */
/* Mean-centering the data, and standardizing it (if necessary)   */
/* -------------------------------------------------------------- */
```

```
   PROC STD(Y);
        LOCAL mean,ctrans,deviat;
        mean=MEANC(Y);
        ctrans=((-1)*mean)+Y';
        deviat=STDC(Y);
        RETP((ctrans./deviat)');
   ENDP;
   X=(((-1)*MEANC(Z))+Z')';
   CLEAR Z;
/* ------------------------------------------------------------ */
/* Variable declaration and computing PC scores for complete data */
/* ------------------------------------------------------------ */
   {Uf,Sf,Vf}=SVD2(X);
   CLEAR Vf;
   UfSf=Uf*Sf;
   CLEAR Uf,Sf;
   Yj=UfSf[.,1:M];
   CLEAR UfSf;
   trackdel=ZEROS(p,1);         @ Tracks deleted variables in X.     @
   peakG5=ONES(p-M,1);          @ Stores the largest Gamma-5 value.  @
   vardel=ONES(p-M,1);          @ Identifies variables to be deleted @
   q=p;                         @ Initial subset size.               @
/* ------------------------------------------------------------ */
/* Computing the  Gamma-5  criterion for each variable omitted   */
/* in turn from the data matrix X.                               */
/* ------------------------------------------------------------ */
/* DO WHILE q>M;
      trackG5=ONES(q,1);        @ Stores  Gamma-5  values.           @
      G5all=-1000*(ONES(p,1);   @ Gamma-5  values  for variables     @
                                @ omitted from X, indexed in the     @
                                @ order in which the variables       @
                                @ appear in X.                       @
      k=1;
      j=1;
      DO WHILE j<=p;
         IF trackdel[j,1]==0;
            trackdel[j,1]=1;    @ Indexing the j-th variable for     @
                                @ omission.                          @
            Xdelj=(DELIF(X',trackdel))';      @ New X.               @
            trackdel[j,1]=0;    @ Indexing omitted variable for      @
                                @ replacement.                       @
            {Udelj,Sdelj,Vdelj}=SVD2(Xdelj); @ SVD of new X.         @
            CLEAR Xdelj, Vdelj;
            UjSj=Udelj*Sdelj;
            CLEAR Udelj,Sdelj;
            Zj=UjSj[.,1:M];
            CLEAR UjSj;
            YjZj=Yj~Zj;
            CLEAR Zj;
            RYjZj=CORRX(YjZj);
            CLEAR YjZj;
            RYZ=RYjZj[1:M,M+1:2*M];
            CLEAR RYjZj;
            cancor=ONES(M,1)-(SVD(RYZ*RYZ')-0.5*ONES(M,1));
            G5all[j,1]=1-((PRODC(cancor))^(1/M));  @ Computing       @
                                                    @ Gamma-5.        @
            trackG5[k,1]=G5all[j,1];
            k=k+1;
            j=j+1;
         ELSE;
```

```
            j=j+1;
         ENDIF;
      ENDO;
/* -------------------------------------------------------------- */
/* Deleting the variable whose omission yields the largest value  */
/* of Gamma-5.                                                    */
/* -------------------------------------------------------------- */
      peakG5[p-q+1,1]=MAXC(trackG5);
      l=1;
      DO WHILE l<=p;
         IF G5all[l,1]==peakG5[p-q+1,1];
            vardel[p-q+1,1]=l;
            trackdel[l,1]=1;
            l=l+1;
         ELSE;
            l=l+1;
         ENDIF;
      ENDO;
      q=q-1;
   ENDO;
/* ------------------------------------ */
/* Identifying the retained variables   */
/* ------------------------------------ */
   slctv=ONES(1,M);    @ Contains the retained variables' indices  @
   k=1;
   l=1;
   DO WHILE l<=p;
      IF trackdel[l,1]==0;
         slctv[1,k]=l;
         k=k+1;
         l=l+1;
      ELSE;
         l=l+1;
      ENDIF;
   ENDO;
/* -------------------- */
/* Printing the results */
/* -------------------- */
   LET VG5names[2,1]=vardel peakG5;
   VG5=MERGEVAR(VG5names);
   LET label[1,2]= Variable Gamma-5;
   PRINT $label;
   PRINT;
   PRINT VG5;
   PRINT;
   PRINT "Selected subset of variables:";
   PRINT;
   PRINT slctv;
   smplnumb=smplnumb+1;
   time2=TIME;                      @ Final time    @
/* -------------------------------------------------------------- */
/* Converting and printing the computational time in seconds      */
/* -------------------------------------------------------------- */
   hr2=(time2[1,1])*3600;
   hr1=(time1[1,1])*3600;
   min2=(time2[2,1])*60;
   min1=(time1[2,1])*60;
   sec2=hr2+min2+(time2[3,1])+((time2[4,1])/100);
   sec1=hr1+min1+(time1[3,1])+((time1[4,1])/100);
   comptime=sec2-sec1;              @ Computational time in seconds @
   PRINT;
   PRINT comptime;
   PRINT;
   PRINT "End of Session [" smplnumb "]";
   PRINT;
ENDO;
```

```
/* ------------------------------------------------------------- */
/*                          Program 6                            */
/*                          ---------                            */
/*                                                               */
/*                   The Gamma-6 criterion                       */
/*                   ---------------------                       */
/*                   Backward elimination                        */
/*                   --------------------                        */
/*                                                               */
/* This program performs variable selection in PCA using the     */
/* Gamma-6 criterion and the backward elimination procedure for  */
/* a given data set.                                             */
/*                                                               */
/* Input:                                                        */
/* -----                                                         */
/*                                                               */
/* 1. Number of Training samples                                 */
/* 2. Sample size                                                */
/* 3. Number of Variables                                        */
/* 4. Data set                                                   */
/* 5. Number of PCs to be used for variable selection            */
/*                                                               */
/* Output:                                                       */
/* ------                                                        */
/*                                                               */
/* 1. Deleted variables and the corresponding  Gamma-6  values   */
/* 2. Selected subset of variables                               */
/* 3. Computational time (in seconds)                            */
/*                                                               */
/* ------------------------------------------------------------- */
   NEW;                        @ Clearing the computer memory     @
   #INCLUDE SVD.H;             @ Loading the procedure for SVD    @
/* ----------------------------------- */
/* Loading the data into the program   */
/* ----------------------------------- */
   PRINT "Program execution in progress...";
   PRINT;
   PRINT "Type in the number of training samples";
   PRINT;
   tns=CON(1,1);
   PRINT;
   PRINT "Type in the sample size";
   PRINT;
   rsize=CON(1,1);
   PRINT;
   f11=11;
   fname2="b:\\n25\\model1";       @ External data file           @
   OPEN f11=^fname2 FOR READ;      @ Opening data file for read   @
   PRINT "Type in the number of PCs to be used";
   PRINT;
   M=CON(1,1);
smplnumb=1;
DO WHILE smplnumb<=tns;
   time1=TIME;                     @ Initial time  @
   Z=READR(f11,rsize);
   n=ROWS(Z);
   p=COLS(Z);
/* ------------------------------------------------------------- */
/* Mean-centering the data, and standardizing it (if necessary)  */
/* ------------------------------------------------------------- */
```

