AD-A086 521 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 12/1
CLASSIFICATION TECHNIQUES FOR MULTIVARIATE DATA ANALYSIS.(U)
MAR 80 J K LEE

UNCLASSIFIED NL

1 of 2
AD
A086521

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

CLASSIFICATION TECHNIQUES FOR MULTIVARIATE DATA ANALYSIS

by

Jin Ki Lee

March 1980

Thesis Advisor: F. R. Richards

Approved for public release; distribution unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A086521	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 Classification Techniques for Multivariate Data Analysis.		5. TYPE OF REPORT & PERIOD COVERED 9 Master's Thesis, ~~March 1980~~
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Jin Ki Lee		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	11	12. REPORT DATE 28 Mar[redacted]80
		13. NUMBER OF PAGES ~~120~~
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940 12 121		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

251450

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

multivariate analysis, principal components analysis, variance-covariance matrix, correlation, lagranian technique discriminant analysis, distance measure (metric), hierarchical clustering, nonhierarchical clustering, similarity matrix

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The multivariate analysis techniques of cluster analysis, principal components analysis, and discriminant analysis are examined in this thesis. The theory and applications of each of the techniques are discussed. Computer software available at the Naval Postgraduate School is discussed and sample jobs are included.

A hierarchical cluster analysis algorithm, available in the IMSL software package, is applied to a set of data extracted from a

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

group of subjects for the purpose of partitioning a collection of 26 attributes of a weapon system into six clusters of super-attributes.

A nonhierarchical clustering procedure, principal components analysis, and discriminant analysis were all applied to a collection of data on tanks considering of twenty-four observations of ten attributes of tanks. The cluster analysis shows that the tanks cluster somewhat naturally by nationality. The principal components analysis and the discriminant analysis show that tank weight is the single most important discriminator among nationality.

Approved for public release, distribution unlimited

Classification Techniques for Multivariate Data Analysis

by

Jin Ki Lee
Lieutenant Colonel, Korean Army
B.S., Korean Military Academy, 1966

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1980

Author Lee jinki

Approved by: FRRichards
Thesis Advisor

Donald R Barr
Co-Advisor or Second Reader

Chairman, Department of Operations Research

DA Schrady
Dean of Information and Policy Sciences

ABSTRACT

The multivariate analysis techniques of cluster analysis, principal components analysis, and discriminant analysis are examined in this thesis. The theory and applications of each of the techniques are discussed. Computer software available at the Naval Postgraduate School is discussed and sample jobs are included.

A hierarchical cluster analysis algorithm, available in the IMSL software package, is applied to a set of data extracted from a group of subjects for the purpose of partitioning a collection of 26 attributes of a weapon system into six clusters of superattributes.

A nonhierarchical clustering procedure, principal components analysis, and discriminant analysis were all applied to a collection of data on tanks considering of twenty-four observations of ten attributes of tanks. The cluster analysis shows that the tanks cluster somewhat naturally by nationality. The principal components analysis and the discriminant analysis show that tank weight is the single most important discriminator among nationality.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. DISCUSSION OF MULTIVARIATE DATA ANALYSIS

A. INTRODUCTION

As a set of statistical techniques, multivariate data analysis is concerned with data collected on several dimensions of the same observations. Techniques can be used for many purpose in the behavioral, mathematical, and administrative sciences - ranging from rigidly controlled experiments to explain relationships assumed to be present in a large mass of data to attempts to cluster similar elements or to find functions of the variables that will best discriminate among preselected subpopulations of the observations.

The heart of any multivariate analysis consists of the data matrix. This matrix is a table that gives the results of a number of observations on a number of variables simultaneously (Table I).

Illustrative Data Matrix

	Variables				
Observations	1	2	3	j	p
1	x_{11}	x_{12}	x_{13}	x_{1j}	x_{1p}
2	x_{21}	x_{22}	x_{23}	x_{2j}	x_{2p}
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
i	x_{i1}	x_{i2}	x_{i3}	x_{1j}	x_{ip}
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
n	x_{n1}	x_{n2}	x_{n3}	x_{nj}	x_{np}

TABLE I.

The table consists of a set of observations (the n rows) and a set of measurements on those observations (the p columns). Cell entries represent the value x_{ij} of observation i on variable j. The values are characteristics of the observations and serve to define the observations in any specific study. The cell values may consist of nominal, ordinal, interval, or ratio-scaled measurements, or varicus combinations of these across columns.

In a general sense "multivariate" analysis would concern two main features:

1. The multivariate character lies in the multiplicity of the p variables, not in the size of the set n .
2. The variables are dependent among themselves so that we can not split off one or more from the others and consider it by itself. The variables must be considered together.

There are three characteristics often used as a basis for the classification of multivariate analysis:

1. whether one's principal focus is on the objects or on the variables of the data matrix;
2. whether the data matrix is partitioned into criterion and independent subsets, and the number of variables in each;
3. whether the cell values represent nominal, ordinal, or interval scale measurements.

This classification results in four major subdivisions of interest:

1. single criterion, multiple predictor association, including multiple regression, analysis of variance and covariance, and two-group discriminant analysis;

2. multiple criterion, multiple predictor association, including canonical correlation, multivariate analysis of variance and covariance, and multiple discriminant analysis;

3. analysis of variable interdependence, including factor analysis, multidimensional scaling, and other types of dimension-reducing methods;

4. analysis of interobject similarity, including cluster analysis and other types of grouping procedures.