```
   PROC STD(Y);
        LOCAL mean,ctrans,deviat;
        mean=MEANC(Y);
        ctrans=((-1)*mean)+Y';
        deviat=STDC(Y);
        RETP((ctrans./deviat)');
   ENDP;
   X=(((-1)*MEANC(Z))+Z')';
   CLEAR Z;
/* -------------------------------------------------------------- */
/* Variable declaration and computing PC scores for complete data */
/* -------------------------------------------------------------- */
   {Uf,Sf,Vf}=SVD2(X);
   CLEAR Vf;
   UfSf=Uf*Sf;
   CLEAR Uf,Sf;
   Yj=UfSf[.,1:M];
   CLEAR UfSf;
   trackdel=ZEROS(p,1);       @ Tracks deleted variables in X.     @
   peakG6=ONES(p-M,1);        @ Stores the largest Gamma-6 value.  @
   vardel=ONES(p-M,1);        @ Identifies variables to be deleted @
   q=p;                       @ Initial subset size.               @
/* -------------------------------------------------------------- */
/* Computing the  Gamma-6  criterion for each variable omitted    */
/* in turn from the data matrix X.                                */
/* -------------------------------------------------------------- */
/* DO WHILE q>M;
      trackG6=ONES(q,1);      @ Stores  Gamma-6  values.           @
      G6all=-1000*(ONES(p,1); @ Gamma-6  values  for variables     @
                              @ omitted from X, indexed in the     @
                              @ order in which the variables       @
                              @ appear in X.                       @
      k=1;
      j=1;
      DO WHILE j<=p;
         IF trackdel[j,1]==0;
            trackdel[j,1]=1;  @ Indexing the j-th variable for     @
                              @ omission.                          @
            Xdelj=(DELIF(X',trackdel))';     @ New X.              @
            trackdel[j,1]=0;  @ Indexing omitted variable for      @
                              @ replacement.                       @
            {Udelj,Sdelj,Vdelj}=SVD2(Xdelj); @ SVD of new X.       @
            CLEAR Xdelj, Vdelj;
            UjSj=Udelj*Sdelj;
            CLEAR Udelj,Sdelj;
            Zj=UjSj[.,1:M];
            CLEAR UjSj;
            YjZj=Yj~Zj;
            CLEAR Zj;
            RYjZj=CORRX(YjZj);
            CLEAR YjZj;
            RYZ=RYjZj[1:M,M+1:2*M];
            CLEAR RYjZj;
            G6all[j,1]=(SUMC(SVD(RYZ*RYZ')))/M;   @ Computing      @
                                                  @ Gamma-6.       @
            trackG6[k,1]=G6all[j,1];
            k=k+1;
            j=j+1;
         ELSE;
            j=j+1;
         ENDIF;
      ENDO;
/* ------------------------------------------------------------------ */
/* Deleting the variable whose omission yields the largest value      */
/* of Gamma-6.                                                        */
/* ------------------------------------------------------------------ */
      peakG6[p-q+1,1]=MAXC(trackG6);
      l=1;
      DO WHILE l<=p;
         IF G6all[l,1]==peakG6[p-q+1,1];
            vardel[p-q+1,1]=l;
            trackdel[l,1]=1;
            l=l+1;
         ELSE;
            l=l+1;
         ENDIF;
      ENDO;
      q=q-1;
   ENDO;
/* ------------------------------------ */
/* Identifying the retained variables   */
/* ------------------------------------ */
   slctv=ONES(1,M);    @ Contains the retained variables' indices  @
   k=1;
   l=1;
   DO WHILE l<=p;
      IF trackdel[l,1]==0;
         slctv[1,k]=l;
         k=k+1;
         l=l+1;
      ELSE;
         l=l+1;
      ENDIF;
   ENDO;
/* --------------------- */
/* Printing the results  */
/* --------------------- */
   LET VG6names[2,1]=vardel peakG6;
   VG6=MERGEVAR(VG6names);
   LET label[1,2]= Variable Gamma-6;
   PRINT $label;
   PRINT;
   PRINT VG6;
   PRINT;
   PRINT "Selected subset of variables:";
   PRINT;
   PRINT slctv;
   smplnumb=smplnumb+1;
   time2=TIME;                     @ Final time    @
/* ------------------------------------------------------------------ */
/* Converting and printing the computational time in seconds          */
/* ------------------------------------------------------------------ */
   hr2=(time2[1,1])*3600;
   hr1=(time1[1,1])*3600;
   min2=(time2[2,1])*60;
   min1=(time1[2,1])*60;
   sec2=hr2+min2+(time2[3,1])+((time2[4,1])/100);
   sec1=hr1+min1+(time1[3,1])+((time1[4,1])/100);
   comptime=sec2-sec1;             @ Computational time in seconds @
   PRINT;
   PRINT comptime;
   PRINT;
   PRINT "End of Session [" smplnumb "]";
   PRINT;
ENDO;
```

```
/* -------------------------------------------------------------- */
/*                            Program 7                           */
/*                            ---------                           */
/*                                                                */
/*         The Delta criterion (Using the Complete graph)         */
/*         ----------------------------------------------         */
/*                      Backward elimination                      */
/*                      --------------------                      */
/*                                                                */
/* This program performs variable selection in PCA using the      */
/* Delta criterion and the backward elimination procedure for     */
/* a given data set.                                              */
/*                                                                */
/* Input:                                                         */
/* -----                                                          */
/*                                                                */
/* 1. Number of Training samples                                  */
/* 2. Sample size                                                 */
/* 3. Number of Variables                                         */
/* 4. Data set                                                    */
/* 5. Number of PCs to be used for variable selection             */
/*                                                                */
/* Output:                                                        */
/* ------                                                         */
/*                                                                */
/* 1. Deleted variables and the corresponding Delta values        */
/* 2. Selected subset of variables                                */
/* 3. Computational time (in seconds)                             */
/*                                                                */
/* -------------------------------------------------------------- */
   NEW;                       @ Clearing the computer memory      @
   #INCLUDE SVD.H;            @ Loading the procedure for SVD     @
/* ------------------------------------ */
/* Loading the data into the program    */
/* ------------------------------------ */
   PRINT "Program execution in progress...";
   PRINT;
   PRINT "Type in the number of training samples";
   PRINT;
   tns=CON(1,1);
   PRINT;
   PRINT "Type in the sample size";
   PRINT;
   rsize=CON(1,1);
   PRINT;
   f11=11;
   fname2="b:\\n25\\model1";       @ External data file            @
   OPEN f11=^fname2 FOR READ;      @ Opening data file for read    @
   PRINT "Type in the number of PCs to be used";
   PRINT;
   M=CON(1,1);
smplnumb=1;
DO WHILE smplnumb<=tns;
   time1=TIME;                     @ Initial time  @
   Z=READR(f11,rsize);
   n=ROWS(Z);
   p=COLS(Z);
/* -------------------------------------------------------------- */
/* Mean-centering the data, and standardizing it (if necessary)   */
/* -------------------------------------------------------------- */
```

```
   PROC STD(Y);
        LOCAL mean,ctrans,deviat;
        mean=MEANC(Y);
        ctrans=((-1)*mean)+Y';
        deviat=STDC(Y);
        RETP((ctrans./deviat)');
   ENDP;
   X=(((-1)*MEANC(Z))+Z')';
   CLEAR Z;
/* -------------------------------------------------------------- */
/* Computing and standardizing the PC scores for complete data    */
/* -------------------------------------------------------------- */
   {Uf,Sf,Vf}=SVD2(X);
   CLEAR Vf;
   UfSf=Uf*Sf;
   CLEAR Uf,Sf;
   Yjunst=UfSf[.,1:M];        @ PC scores                          @
   CLEAR UfSf;
   Yj=((Yjunst')./STDC(Yjunst))';  @ Standardized PC scores        @
   Yjt=Yj';
   CLEAR Yj;
/* -------------------------------------------------------------- */
/* Computing the distance matrix for the complete data            */
/* -------------------------------------------------------------- */
   nf=1;
   DO UNTIL nf>n-1;
      Yjnf=SQRT(SUMC((Yjt[.,nf]-Yjt[.,nf+1:n])^2));
      IF nf==1;
         Yjnfd=Yjnf;
      ELSE;
         Yjnfd=Yjnfd|Yjnf;  @ Distance matrix for complete data    @
      ENDIF;
      nf=nf+1;
   ENDO;
   CLEAR Yjt;
/* --------------------- */
/* Declaring variables   */
/* --------------------- */
   trackdel=ZEROS(p,1);       @ Tracks deleted variables in X.     @
   leastD=ONES(p-M,1);        @ Stores the least Delta value.      @
   vardel=ONES(p-M,1);        @ Identifies variables to be deleted @
   q=p;                       @ Initial subset size.               @
/* -------------------------------------------------------------- */
/* Computing the Delta criterion for each variable omitted        */
/* in turn from the data matrix X.                                */
/* -------------------------------------------------------------- */
/* DO WHILE q>M;
      trackD=ONES(q,1);       @ Stores the Delta values.           @
      Dall=-1000*(ONES(p,1);  @ Delta values for variables         @
                              @ omitted from X, indexed in the     @
                              @ order in which the variables       @
                              @ appear in X.                       @
      k=1;
      j=1;
      DO WHILE j<=p;
         IF trackdel[j,1]==0;
            trackdel[j,1]=1;  @ Indexing the j-th variable for     @
                              @ omission.                          @
            Xdelj=(DELIF(X',trackdel))';     @ New X.              @
            trackdel[j,1]=0;  @ Indexing omitted variable for      @
```

```
                              @ replacement.                        @
            {Udelj,Sdelj,Vdelj}=SVD2(Xdelj); @ SVD of new X.        @
            CLEAR Xdelj, Vdelj;
            UjSj=Udelj*Sdelj;
            CLEAR Udelj,Sdelj;
            Zjunst=UjSj[.,1:M];    @ PC scores of new X             @
            CLEAR UjSj;
            Zj=((Zjunst')./STDC(Zjunst))';   @ Standardized PC sc.  @
            Zjt=Zj';
            CLEAR Zj;
/* --------------------------------------------------- */
/* Computing the distance matrix for the subset data   */
/* --------------------------------------------------- */
            nj=1;
            DO UNTIL nj>n-1;
               Zjnj=SQRT(SUMC((Zjt[.,nj]-Zjt[.,nj+1:n])^2));
               IF nj==1;
                  Zjnjd=Zjnj;
               ELSE;
                  Zjnjd=Zjnjd|Zjnj;  @ Distance matrix for          @
                                     @ the subset data.             @
               ENDIF;
               nj=nj+1;
            ENDO;
            CLEAR Zjt;
            Dall[j,1]=SUMC(ABS(Yjnfd-Zjnjd));   @ Computing Delta   @
            trackD[k,1]=Dall[j,1];
            k=k+1;
            j=j+1;
         ELSE;
            j=j+1;
         ENDIF;
      ENDO;
/* --------------------------------------------------------------- */
/* Deleting the variable whose omission yields the least value     */
/* of Delta.                                                       */
/* --------------------------------------------------------------- */
      leastD[p-q+1,1]=MINC(trackD);
      l=1;
      DO WHILE l<=p;
         IF Dall[l,1]==leastD[p-q+1,1];
            vardel[p-q+1,1]=l;
            trackdel[l,1]=1;
            l=l+1;
         ELSE;
            l=l+1;
         ENDIF;
      ENDO;
      q=q-1;
   ENDO;
/* ------------------------------------ */
/* Identifying the retained variables   */
/* ------------------------------------ */
   slctv=ONES(1,M);    @ Contains the retained variables' indices   @
   k=1;
   l=1;
   DO WHILE l<=p;
      IF trackdel[l,1]==0;
         slctv[1,k]=l;
         k=k+1;
         l=l+1;
      ELSE;
         l=l+1;
      ENDIF;
   ENDO;
/* --------------------- */
/* Printing the results  */
/* --------------------- */
   LET VDnames[2,1]=vardel leastD;
   VD=MERGEVAR(VDnames);
   LET label[1,2]= Variable Delta;
   PRINT label;
   PRINT;
   PRINT VD;
   PRINT;
   PRINT "Selected subset of variables:";
   PRINT;
   PRINT slctv;
   smplnumb=smplnumb+1;
   time2=TIME;                      @ Final time    @
/* -------------------------------------------------------------- */
/* Converting and printing the computational time in seconds      */
/* -------------------------------------------------------------- */
   hr2=(time2[1,1])*3600;
   hr1=(time1[1,1])*3600;
   min2=(time2[2,1])*60;
   min1=(time1[2,1])*60;
   sec2=hr2+min2+(time2[3,1])+((time2[4,1])/100);
   sec1=hr1+min1+(time1[3,1])+((time1[4,1])/100);
   comptime=sec2-sec1;              @ Computational time in seconds @
   PRINT;
   PRINT comptime;
   PRINT;
   PRINT "End of Session [" smplnumb "]";
   PRINT;
ENDO;
```

```
/* -------------------------------------------------------------- */
/*                          Program 8                             */
/*                          ---------                             */
/*                                                                */
/* The Delta Star criterion (using the K-Nearest Neighbour graph) */
/* -------------------------------------------------------------- */
/*                    Backward elimination                        */
/*                    --------------------                        */
/*                                                                */
/* This program performs variable selection in PCA using the      */
/* Delta Star criterion and the backward elimination procedure    */
/* for a given data set.                                          */
/*                                                                */
/* Input:                                                         */
/* -----                                                          */
/*                                                                */
/* 1. Number of Training samples                                  */
/* 2. Sample size                                                 */
/* 3. Number of Variables                                         */
/* 4. Data set                                                    */
/* 5. Number of PCs to be used for variable selection             */
/*                                                                */
/* Output:                                                        */
/* ------                                                         */
/*                                                                */
/* 1. Deleted variables and the corresponding Delta Star values   */
/* 2. Selected subset of variables                                */
/* 3. Computational time (in seconds)                             */
/*                                                                */
/* -------------------------------------------------------------- */
   NEW;                        @ Clearing the computer memory     @
   #INCLUDE SVD.H;             @ Loading the procedure for SVD    @
/* ------------------------------------ */
/* Loading the data into the program    */
/* ------------------------------------ */
   PRINT "Program execution in progress...";
   PRINT;
   PRINT "Type in the number of training samples";
   PRINT;
   tns=CON(1,1);
   PRINT;
   PRINT "Type in the sample size";
   PRINT;
   rsize=CON(1,1);
   PRINT;
   f11=11;
   fname2="b:\\n25\\model1";      @ External data file            @
   OPEN f11=^fname2 FOR READ;     @ Opening data file for read    @
   PRINT "Type in the number of PCs to be used";
   PRINT;
   M=CON(1,1);
smplnumb=1;
DO WHILE smplnumb<=tns;
   time1=TIME;                    @ Initial time  @
   Z=READR(f11,rsize);
   n=ROWS(Z);
   p=COLS(Z);
/* -------------------------------------------------------------- */
/* Mean-centering the data, and standardizing it (if necessary)   */
/* -------------------------------------------------------------- */
```