The first two categories involve dependence structures where the data matrix is partitioned into criterion and independent subsets; in both cases interest is focused on the variables. The last two categories are concerned with interdependence - either focusing on variables or on observations. Within each of four categories, various techniques are differentiated in terms of the type of scale assumed.

In this research, we consider only the following techniques of multivariate analysis:

1. Principal components analysis
2. Discriminant analysis
3. Cluster analysis

II. PRINCIPAL COMPONENTS ANALYSIS

The basic idea of principal components analysis is to describe the dispersion of an array of n points in p-dimensional space by introducing a new set of orthogonal linear coordinates so that the sample variances of the given data points with respect to these derived coordinates are in decreasing order of magnitude. Thus the first principal component is such that the projection of given points onto it have maximum variance among all possible linear coordinates; the second principal component has maximum variance subject to being orthogonal to the first; and so on.

Suppose that the random variables X_1 , X_2 , ..., X_p of interest have a certain multivariate distribution with finite mean vector u and variance-covariance matrix Σ .

From this population a sample of n independent observation vectors has been drawn. The observation can be written as the usual $n$xp data matrix.

$$X = \begin{bmatrix} x_{11} & \cdots\cdots\cdots & x_{ip} \\ \vdots & & \vdots \\ x_{n1} & \cdots\cdots\cdots & x_{np} \end{bmatrix} = \begin{bmatrix} X_1' \\ \vdots \\ X_n' \end{bmatrix} \tag{1}$$

The estimate of Σ will be the usual sample variance-covariance matrix S defined as follows:

$$S = \frac{1}{n - 1} A$$

$$A = \sum_{j=1}^{n} (X_j - \bar{X})(X_j - \bar{X})' \qquad (2)$$

The information we shall need for our principal components analysis will be contained in S . However, it will be necessary to make a choice of measures of dependence: should we work with the variances and covariances of the observations, and carry out the analysis in original unit of the responses, or would a more accurate picture of the dependence pattern be obtained if each x_{ij} were transformed to a standarized score

$$Z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}$$

and the correlation matrix R employed? The components obtained from S and R in general not the same, nor is it possible to pass from one solution to the other by a simple scaling of the coefficients.

If the responses are in widely different units (i.e., number of crew, weight in tons, speed in kilometer per hour, etc.) with large differences in the magnitudes, linear compounds of original quantities would have little meaning and standarized variates and correlation matrix should be employed. Conversely, if the responses are reasonably commensurable, the covariance form has a greater statistical appeal, for the i-th principal component is that linear compound of the responses which explains the i-th largest portion of the total response variance, and maximization of such total variance of standard scores is rather artificial.

The first principal component of the complex of sample values of the responses X_1, X_2, ..., X_p is the linear compound

$$Y_1 = a_{11}X_1 + \dots\dots\dots + a_{p1}X_p \tag{3}$$

whose coefficients a_{i1} are the elements of the eigenvector associated with the greatest eigenvalue λ_1 of the sample variance-covariance matrix of the responses. The a_{i1} are are unique up to multiplication by a scale factor, and if they are scaled so that $a'_1a_1 = 1$, the eigenvalue λ_1 is interpretable as the sample variance of Y_1.

Numerical representation of the first principal component is to find the vector A_1 such that

$$\begin{aligned} Y_1 &= a_{11}X_1 + \dots\dots\dots + a_{p1}X_p \\ &= A_1'X \end{aligned} \tag{4}$$

which maximizes sample variance

$$\begin{aligned} s_{Y_1}^2 &= \sum_{i=1}^{p} \sum_{j=1}^{p} a_{i1}a_{j1}s_{ij} \\ &= A'_1SA_1 \end{aligned} \tag{5}$$

for all coefficient vectors normalized so that $A'_1A_1 = 1$. To determine the coefficients, the normalization constraint is introduced by means of Lagrange multiplier and the resulting expression is differentiated with respect to A'_1 :

$$\begin{aligned} \frac{\partial}{\partial A_1}[S_{Y_1}^2 - \lambda_1(1 - A_1'A_1)] &= \frac{\partial}{\partial A_1}[A_1'SA_1 + \lambda_1(1 - A_1'A_1)] \\ &= 2(S - \lambda_1 I)A_1 \end{aligned} \tag{6}$$

The coefficients must satisfy the p simultaneous linear equations.

$$(S - \lambda_1 I)A_1 = 0 \tag{7}$$

If the solution to these equation is to be other than the null vector, the value of λ_1 must be chosen so that

$$|S - \lambda_1 I| = 0 \tag{8}$$

λ_1 is thus an eigenvalue of the variance-covariance matrix, and A_1 is its associated eigenvector. To determine which of the p eigenvalues should be used, premultiply the

the system of equation (7) by A_1' . Since $A_1'A_1 = 1$, it follows that

$$\lambda_1 = A_1' \, S \, A_1 = s_{Y_1}^{\;2}$$

But the coefficient vecotr was chosen to maximize this variance, and therefore, λ_1 must be the greatest eigenvalue of S .

The second principal component is that linear compound

$$Y_2 = a_{12}X_1 + \ldots\ldots\ldots + a_{p2} \, X_p \qquad (9)$$

whose coefficients have been chosen, subject to the constraints

$$\begin{aligned} A_2' \, A_2 &= 1 \\ A_1' \, A_2 &= 0 \end{aligned} \qquad (10)$$

so that the variance of Y_2 , $A_2' \, S \, A_2$, is a maximum. The first constraint is merely a scaling to assure the uniqueness of the coefficients, while the second requires that A_1 and A_2 be orthogonal.