```
   PROC STD(Y);
        LOCAL mean,ctrans,deviat;
        mean=MEANC(Y);
        ctrans=((-1)*mean)+Y';
        deviat=STDC(Y);
        RETP((ctrans./deviat)');
   ENDP;
   X=STD(Z);
   CLEAR Z;
/* -------------------------------------------------------------- */
/* Computing and standardizing the PC scores for complete data    */
/* -------------------------------------------------------------- */
   {Uf,Sf,Vf}=SVD2(X);
   CLEAR Vf;
   UfSf=Uf*Sf;
   CLEAR Uf,Sf;
   Yjunst=UfSf[.,1:M];        @ PC scores                            @
   CLEAR UfSf;
   Yj=((Yjunst')./STDC(Yjunst))';  @ Standardized PC scores          @
   Yjt=Yj';
   CLEAR Yj;
/* -------------------------------------------------------------- */
/* Computing the distance matrix for the complete data and        */
/* simultaneously constructing the KNN graph                      */
/* -------------------------------------------------------------- */
   nf=1;
   DO WHILE nf<=n;
      Yjnf=SQRT(SUMC((Yjt[.,nf]-Yjt)^2));
      IF nf==1;
         wt=SORTC(Yjnf,1);
         minwt=wt[2,1];
         Yjnfd=Yjnf[2:n,1];
      ELSE;
         wt=SORTC(Yjnf,1);
         minwt=minwt|wt[2,1];
         IF nf==n;
            Yjnfd=Yjnfd;
         ELSE;
            Yjnfd=Yjnfd|Yjnf[nf+1:n,1]; @ Distance matrix            @
         ENDIF;
      ENDIF;
      nf=nf+1;
   ENDO;
   CLEAR Yjt;
/* -------------------------------------------------------------- */
/* Obtaining and storing the indices of the KNN graph             */
/* -------------------------------------------------------------- */
   knnind=INDNV(minwt,Yjnfd);
   Yjnfd=Yjnfd[knnind,1];
/* --------------------- */
/* Declaring variables   */
/* --------------------- */
   trackdel=ZEROS(p,1);       @ Tracks deleted variables in X.       @
   leastDS=ONES(p-M,1);       @ Stores the least Delta Star value.  @
   vardel=ONES(p-M,1);        @ Identifies variables to be deleted  @
   q=p;                       @ Initial subset size.                 @
/* -------------------------------------------------------------- */
/* Computing the Delta Star criterion for each variable omitted   */
/* in turn from the data matrix X.                                */
/* -------------------------------------------------------------- */
/* DO WHILE q>M;
      trackDS=ONES(q,1);      @ Stores the Delta Star values.      @
      DSall=-1000*(ONES(p,1); @ Delta Star values for variables    @
                              @ omitted from X, indexed in the     @
                              @ order in which the variables       @
                              @ appear in X.                       @
      k=1;
      j=1;
      DO WHILE j<=p;
         IF trackdel[j,1]==0;
            trackdel[j,1]=1;  @ Indexing the j-th variable for     @
                              @ omission.                          @
            Xdelj=(DELIF(X',trackdel))';     @ New X.              @
            trackdel[j,1]=0;  @ Indexing omitted variable for      @
                              @ replacement.                       @
            {Udelj,Sdelj,Vdelj}=SVD2(Xdelj); @ SVD of new X.       @
            CLEAR Xdelj, Vdelj;
            UjSj=Udelj*Sdelj;
            CLEAR Udelj,Sdelj;
            Zjunst=UjSj[.,1:M];    @ PC scores for new X           @
            CLEAR UjSj;
            Zj=((Zjunst')./STDC(Zjunst))';   @ Standardized PC sc. @
            Zjt=Zj';
            CLEAR Zj;
/* --------------------------------------------------- */
/* Computing the distance matrix for the subset data   */
/* --------------------------------------------------- */
            nj=1;
            DO UNTIL nj>n-1;
               Zjnj=SQRT(SUMC((Zjt[.,nj]-Zjt[.,nj+1:n])^2));
               IF nj==1;
                  Zjnjd=Zjnj;
               ELSE;
                  Zjnjd=Zjnjd|Zjnj;  @ Distance matrix for         @
                                     @ the subset data.            @
               ENDIF;
               nj=nj+1;
            ENDO;
            CLEAR Zjt;
/* -------------------------------------------------------------- */
/* Locating the distances that define the KNN graph in the        */
/* distance matrix for the subset data and computing Delta Star   */
/* -------------------------------------------------------------- */
            Zjnjd=Zjnjd[knnind,1];
            DSall[j,1]=SUMC(ABS(Yjnfd-Zjnjd));  @ Delta Star       @
            trackDS[k,1]=Dall[j,1];
            k=k+1;
            j=j+1;
         ELSE;
            j=j+1;
         ENDIF;
      ENDO;
/* -------------------------------------------------------------- */
/* Deleting the variable whose omission yields the least value    */
/* of Delta.                                                      */
/* -------------------------------------------------------------- */
      leastDS[p-q+1,1]=MINC(trackDS);
      l=1;
      DO WHILE l<=p;
         IF DSall[l,1]==leastDS[p-q+1,1];
```

```
            vardel[p-q+1,l]=1;
            trackdel[l,1]=1;
            l=l+1;
         ELSE;
            l=l+1;
         ENDIF;
      ENDO;
      q=q-1;
   ENDO;
/* ------------------------------------ */
/* Identifying the retained variables   */
/* ------------------------------------ */
   slctv=ONES(1,M);    @ Contains the retained variables' indices  @
   k=1;
   l=1;
   DO WHILE l<=p;
      IF trackdel[l,1]==0;
         slctv[1,k]=l;
         k=k+1;
         l=l+1;
      ELSE;
         l=l+1;
      ENDIF;
   ENDO;
/* --------------------- */
/* Printing the results  */
/* --------------------- */
   LET VDSnames[2,1]=vardel leastDS;
   VDS=MERGEVAR(VDSnames);
   LET label[1,2]= Variable DeltaS;
   PRINT label;
   PRINT;
   PRINT VDS;
   PRINT;
   PRINT "Selected subset of variables:";
   PRINT;
   PRINT slctv;
   smplnumb=smplnumb+1;
   time2=TIME;                     @ Final time    @
/* --------------------------------------------------------------- */
/* Converting and printing the computational time in seconds       */
/* --------------------------------------------------------------- */
   hr2=(time2[1,1])*3600;
   hr1=(time1[1,1])*3600;
   min2=(time2[2,1])*60;
   min1=(time1[2,1])*60;
   sec2=hr2+min2+(time2[3,1])+((time2[4,1])/100);
   sec1=hr1+min1+(time1[3,1])+((time1[4,1])/100);
   comptime=sec2-sec1;             @ Computational time in seconds @
   PRINT;
   PRINT comptime;
   PRINT;
   PRINT "End of Session [" smplnumb "]";
   PRINT;
ENDO;
```