The coefficients of the second component can also be found by the Lagrangian technique with two multipliers λ_2 and μ . Differentiating this with respect to A_2 gives:

$$\frac{\partial}{\partial A_2}[A_2' \, S \, A_2 + \lambda_2(1 - A_2' \, A_2) + \mu A_1' \, A_2]$$
$$= 2(S - \lambda_2 I)A_2 + \mu A_1 \qquad (11)$$

If the right-hand side is set equal to 0 and premultiplied by A_1' , it follows from the normalization and orthogonality conditions that

$$2 \, A_2' \, S \, A_2 + \mu = 0 \qquad (12)$$

Similar premultiplication of the equation (7) by A_2' implies that

$$A_2' \, S \, A_2 = 0 \qquad (13)$$

and hence $\mu = 0$. The second vector must satisfy

$$(S - \lambda_2 I)A_2 = 0 \qquad (14)$$

And it follows that the coefficients of the second component are thus the elements of the eigenvector corresponding to the second greatest eigenvalue. The remaining principal components are found in their turn in the same manner from the other eigenvectors.

Thus the j-th principal component of the sample of p-variate observations is the linear compound

$$Y_j = a_{1j}X_1 + \ldots\ldots\ldots + a_{pj}X_p \qquad (15)$$

whose coefficients are the elements of the eigenvector of the sample variance-covariance matrix S corresponding to the j-th largest eigenvalue λ_j . If $\lambda_i \neq \lambda_j$, the coefficients of the i-th and j-th components are necessarily orthogonal; if $\lambda_i = \lambda_j$, the elements can be chosen to be orthogonal, although an infinity of such orthogonal vectors exists. The sample variance of the j-th components is λ_j , and the total system variance is thus

$$\lambda_1 + \lambda_2 + \dots\dots\dots + \lambda_p = \text{tr } S \qquad (16)$$

The importance of the j-th component in a more parsimonious description of the system is measured by

$$\frac{\lambda_j}{\text{tr } S} \qquad (17)$$

which gives the fraction of the total variance contributed to the j-th component.

III. DISCRIMINANT ANALYSIS

A. INTRODUCTION

The basic idea of discriminant analysis consists of assigning an individual from a group of individuals to one of several known or unknown distinct propulations, on the basis of observations on several characters of the individual or group and a sample of observations on these characters from the populations if these are unknown.

Fisher (1936) was the first to suggest a linear function of variables representing different characters, hereafter called the linear discriminant function (discriminator) for classifying an individual into one of two populations. Later research extended the analysis to classification into one of k populations.

For the univariate case Fisher suggested a rule which classifies an observation x into the i-th univariate population if

$$X - \bar{X}_i = \min(X - \bar{X}_1 , X - \bar{X}_2) , \quad i = 1,2 \qquad (18)$$

where $\bar{X}_i$ is the sample mean based on a sample of size N_1 from i-th population. For two p-variate populations π_1 and π_2 (with the same covariance matrix) Fisher replaced the vector random variable by an optimum linear combination of its components obtained by maximizing the ratio of the difference of the expected values of a linear combination under π_1 and π_2 to its standard deviation. He then used his univariate discrimination method with this optimum linear combination of components as the random variable.

Rao (1948) considered the problem of classifying people into one of these populations castes of India. He assumed that each of the three populations could be characterized by four variables - structure (x_1), sitting height (x_2), nasal depth (x_3), and nasal height (x_4) - of each member of the population. On the basis of sample observations on these characters from the three populations the problem is to classify an individual with observation $X = (x_1,x_2,x_3,x_4)^T$ into one of three populations. He used a linear discriminator to obtain the solution.

B. THEORY

In general, the underlying assumptions of discriminant analysis are:

1. the groups being investigated are discrete and identifiable;
2. each observation in each group can be described by a set of measurements on p characteristics or variables;
3. these p variables are assumed to have a multivariate normal distribution in each population.

The purposes of discriminant analysis are:

1. to test for mean group differences and to describe the overlaps among groups;
2. to construct classification schemes based upon the set of p variables in order to assign previously unclassified observations to the appropriate groups.

Hence, the problem of studying the direction of group differences is, equivalently, a problem of finding a linear combination of the original independent variables that shows large differences in group means. In short, discriminant analysis is a method for determining scuh linear combinations.

The first step toward determining a linear combination of a set of variables such that several group means on this linear combination will differ widely among themselves, is to decide on a criterion for measuring such group-mean differences. Once a linear combination has been constructed, that means there is just a single tranformed variable. Hence, the F-ratio for testing the significance of the over all difference among several group means on a single variable suggests an appropriate criterion.

$$F = \frac{v'Bv}{v'Wv} = \lambda \qquad (19)$$

where $v' = (v_1, v_2, \ldots\ldots\ldots\ldots, v_p)$, a set of weight which maximizes λ .

$$B = \sum_{i=1}^{G} N_i(\bar{x}_{i.} - \bar{x}_{..})(\bar{x}_{i.} - \bar{x}_{..})'$$

$$W = \sum_{i=1}^{G} \sum_{I=1}^{n_i} (x_{ij} - \bar{x}_{i.})(x_{ij} - \bar{x}_{i.})'$$

x_{ij} is the jth observation vector in the i-th group.