```
/* -------------------------------------------------------------- */
/*                            Program 9                           */
/*                            ---------                           */
/*                                                                */
/*  The Delta Star criterion (using the K-Minimum Spanning Tree)  */
/*  ------------------------------------------------------------- */
/*                      Backward elimination                      */
/*                      --------------------                      */
/*                                                                */
/* This program performs variable selection in PCA using the      */
/* Delta Star criterion and the backward elimination procedure   */
/* for a given data set.                                          */
/*                                                                */
/* Input:                                                         */
/* -----                                                          */
/*                                                                */
/* 1. Number of Training samples                                  */
/* 2. Sample size                                                 */
/* 3. Number of Variables                                         */
/* 4. Data set                                                    */
/* 5. Number of PCs to be used for variable selection             */
/*                                                                */

/* Output:                                                        */
/* ------                                                         */
/*                                                                */
/* 1. Deleted variables and the corresponding Delta Star values  */
/* 2. Selected subset of variables                                */
/* 3. Computational time (in seconds)                             */
/*                                                                */
/* -------------------------------------------------------------- */
   NEW;                       @ Clearing the computer memory       @
   #INCLUDE SVD.H;            @ Loading the procedure for SVD      @
/* ------------------------------------ */
/* Loading the data into the program    */
/* ------------------------------------ */
   PRINT "Program execution in progress...";
   PRINT;
   PRINT "Type in the number of training samples";
   PRINT;
   tns=CON(1,1);
   PRINT;
   PRINT "Type in the sample size";
   PRINT;
   rsize=CON(1,1);
   PRINT;
   f11=11;
   fname2="b:\\n25\\model1";       @ External data file            @
   OPEN f11=^fname2 FOR READ;      @ Opening data file for read    @
   PRINT "Type in the number of PCs to be used";
   PRINT;
   M=CON(1,1);
smplnumb=1;
DO WHILE smplnumb<=tns;
   time1=TIME;                     @ Initial time  @
   Z=READR(f11,rsize);
   n=ROWS(Z);
   p=COLS(Z);
```

```
/* -------------------------------------------------------------- */
/* Mean-centering the data, and standardizing it (if necessary)   */
/* -------------------------------------------------------------- */
   PROC STD(Y);
        LOCAL mean,ctrans,deviat;
        mean=MEANC(Y);
        ctrans=((-1)*mean)+Y';
        deviat=STDC(Y);
        RETP((ctrans./deviat)');
   ENDP;
   X=STD(Z);
   CLEAR Z;
/* -------------------------------------------------------------- */
/* Computing and standardizing the PC scores for complete data    */
/* -------------------------------------------------------------- */
   {Uf,Sf,Vf}=SVD2(X);
   CLEAR Vf;
   UfSf=Uf*Sf;
   CLEAR Uf,Sf;
   Yjunst=UfSf[.,1:M];          @ PC scores                        @
   CLEAR UfSf;
   Yj=((Yjunst')./STDC(Yjunst))';  @ Standardized PC scores        @
   Yjt=Yj';
   CLEAR Yj;

/* -------------------------------------------------------------- */
/* Computing the distance matrix for the complete data            */
/* -------------------------------------------------------------- */
   nf=1;
   DO UNTIL nf>n-1;
      Yjnf=SQRT(SUMC((Yjt[.,nf]-Yjt)^2));
      IF nf==1;
         dist=Yjnf;
      ELSE;
         dist=dist¦Yjnf;    @ Distance matrix for complete data    @
      ENDIF;
      nf=nf+1;
   ENDO;
   CLEAR Yjt;
/* -------------------------------------------------------------- */
/* Constructing the KMST from the complete data configuration     */
/* -------------------------------------------------------------- */
   big=).5*n*(n-1);
   itree=CUMSUMC(ONES(n-1,1));
   jtree=-n*ONES(n,1);
   jtree[n,1]=0;
   i=1;
   DO WHILE i<=n-1;
      distmn=big;
      j=1;
      DO WHILE j<=n-1;
         inode=jtree[j,1];
         IF inode<=0;
            d=dist[-inode,j];
              IF d<distmn;
                 distmn=d;
                 imin=j;
                 j=j+1;
              ELSE;
                 j=j+1;
              ENDIF;
         ELSE;
              j=j+1;
         ENDIF;
      ENDO;
      jtree[imin,1]=-jtree[imin,1];
      j=1;
      DO WHILE j<=n-1;
         inode=jtree[j,1];
         IF inode<=0;
            IF dist[j,imin]<dist[j,-inode];
               jtree[j,1]=-imin;
               j=j+1;
            ELSE;
               j=j+1;
            ENDIF;
         ELSE;
            j=j+1;
         ENDIF;
      ENDO;
   i=i+1;
   ENDO;
/* -------------------------------------------------------------- */
/* Obtaining and storing the indices of the KMST                  */
/* -------------------------------------------------------------- */
   Yjnfd=(dist[1,2:n])';
   i=2;
   DO WHILE i<=n-1;
      Yjnfd=Yjnfd|(dist[i,(i+1):n])';
      i=i+1;
   ENDO;
   treeind=ONES(n-1,1);
   i=1;
   DO WHILE i<=n-1;
      treeind[i,1]=INDNV(dist[itree[i,1],jtree[i,1]],Yjnfd);
      i=i+1;
   ENDO;
   Yjnfd=Yjnfd[treeind,1];
/* --------------------- */
/* Declaring variables   */
/* --------------------- */
   trackdel=ZEROS(p,1);       @ Tracks deleted variables in X.     @
   leastDS=ONES(p-M,1);       @ Stores the least Delta Star value. @
   vardel=ONES(p-M,1);        @ Identifies variables to be deleted @
   q=p;                       @ Initial subset size.               @
/* -------------------------------------------------------------- */
/* Computing the Delta Star criterion for each variable omitted   */
/* in turn from the data matrix X.                                */
/* -------------------------------------------------------------- */
/* DO WHILE q>M;
      trackDS=ONES(q,1);      @ Stores the Delta Star values.      @
      DSall=-1000*(ONES(p,1); @ Delta Star values for variables    @
                              @ omitted from X, indexed in the     @
                              @ order in which the variables       @
                              @ appear in X.                       @
      k=1;
      j=1;
      DO WHILE j<=p;
         IF trackdel[j,1]==0;
            trackdel[j,1]=1;  @ Indexing the j-th variable for     @
```