$\bar{x}_{..}$ is the grand mean vector of the data.

G is the number of groups.

n_i is the number of observations in the ith group.

Prime notation indicates transpose.

This ratio λ, called the discriminant criterion, was originally proposed by Fisher in connection with his two-group discriminant function. Once a criterion for group differentiation has been determined, a set of weights, $(v_1 \; v_2 , \ldots\ldots\ldots, v_p)$, which maximizes this criterion, should be determined. This is accomplished by taking the partial derivative of λ with respect to each component v_i of v and setting the result equal to zero.

$$\frac{\partial\lambda}{\partial v} = \frac{2[(Bv)(v'Wv) - (v'Wv)(Wv)]}{(v'Wv)^2} \qquad (20)$$

$$= \frac{2(Bv - Wv)}{v'Wv} = 0$$

which is equivalent to

$$(B - \lambda W)v = 0$$
$$(W^{-1}B - \lambda I)v = 0 \qquad (21)$$

This equation is of the form

$$(A - \lambda I)v = 0 \qquad (22)$$

It's solution, yielding the eigenvalues λ_p and associated eigenvectors V_p of the matrix A, is therefore the same as in the principal components analysis, and thus the solved problem satisfies the problem of maximizing the discriminant criterion.

In the last equation, the number of non-zero eigenvalues of a square matrix A is equal to the rank of A. With $W^{-1}B$ playing the role of A, the number of non-zero eigenvalues depends on the rank of B, since the rank of the product of two matrices can not exceed the smaller of the two factor matrices' ranks, and W^{-1} (being nonsingular) must be of full rank p, while the rank of B is usually smaller than p. Thus it is possible to denote the rank of B by $r = \min(G-1,p)$.

From the fact that the eigenvalues λ_p are the values assumed by the discriminant criterion for linear combination using the elements of the corresponding eigenvectors P as combining weights, it is clear that the eigenvector

$V_1' = (v_{11}, v_{12}, \ldots\ldots\ldots, v_{1p})$ provides a set of weights such that the transformed variable

$$Y_1 = v_{11}X_1 + v_{12}X_2 + \ldots\ldots\ldots\ldots + v_{1p}X_p \qquad (23)$$

has the largest discriminant-criterion, λ, achievable by any linear combination of the p independent variables.

What are the properties of the remaining eigenvectors, $v_2, v_3, \ldots\ldots, v_p$? The second discriminant function $Y_2 = v_{21}X_1 + v_{22}X_2 + \ldots\ldots\ldots + v_{2p}X_p$ whose weights are the elements of the eigenvector v_2 associated with the second largest eigenvalue λ_2 of $W^{-1}B$ has the largest discriminant-criterion among those linear combinations of the X_i that are uncorrelated with the first discriminant function in the total sample observation. Its proof is analogous of that of princpal components analysis. Each discriminant function has a relative (or conditional) maximum value for its discriminant criterion. Therefore, it needs nonly to show that Y_2 is uncorrelated with Y_1. Noting that this correlation is proportional to $v_1'Tv_2$ (where $T = W + B$), we have to prove that $v_1'Tv_2 = 0$.

$$(B - \lambda_i W)v_i = 0 \quad \text{for each } i \qquad (24)$$

hence,

$$Bv_1 = \lambda_i Wv_1 \quad \text{and} \quad Bv_2 = \lambda_2 Wv_2$$

premultiplying these equations by v_2' and v_1' respectively,

$$\begin{aligned} v_2'Bv_1 &= \lambda_1 v_2'Wv_1 \\ v_1'Bv_2 &= \lambda_2 v_1'Wv_2 \end{aligned} \qquad (25)$$

taking the transpose of both sides of the first equation (B and W are symmetric)

$$v_1'BV_2 = \lambda_1 v_1'WV_2$$

thus

$$\lambda_1 V_1'WV_2 = \lambda_2 V_1'WV_2$$

$$(\lambda_1 - \lambda_s)V_1'WV_2 = 0$$

since $\lambda_1 \neq \lambda_2,\ V_1'WV_2 = 0$

therefore, $V_1'WV_2 = 0$ which means V_1 and V_2 are uncorrelated, and Y_2 has this property: its discriminant-criterion value, λ_2 , is the largest achievable by any linear combination of X's that is uncorrelated (in the total sample) with Y_1 . Similarly

$$Y_3 = v_{31}X_1 + v_{32}X_2 + \ldots\ldots\ldots + v_{3p}X_p \qquad (36)$$

has the largest possible discriminant-criterion value (λ_3) among all linear combinations of the X's that are uncorrelated with both Y_1 and Y_2; and so on until Y_r using the elements of V_r as weights, has the largest possible discriminant-criterion value among linear combinations that are uncorrelated with all the preceding linear combinations $Y_1, Y_2, \ldots\ldots, Y_{r-1}$. The linear combinations $Y_1, Y_2, \ldots\ldots, Y_r$ are called the first, second, ..., r th (linear) discriminant functions for optimally differentiating among the g given groups.

The situation here is reminiscent of principal components analysis. There, the dimension corresponding to the first component had maximum variance; the second-component dimension had maximum variance among those uncorrelated with the first; and so on. In discriminant analysis, the ratio of between-to within-groups sums-of-squares merely takes the place of variance as the criterion in determining the successive dimensions. However, an important difference between the dimensions identified in discriminant analysis and those in component analysis is that the former are generally not mutually orthogonal in the test space, even though they are uncorrelated. That is, the axis representing the discriminant functions are not a subset of axes obtainable by rigid rotation of the original system of p axes; the discriminant rotation in an oblique rotation.

Just as in the principal components analysis, the dimensions represented by the discriminant functions may be interpreted meaningfully. Even if they are not, it may be possible to achieve parsimony by reducing the dimensionality of the space needed to describe group differences. In seeking to interpret the discriminant functions, the goal is to determine which of the original p variables contribute most to each function. For this prupose, comparison of the realtive magnitudes of the combining weights as given by the elements of each eigenvector of $W^{-1}B$ is inappropriate because these are weights to be applied to the variables in raw-score scales, and are hence affected by the particular unit used for each variable.

To eleminate the spurious effects of units of measurement on the magnitudes of combining weights, standarized variables should be used.

The relative magnitudes of these standarized weights may be assessed by multiplying each raw-score weight by the standard deviation of the corresponding variable as computed from the within-groups SSCP (Sum of Squares, Cross product) matrix. This amounts to multiplying each element of a given eigenvector V_m by the square root of the corresponding diagonal element of W . Thus, for each m , define

$$v_{mi}{}^{*} - w_{ii}\, v_{mi} \qquad i = 1,2,\ldots\ldots,p \qquad (27)$$

as the standarized discriminant weights. The relative contribution of the i th variable to the m th discriminant function may then be gauged by the magnitude of $v_{mi}{}^{*}$ in comparison with the other weights $v_{mj}{}^{*}$.

Up to this point, it has been shown that the dimensionality of the discriminant space is equal to the number of nonzero eigenvalues of $W^{-1}B$, which is the smaller of the two numbers, G-1 and p . It may often happen, that the number of significant discriminant dimensions may be even smaller. That is, not all of the discriminant function may represent dimensions along which statistically significant group differences occur.

C. SIGNIFICANCE TEST IN DISCRIMINANT ANALYSIS

A basic quantity in testing the significance of the overall difference among several group centroids (mean vectors) the ratio of the determinants of the within-groups and the total SSCP matrices, known as Wilks' Λ criterion.

$$\Lambda = \frac{|W|}{|T|} \tag{28}$$

Motivation for use of this equation may be seen as follows:

$$\begin{aligned} \frac{1}{\Lambda} &= \frac{|T|}{|W|} \quad |W^{-1}T| \\ &= |W^{-1}(W + B)| \qquad (29) \\ &= (1 + \lambda)(1 + \lambda_2);\ldots.,(1 + \lambda_r) \end{aligned}$$

where $\lambda_1,\lambda_2,\ldots,\lambda_r$ are the nonzero eigenvalues of $W^{-1}B$. Consequently, Bartlet's V statistic for testing the significance of an observed value can be expressed as

$$\begin{aligned} V &= - [N - 1 - (p + G)/2]\ln\Lambda \\ &= [N - 1 - (p + G)/2] \ln [(1 + \lambda_1)(1 + \lambda_r)] \qquad (30) \\ &= [N - 1 - (p + G)/2] \sum_{m=1}^{r} \ln(1 + \lambda m) \end{aligned}$$

This statistic is distributed approximately chi-square with p(G-1) degrees of freedom.

Because of the uncorrelatedness of the successive discriminant functions, the successive terms $\ln(1 + \lambda_m)$ in the last expression above are statistically independent (assuming multivariate normality of the original p variables). As a result, the additive components of V are each approximately distributed as a chi-square variate. More specifically, the m th component,

$$V_m = [N - 1 - (p + G)/2] \ln (1 + \lambda_m) \qquad (31)$$

is approximately chi-square with $p \pm G - 2m$ degrees of freedom. It may be readily verified that the sum of the number of degree of freedom (n.d.f) of the r components, that is, $(p + G - 2) + (p + G = 4) + ...+(p + G - 2r)$, is equal to p(G - 1) regardless of whether r= G - 1 or p .

Consequently, when we cumulatively subtract V_1,V_2 , and so on from V , the remainder each time is also a chi-square variate; and these successive remainders become appropriate statistics for testing whether the residual discrimination after removing the first discriminant function, the first and second discriminant function, and so forth, is statistically significant. The successive test statistics and their n.d.f.'s may be summarized as follows:

Residual After Removing	Approximate - Statistic	n.d.f.
First discriminant Function	$V - V_1$	$p(G-1) - (p+G-2)$ $= (p-1)(G-2)$
First 2 discriminant Function	$V - V_1 - V_2$	$(p-1)(G-2)-(p+G-4)$ $=(p-2)(G-3)$
First 3 discriminant Function	$V - V_1 - V_2 - V_3$	$(p-2)(G-3)-(p+G-6)$ $=(p-3)(G-4)$
⋮	⋮	⋮
First s discriminant Function	$V-V_1-V_2-V_3 \ldots -V_s$	$(p-s)(G-(s+1))$

As soon as the residual, after removing the first s discriminant functions becomes smaller than the prescribed percentile point (that is, the 100(1 - α)th percentile) of the appropriate chi-square distribution, we may conclude that only the first s discriminant functions are significant at that α level. If the number of significant discriminant functions thus found is smaller than r (as will often be the case), we will have effected a further reduction in the dimensionality of the space required to describe the differences among the G groups from which our sample groups were drawn. The remaining r-s dimensions may be regarded as immaterial for population differentiation, since our sample differences along these dimensions can be attributed to sampling error.