```
                              @ omission.                           @
            Xdelj=(DELIF(X',trackdel))';     @ New X.               @
            trackdel[j,1]=0;  @ Indexing omitted variable for       @
                              @ replacement.                        @
            {Udelj,Sdelj,Vdelj}=SVD2(Xdelj); @ SVD of new X.        @
            CLEAR Xdelj, Vdelj;
            UjSj=Udelj*Sdelj;
            CLEAR Udelj,Sdelj;
            Zjunst=UjSj[.,1:M];    @ PC scores for new X            @
            CLEAR UjSj;
            Zj=((Zjunst')./STDC(Zjunst))';   @ Standardized PC sc. @
            Zjt=Zj';
            CLEAR Zj;
/* --------------------------------------------------- */
/* Computing the distance matrix for the subset data   */
/* --------------------------------------------------- */
            nj=1;
            DO UNTIL nj>n-1;
               Zjnj=SQRT(SUMC((Zjt[.,nj]-Zjt[.,nj+1:n])^2));
               IF nj==1;
                  Zjnjd=Zjnj;
               ELSE;
                  Zjnjd=Zjnjd|Zjnj;  @ Distance matrix for          @
                                     @ the subset data.             @
               ENDIF;
               nj=nj+1;
            ENDO;
            CLEAR Zjt;
/* -------------------------------------------------------------- */
/* Locating the distances that define the KMST graph in the       */
/* distance matrix for the subset data and computing Delta Star   */
/* -------------------------------------------------------------- */
            Zjnjd=Zjnjd[treeind,1];
            DSall[j,1]=SUMC(ABS(Yjnfd-Zjnjd));  @ Delta Star        @
            trackDS[k,1]=Dall[j,1];
            k=k+1;
            j=j+1;
         ELSE;
            j=j+1;
         ENDIF;
      ENDO;
/* -------------------------------------------------------------- */
/* Deleting the variable whose omission yields the least value    */
/* of Delta.                                                      */
/* -------------------------------------------------------------- */
      leastDS[p-q+1,1]=MINC(trackDS);
      l=1;
      DO WHILE l<=p;
         IF DSall[l,1]==leastDS[p-q+1,1];
            vardel[p-q+1,1]=l;
            trackdel[l,1]=1;
            l=l+1;
         ELSE;
            l=l+1;
         ENDIF;
      ENDO;
      q=q-1;
   ENDO;
```

```
/* ------------------------------------ */
/* Identifying the retained variables   */
/* ------------------------------------ */
   slctv=ONES(1,M);    @ Contains the retained variables' indices  @
   k=1;
   l=1;
   DO WHILE l<=p;
      IF trackdel[l,1]==0;
         slctv[1,k]=l;
         k=k+1;
         l=l+1;
      ELSE;
         l=l+1;
      ENDIF;
   ENDO;
/* --------------------- */
/* Printing the results  */
/* --------------------- */
   LET VDSnames[2,1]=vardel leastDS;
   VDS=MERGEVAR(VDSnames);
   LET label[1,2]= Variable DeltaS;
   PRINT label;
   PRINT;
   PRINT VDS;
   PRINT;
   PRINT "Selected subset of variables:";
   PRINT;
   PRINT slctv;
   smplnumb=smplnumb+1;
   time2=TIME;                     @ Final time    @
/* --------------------------------------------------------------- */
/* Converting and printing the computational time in seconds      */
/* --------------------------------------------------------------- */
   hr2=(time2[1,1])*3600;
   hr1=(time1[1,1])*3600;
   min2=(time2[2,1])*60;
   min1=(time1[2,1])*60;
   sec2=hr2+min2+(time2[3,1])+((time2[4,1])/100);
   sec1=hr1+min1+(time1[3,1])+((time1[4,1])/100);
   comptime=sec2-sec1;             @ Computational time in seconds @
   PRINT;
   PRINT comptime;
   PRINT;
   PRINT "End of Session [" smplnumb "]";
   PRINT;
ENDO;
```

Following is a program written in GAUSS (1988) to illustrate the selection of variables in PCA using our various selection criteria incorporated into the stepwise procedure. For convinience, we present only the program that corresponds to the M^2-Procrustes criterion of Krzanowski (1987). Similar programs which correspond to the proposed criteria for variable selection, namely, $\hat{\gamma}_5$, $\hat{\gamma}_6$, δ and δ^* were also written in GAUSS (1988) to conduct the Monte Carlo study in Chapter 7 and the analyses of real data sets in Chapter 8, and copies of these programs are available from either the author or the principal project supervisor.

```
/* -------------------------------------------------------------- */
/*                          Program 10                            */
/*                          ----------                            */
/*                                                                */
/*     Krzanowski's (1987) M-squared-Procrustes criterion         */
/*     ---------------------------------------------------        */
/*                      Stepwise procedure                        */
/*                      ------------------                        */
/*                                                                */
/* This program performs variable selection in PCA using the M-   */
/* squared criterion and the stepwise selection procedure for     */
/* a given data set.                                              */
/*                                                                */
/* Input:                                                         */
/* -----                                                          */
/*                                                                */
/* 1. Number of Training samples                                  */
/* 2. Sample size                                                 */
/* 3. Number of Variables                                         */
/* 4. Data set                                                    */
/* 5. Number of PCs to be used for variable selection             */
/*                                                                */
/* Output:                                                        */
/* ------                                                         */
/*                                                                */
/* 1. Deleted variables and the corresponding M-squared values    */
/* 2. Selected subset of variables                                */
/* 3. Computational time (in seconds)                             */
/*                                                                */
/* -------------------------------------------------------------- */
   NEW;                        @ Clearing the computer memory      @
   #INCLUDE SVD.H;             @ Loading the procedure for SVD     @
/* ------------------------------------ */
/* Loading the data into the program    */
/* ------------------------------------ */
   PRINT "Program execution in progress...";
   PRINT;
   PRINT "Type in the number of training samples";
   PRINT;
   tns=CON(1,1);
   PRINT;
   PRINT "Type in the sample size";
   PRINT;
   rsize=CON(1,1);
   PRINT;
   f11=11;
   fname2="b:\\n25\\model1";       @ External data file            @
   OPEN f11=^fname2 FOR READ;      @ Opening data file for read    @
   PRINT "Type in the number of PCs to be used";
   PRINT;
   M=CON(1,1);
smplnumb=1;
DO WHILE smplnumb<=tns;
   time1=TIME;                     @ Initial time  @
   Z=READR(f11,rsize);
   n=ROWS(Z);
   p=COLS(Z);
/* ------------------------------------------------------------- */
/* Mean-centering the data, and standardizing it (if necessary)   */
/* ------------------------------------------------------------- */
   PROC STD(Y);
        LOCAL mean,ctrans,deviat;
        mean=MEANC(Y);
        ctrans=((-1)*mean)+Y';
        deviat=STDC(Y);
        RETP((ctrans./deviat)');
   ENDP;
   X=(((-1)*MEANC(Z))+Z')';
   CLEAR Z;
/* ------------------------------------------------------------- */
/* Variable declaration and computing PC scores for complete data */
/* ------------------------------------------------------------- */
   {Uf,Sf,Vf}=SVD2(X);
   CLEAR Vf;
   UfSf=Uf*Sf;
   CLEAR Uf,Sf;
   Yj=UfSf[.,1:M];
   CLEAR UfSf;
   trackdel=ZEROS(p,1);       @ Tracks deleted variables in X.      @
   lstM22=ONES(p-M,1);        @ Stores the least M-squared value.   @
   vardel=ONES(p-M,1);        @ Identifies variables to be deleted  @
   q=p;                       @ Initial subset size.                @
/* ------------------------------------------------------------- */
/* Computing the M-squared criterion for each variable omitted    */
/* in turn from the data matrix X.                                */
/* ------------------------------------------------------------- */
/* DO WHILE q>M;
      trackM2=ONES(q,1);      @ Stores M-squared values.            @
      M2all=-1000*(ONES(p,1); @ M-squared values for variables      @
                              @ omitted from X, indexed in the      @
                              @ order in which the variables        @
                              @ appear in X.                        @
      k=1;
      j=1;
      DO WHILE j<=p;
         IF trackdel[j,1]==0;
            trackdel[j,1]=1;  @ Indexing the j-th variable for      @
```