IV. CLUSTER ANALYSIS

A. ORIGIN AND THEORY

Clustering is the grouping of similar objects. The principal functions of clustering are to name, to display, to summarize, to predict, and to aid in interpretation of data with many dimensions. Clustering techniques were first developed in the field of biological taxonomy. It is one of several methodologies included in the broader category called classification.

The cluster analysis problem is the last step we consider in the progression of category sorting problems. While in discriminant analysis some part of the structure is known and missing information is estimated from labeled samples, the operational objectives of clustering is to classify new observations, that is, recognize them as members of one category or another. In cluster analysis little or nothing is known about the category structure. All that is available is a collection of observations whose category membership are known. We seek to discover a category structure which fits the observations. The problem may be stated as one of finding the "natural groups", which means to sort the observations into groups such that the degree of "natural association" is high among members of the same group and low between members of different groups.

Cluster analysis techniques have been applied in many fields of study. The literature is both voluminous and diverse, the terminology differing from one field to another. "Numerical taxonomy" is frequently substituted for cluster analysis among biologists, botanists, and ecologists, while some social scientists may refer "typology". Other frequently encountered terms are pattern recognition and partitioning. While discriminant analysis has been studied by statisticians for nearly 45 years, cluster analysis has only recently come to statistical notice. Any method which partition a set of objects into subsets on the basis of measurements taken on every object qualifies as a clustering method.

Most of the well known clustering techniques fall into one of two main categories: (1) hierachical and (2) non-hierachical (partitioning). The former is one in which every cluster obtained at any stage is a merger of clusters at previous stages. The nonhierachial procedures however form new clusters by lumping and splitting old ones. We consider both categories shortly.

In a geometric sense, every observation may be viewed as a point in p-dimensional Euclidean space. This swarm of data points may contain dense regions or "clouds" of data points which are separable from other regions containing a low density of points. These denser regions constitute what are known as clusters. In one and two dimensional cases, it is easy to visualize and to detect the clusters from scatter plots, assuming that the clusters exist. In higher dimensions, clustering becomes extremely difficult without the aid of a computer.

Mathematical clustering techniques usually require a measure of similarity to be defined for every pairwise combination of the entities to be clustered. In order to solve the cluster problem, it is desirable to define the terms "similarity" and "difference" in a quantitative fashion. A researcher would assign two observations to the same group if the distance between them is sufficiently small, or to different clusters if this distance is sufficiently large.

At this point, two questions may be brought on. The first one is "how do we measure the distance between the observations?" and the second one is "how small is small enough?" and how large is large enough? These will be discussed in the following sections.

B. MEASURES OF DISTANCE

1. General

Let E_p be a symbolic representation for a measurement in p-dimensional space and let X,Y, and Z be any of these points in E_p . Then any nonnegative real-valued function D(X,Y) satisfying the following conditions qualifies as a distance function (or metric).

1. $D(X,Y) = 0$ if and only if $X = Y$
2. $D(X,Y) \geq 0$ for all X and Y in E_p

3. $D(X,Y) = D(Y,X)$

4. $D(X,Y) \leq D(X,Z) + D(Y,Z)$

Many clustering algorithm assume such distances given and set about constructing clusters of objects within which the distances are small. The choice of distance function is no less important than the choice of variables to be used in the study. A serious difficulty in choosing a distance lies in the fact that a clustering structure is more primitive than a distance function and that knowledge of clusters changes the choice of distance function. Thus a variable that distinguishes well between two established clusters should be given more weight in computing distances than a "junk" variable that distinguishes badly.

2. Euclidean Distance

The Euclidean distance between the I-th and K-th observations of a data matrix X is defined as

$$D(I,K) = \left[\sum_{1 \leq J \leq p} \{X(I,J) - X(K,J)\}^2 \right]^{1/2} \qquad (32)$$

where J is J-th variable. In one, two, or three dimensional space, this is just a "straight line" distance between the vectors corresponding to the I-th and K-th observations. When the variables are measured in different units, it is necessary to prescale the variabes to make their values comparable or, equivalently, to compute a weighted Euclidean distance.

$$D(I,K) = \left[\sum_{1 \leq J \leq p} W(J)(X(I,J) - (X(K,J))^2 \right]^{1/2} \qquad (33)$$

This form of distance is not necessary if all variables are measured on the same scale. However, even in this case, weights might be used to increase or decrease the importance of same variable. Various weighting schemes have been utilized in practice. One common weighting scheme lets $W(J)$ be the reciprocal of the variance of variable J .

A general class of squared distance functions is provided by utilizing positive definite quadratic forms. Specifically, if p represents a p-dimensional observation to be assigned to one of s groups, then to measure the squared distance between the observation β and the centroid (mean vector) of the i-th group one may consider the function

$$D_i = (\beta - \bar{x}_{i.})^T M (\beta - \bar{x}_{i.})$$

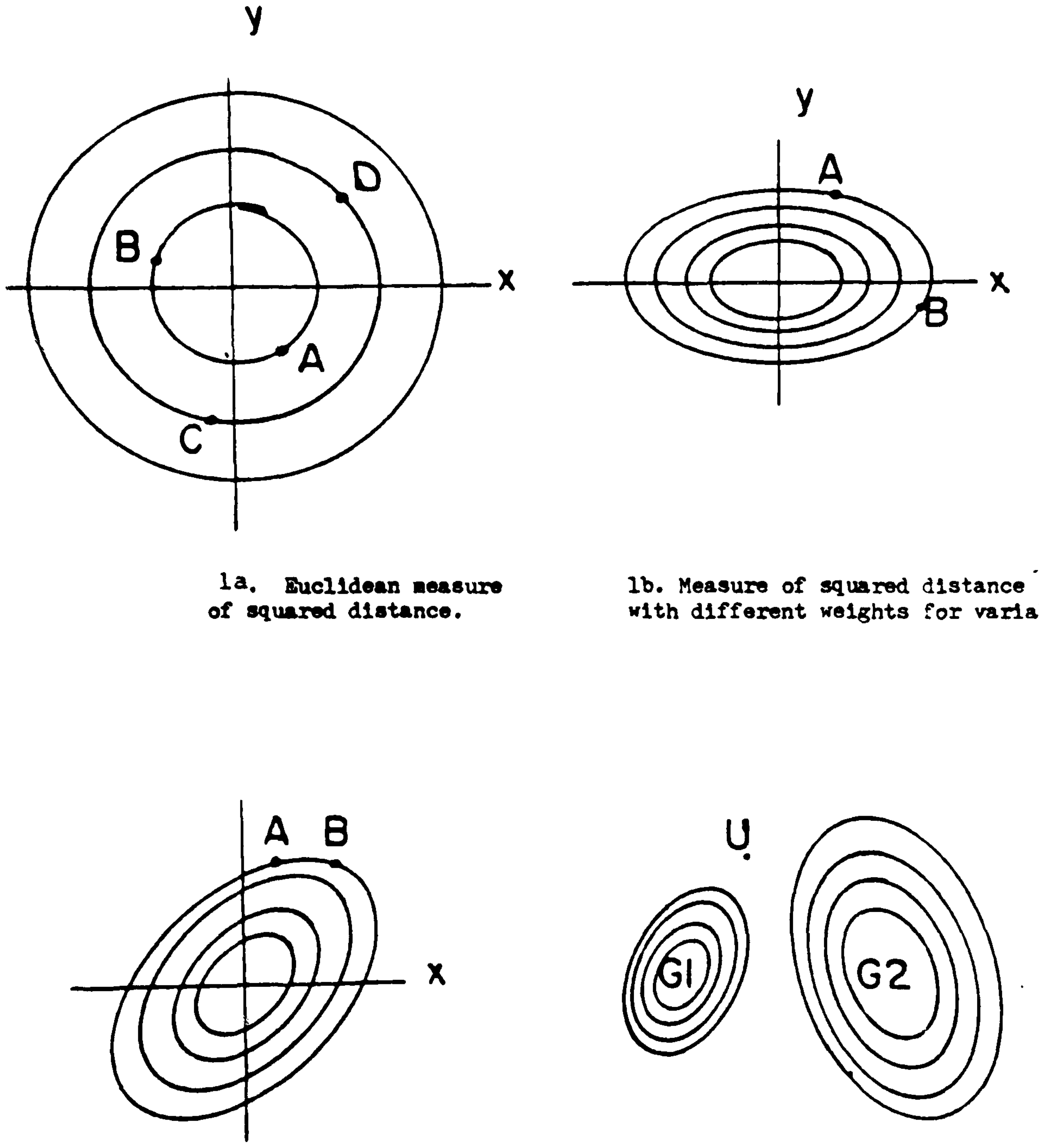
where M is a positive definite matrix to ensure that $D_i \leq 0$. Different distance functions are represented by different choices of the matrix M . When $M = I$ (the identify matrix) the resulting metric is the standard Euclidean distance. Distances with the Euclidean metric are shown in Figure 1a. The variance within the data may make the unweighted Euclidean metric inappropriate. As shown on the Figure 1b, where X has a larger variance than Y , one may wish to weight a deviation in the X direction less than an equal deviation in the Y direction. This is a weighted Euclidean distance frunction which makes point A and B equidistance from the origin. In this case, the matrix M is diagonal elements which are the reciprocals of the variances of the different variables.

Extending this idea further, it may be possible to consider the covariance among variables as well. Figure 1c shows how the axis may be rotated so that the major axis is oriented in a direction of reflecting the positive correlation between X and Y . Again, points on the same ellipse are considered equidistance from the origin. The matrix M in this case is the inverse of the covariance matrix.

Further extension of this concept will expalin some sort of generalized distance function. If C_i represents the covariance matrix of the i th cluster then the distance function

$$D_i = (\beta - \bar{x}_{i.})^T C_i^{-1} (\beta - \bar{x}_{i.})$$

uses the appropriate covariance structure when determining the distance to a particular cluster centroid. Since C_i changes to reflect the dispersion internal to each particular cluster, the use of this metric exploits differences in the dispersion characteristics of the different groups. As shown on Figure 1d, not how a new observation (denoted by u) is



1a. Euclidean measure of squared distance.

1b. Measure of squared distance with different weights for variables.

1c. Generalized squared distance measure.

1d. Classification when within-group dispersions are different.

Figure 1. Euclidean Distance

closer to the centroid of group one (G1) in terms of Euclidean distance but is more likely to be assigned to group two (G2) when using the C_i matrix.