```
                              @ omission.                          @
            Xdelj=(DELIF(X',trackdel))';     @ New X.              @
            trackdel[j,1]=0;  @ Indexing omitted variable for      @
                              @ replacement.                       @
            {Udelj,Sdelj,Vdelj}=SVD2(Xdelj); @ SVD of new X.       @
            CLEAR Xdelj, Vdelj;
            UjSj=Udelj*Sdelj;
            CLEAR Udelj,Sdelj;
            Zj=UjSj[.,1:M];
            CLEAR UjSj;
            M2all[j,1]=SUMC(DIAG(Yj'*Yj+Zj'*Zj))
                       -2*SUMC(SVD(Zj'*Yj));    @ Computing M-squ. @
            CLEAR Zj;
            trackM2[k,1]=M2all[j,1];
            k=k+1;
            j=j+1;
         ELSE;
            j=j+1;
         ENDIF;
      ENDO;
/* -------------------------------------------------------------- */
/* Deleting the variable whose omission yields the least M-squ.  */
/* -------------------------------------------------------------- */
      lstM22[p-q+1,1]=MINC(trackM2);
      l=1;
      DO WHILE l<=p;
         IF M2all[l,1]==lstM22[p-q+1,1];
            vardel[p-q+1,1]=l;
            M2delv=lstM22[p-q+1,1];  @ M-squared value for         @
                                     @ deleted variable            @
            trackdel[l,1]=1;
            l=l+1;
         ELSE;
            l=l+1;
         ENDIF;
      ENDO;


/* -------------------------------------------------------------- */
/* Checking whether each deleted variable should re-enter and     */
/* adding them back into the system if necessary.                 */
/* -------------------------------------------------------------- */
      IF q<p;
         lstM2all=-1000*(ONES(p,1)); @ Stores least M-squared      @
                                     @ values, indexed in the      @
                                     @ order in which the          @
                                     @ corresponding variables     @
                                     @ appear in the data matrix X @
         lstM21=ONES(p-q+1,1);  @ Stores the least M-squared value @
         i=1;
         s=1;
         DO WHILE i<=p;
            IF trackdel[i,1]==1; @ If a variable has been deleted, @
               trackdel[i,1]=0;  @ undelete it.                    @
               trackM2=ONES(q,1);
               M2all=-1000*(ONES(p,1));
               k=1;
               j=1;
               DO WHILE j<=p;
                  IF trackdel[j,1]==0;
```

```
                     trackdel[j,1]=1;
                     Xdelj=(DELIF(X',trackdel))';
                     trackdel[j,1]=0;
                     {Udelj,Sdelj,Vdelj}=SVD2(Xdelj);
                     CLEAR Xdelj,Vdelj;
                     UjSj=Udelj*Sdelj;
                     CLEAR Udelj,Sdelj;
                     Zj=UjSj[.,1:M];
                     CLEAR UjSj;
                     M2all[j,1]=SUMC(DIAG(Yj'*Yj+Zj'*Zj))
                                -2*SUMC(SVD(Zj'*Yj));
                     CLEAR Zj;
                     trackM2[k,1]=M2all[j,1];
                     k=k+1;
                     j=j+1;
                  ELSE;
                     j=j+1;
                  ENDIF;
               ENDO;
               lstM2all[i,1]=MINC(trackM2);
               lstM21[s,1]=lstM2all[i,1];
               trackdel[i,1]=1;
               s=s+1;
               i=i+1;
            ELSE;
               i=i+1;
            ENDIF;
         ENDO;
/* -------------------------------------------------------------- */
/* If the return of one of the deleted variables yields a value   */
/* of M-squared smaller than the current least value, then re-add */
/* that variable and delete the variable whose omission yields    */
/* the least M-squared value in the presence of the re-added var. */
/* -------------------------------------------------------------- */
         IF MINC(lstM21)<M2delv;
            l=1;
            DO WHILE l<=p;
               IF lstM2all[l,1]==MINC(lstM21);
                  retv=l;                    @ Returning variable  @
                  l=l+1;
               ELSE;
                  l=l+1;
               ENDIF;
            ENDO;
            trackdel[retv,1]=0; @ Indexing the variable for return @
            trackM2=ONES(q,1);
            M2all=-1000*(ONES(p,1));
            k=1;
            j=1;
            DO WHILE j<=p;
               IF trackdel[j,1]==0;
                  trackdel[j,1]=1;
                  Xdelj=(DELIF(X',trackdel))';
                  trackdel[j,1]=0;
                  {Udelj,Sdelj,Vdelj}=SVD2(Xdelj);
                  CLEAR Xdelj,Vdelj;
                  UjSj=Udelj*Sdelj;
                  CLEAR Udelj,Sdelj;
                  Zj=UjSj[.,1:M];
                  CLEAR UjSj;
                  M2all[j,1]=SUMC(DIAG(Yj'*Yj+Zj'*Zj))
                                -2*SUMC(SVD(Zj'*Yj));
                  CLEAR Zj;
                  trackM2[k,1]=M2all[j,1];
                  k=k+1;
                  j=j+1;
               ELSE;
                  j=j+1;
               ENDIF;
            ENDO;
            lstM22[p-q+1,1]=MINC(trackM2);
            l=1;
            DO WHILE l<=p;
               IF M2all[l,1]==lstM22[p-q+1,1];
                  vardel[p-q+1,1]=l;
                  trackdel[l,1]=1;
                  l=l+1;
               ELSE;
                  l=l+1;
               ENDIF;
            ENDO;
            q=q-1;
         ELSE;
            q=q-1;
         ENDIF;
   ENDO;
/* --------------------- */
/* Printing the results  */
/* --------------------- */
   LET VM2names[2,1]=vardel lstM22;
   VM2=MERGEVAR(VM2names);
   LET label[1,2]= Variable M-squared;
   PRINT label;
   PRINT;
   PRINT VM2;
   PRINT;
   PRINT "Selected subset of variables:";
   PRINT;
   PRINT slctv;
   smplnumb=smplnumb+1;
   time2=TIME;                     @ Final time    @
/* -------------------------------------------------------------- */
/* Converting and printing the computational time in seconds      */
/* -------------------------------------------------------------- */
   hr2=(time2[1,1])*3600;
   hr1=(time1[1,1])*3600;
   min2=(time2[2,1])*60;
   min1=(time1[2,1])*60;
   sec2=hr2+min2+(time2[3,1])+((time2[4,1])/100);
   sec1=hr1+min1+(time1[3,1])+((time1[4,1])/100);
   comptime=sec2-sec1;             @ Computational time in seconds @
   PRINT;
   PRINT comptime;
   PRINT;
   PRINT "End of Session [" smplnumb "]";
   PRINT;
ENDO;
```

```
/* -------------------------------------------------------------- */
/*                           Program 11                           */
/*                           ----------                           */
/*                                                                */
/*                  Squared multiple correlation                  */
/*                  ----------------------------                  */
/*           for all possible subsets (subset size = 3)           */
/*           ------------------------------------------           */
/*                                                                */
/* This program computes the squared multiple correlation         */
/* between three retained variables and each of the discarded    */
/* variables for all possible subsets of retained variables.      */
/*                                                                */
/* Input:                                                         */
/* -----                                                          */
/*                                                                */
/* 1. Sample size                                                 */
/* 2. Number of Variables                                         */
/* 3. Data set                                                    */
/*                                                                */
/* Output:                                                        */
/* ------                                                         */
/*                                                                */
/* 1. Squared multiple correlation for each                       */
/*    of all possible subsets                                     */
/*                                                                */
/* -------------------------------------------------------------- */
/* ------------------------------------ */
/* Loading the data into the program    */
/* ------------------------------------ */
   PRINT "Program execution in progress...";
   PRINT;
   PRINT "Type in the sample size";
   PRINT;
   n=CON(1,1);
   PRINT;
   PRINT "Type in the number of variables";
   PRINT;
   p=CON(1,1);
   PRINT;
   M=3;          @ Size of the subset of retained variables        @
   f12=12;
   fname2="b:\\n100000\\model1";   @ External data file            @
   OPEN f12=^fname2 FOR READ;      @ Opening data file for read    @
   z=READR(f12,n);
   subset=ONES(1,M);
   delv=-1000*(ONES(p,1));  @ Deleted variable                     @
/* -------------------------------------------------------------- */
/* Computing the squared multiple correlation for each of all     */
/* possible subsets.                                              */
/* -------------------------------------------------------------- */
   j=1;
   DO WHILE j<=p-2;
      subset[1,1]=j;
      delv[j,1]=j;
      descr2=z[.,j];
      k=j+1;
      DO WHILE k<=p-1;
         subset[1,2]=k;
         delv[k,1]=k;
      descr3=z[.,k];
      l=k+1;
      DO WHILE l<=p;
         subset[1,3]=l;
         delv[l,1]=1;
         descr=descr2~descr3~zz[.,l];
         rmsqu=ONES(1,p-M);
         r=1;
         s=1;
         DO WHILE r<=p;
            IF delv[r,1]==-1000;
               pred=z[.,r];
               pd=pred~descr;
               cpd=pd'-MEANC(pd);
               cov=(1/(n-1))*(cpd*cpd');  @ Covariance matrix   @
                                          @ the subset and each @
                                          @ of the deleted      @
                                          @ variables           @
               a11=cov[1,1];
               e12=cov[1,2:M+1];
               e22=cov[2:M+1,2:M+1];
               rmsqu[1,s]=(e12*INV(e22)*e12')/a11;  @ Squared   @
                                                    @ multiple  @
                                                    @ corr.     @
               s=s+1;
               r=r+1;
            ELSE;
               r=r+1;
            ENDIF;
         ENDO;
         PRINT subset;
         PRINT;
         PRINT rmsqu;
         PRINT;
         delv[l,1]=-1000;
         l=l+1;
      ENDO;
      delv[k,1]=-1000;
      k=k+1;
   ENDO;
   delv[j,1]=-1000;
   j=j+1;
ENDO;
PRINT;
PRINT "End of session";
```

```
/* -------------------------------------------------------------- */
/*                           Program 12                           */
/*                           ----------                           */
/*                                                                */
/*                  Squared multiple correlation                  */
/*                  ----------------------------                  */
/*           for all possible subsets (subset size = 4)           */
/*           ------------------------------------------           */
/*                                                                */
/* This program computes the squared multiple correlation         */
/* between four retained variables and each of the discarded     */
/* variables for all possible subsets of retained variables.      */
/*                                                                */
/* Input:                                                         */
/* -----                                                          */
/*                                                                */
/* 1. Sample size                                                 */
/* 2. Number of Variables                                         */
/* 3. Data set                                                    */
/*                                                                */
/* Output:                                                        */
/* ------                                                         */
/*                                                                */
/* 1. Squared multiple correlation for each                       */
/*    of all possible subsets                                     */
/*                                                                */
/* -------------------------------------------------------------- */
/* ------------------------------------ */
/* Loading the data into the program    */
/* ------------------------------------ */
   PRINT "Program execution in progress...";
   PRINT;
   PRINT "Type in the sample size";
   PRINT;
   n=CON(1,1);
   PRINT;
   PRINT "Type in the number of variables";
   PRINT;
   p=CON(1,1);
   PRINT;
   M=4;          @ Size of the subset of retained variables        @
   f12=12;
   fname2="b:\\m100000\\model4";   @ External data file            @
   OPEN f12=^fname2 FOR READ;      @ Opening data file for read    @
   z=READR(f12,n);
   subset=ONES(1,M);
   delv=-1000*(ONES(p,1));  @ Deleted variable                     @
/* -------------------------------------------------------------- */
/* Computing the squared multiple correlation for each of all     */
/* possible subsets.                                              */
/* -------------------------------------------------------------- */
   i=1;
   DO WHILE i<=p-3;
      subset[1,1]=i;
      delv[i,1]=i;
      descr1=z[.,i];
      j=i+1;
      DO WHILE j<=p-2;
         subset[1,2]=j;
         delv[j,1]=j;
      descr2=z[.,j];
      k=j+1;
      DO WHILE k<=p-1;
         subset[1,3]=k;
         delv[k,1]=k;
         descr3=z[.,k];
         l=k+1;
         DO WHILE l<=p;
            subset[1,4]=l;
            delv[l,1]=l;
            descr=descr1~descr2~descr3~zz[.,l];
            rmsqu=ONES(1,p-M);
            r=1;
            s=1;
            DO WHILE r<=p;
               IF delv[r,1]==-1000;
                  pred=z[.,r];
                  pd=pred~descr;
                  cpd=pd'-MEANC(pd);
                  cov=(1/(n-1))*(cpd*cpd');
                                          @ Covariance matrix   @
                                          @ the subset and each @
                                          @ of the deleted      @
                                          @ variables           @
                  a11=cov[1,1];
                  e12=cov[1,2:M+1];
                  e22=cov[2:M+1,2:M+1];
                  rmsqu[1,s]=(e12*INV(e22)*e12')/a11;
                                                    @ Squared   @
                                                    @ multiple  @
                                                    @ corr.     @
                  s=s+1;
                  r=r+1;
               ELSE;
                  r=r+1;
               ENDIF;
            ENDO;
            PRINT subset;
            PRINT;
            PRINT rmsqu;
            PRINT;
            delv[l,1]=-1000;
            l=l+1;
         ENDO;
         delv[k,1]=-1000;
         k=k+1;
      ENDO;
      delv[j,1]=-1000;
      j=j+1;
   ENDO;
   delv[i,1]=-1000;
   i=i+1;
ENDO;
PRINT;
PRINT "End of session";
```