3. Mahalanobis Distance

Another choice for the M matrix in equation (1) is p^{-1} where P represents the pooled within groups covariance matrix of all the clusters.

$$P = \frac{1}{\sum_{i=1}^{G} (n_i - 1)} W \qquad (34)$$

where

$$W = \sum_{k=1}^{G} W_k$$

This distance is the well known Mahalanobis distance. Note that P does not change from group to group. To ensure the non-singularity of P it must be true that $p \leq (N - G)$, where N represents the total number of observations over all groups. Rewriting the distance,

$$D_i = (\beta - \bar{x}_{i.})^T W^{-1} (\beta - \bar{x}_{i.}) \qquad (35)$$

defines a distance between mean vectors β and $\bar{x}_{i.}$ and common covariance matrix W. The Mahalanobis distance function adjusts for both scale of measurement of the variables and covariation among the variables. Use of this metric is equivalent to computing distances on variables transformed to their principal components. This metric is invariant under any nonsingular transformation of original variables. For consider the transformation

$$Y = BX \qquad (36)$$

and let $D(Y_i, Y_j)$ represent Mahalanobis distance between Y_i and Y_j .

$$\begin{aligned} D(Y_i, Y_j) &= (Y_i - Y_j)^T P_Y^{-1} (Y_i - Y_j) \\ &= (BX_i - BX_j)^T P_Y^{-1} (BX_i - BX_j) \\ &= (X_i - X_j)^T \; B^T \; P_Y^{-1} \; B(X_i - X_j) \\ &= (X_i - X_j)^T \; B^T \; (BP_X B^T)^{-1} B(X_i \quad X_j) \\ &= (X_i - X_j)^T \; P_X^{-1} (X_i - X_j) \\ &= D(X_i, X_j) \end{aligned}$$

Some other common metrics are listed below:

1. L_1 norm (City Block)

$$D(X_i, X_j) = \sum_{k=1}^{P} |X_{ki} - X_{kj}|$$

2. L_p norm (Minkowsky Metrics)

$$D(X_i, X_j) = \Big(\sum_{k=1}^{P} |X_{ki} - X_{kj}|^{P} \Big)^{1/p}$$

3. Uniform norm

$$D(X_i,X_j) = \underset{k = 1,\ 2,\ \ldots,p}{\text{Superemum}} \{|X_{ki} - X_{kj}|\}$$

C. HIERARCHICAL CLUSTERING

1. General

The previously discussed distance measures may be used to construct a similarity matrix describing the length of all pairwise relationships among the entities (variables or data units) in the data set. The methods of hierachical cluster analysis operate on this similarity matrix to construct a tree depicting specified relationships among the entities. As shown on Figure 2, the branches on the left each represent one entity while the root represents the entire collection of entities. Moving down the tree from the branches toward the root depicts increasing aggregation of the entities into clusters. Hierarchical clustering methods which build a tree from branches to root often are called agglomerative methods.

Once a tree is constructed for N entities, the analyst may choose from as many as N sets of clusters. These clusters are nested. From the agglomerative view, when two entities are merged they are joined togehter permanently and considered as one entity for later merges; from the divisive view, when a group of entities is split into two parts, the parts are separated permenently and may be treated independently for the remainder of the analysis.

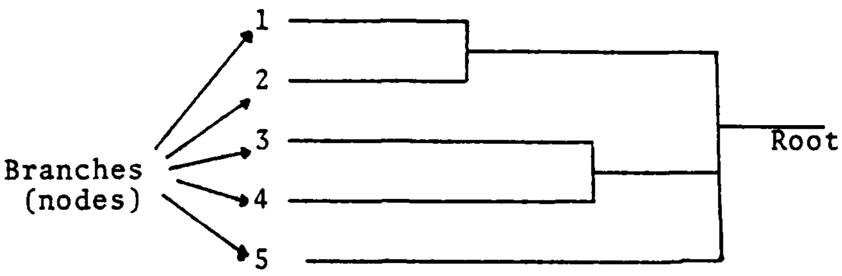


Figure 2. Tree for Hierarchical Clustering

Herein lie both the strength and weakness of hierarchical methods: by taking early decisions as permanent, the number of posibilities that need be examined is reduced greatly as compared with complete enumeration; but this same convention precludes discovering early mistakes or capitalizing on later opportunities.

There are three major hierarchical clustering concepts:

1. Linkage Methods
2. Centroid Methods
3. Error sum of squares or variance methods.

All of these methods are suitable for clustering data units. However, only the linkage methods are considered in this research.

2. The General Agglomerative Procedure

Let s_{ij} be the similarity between entities i and j as defined by one of the distance measures previously discussed. Assuming that the similarity is symmetric, the complete schedule of similarities for all $\binom{N}{2} = \frac{1}{2}N(N - 1)$

possible pairwise combinations of entities may be arrayed in a lower triangular similarity matrix as in Figure 3. The s_{ij} entries are nonnegative. This limitation is of consequence only for correlation and the cosine of the angle between vectors; the distinction between positive and negative association cannot be utilized in these clustering methods.

$$S = \begin{array}{|lllll}
s_{21} & & & & \\
s_{31} & s_{32} & & & \\
s_{41} & s_{42} & s_{43} & & \\
\cdot & \cdot & \cdot & & \\
\cdot & \cdot & \cdot & \cdot & \\
\cdot & \cdot & \cdot & \cdot & \\
s_{n1} & s_{n2} & s_{n3} & \cdots & s_{n(n-1)} \\
\hline
\end{array}$$

Figure 3. Lower Triangular Similarity Matrix

A simple remedy is to use the absolute value or the square of the measure if it can assume negative values. Once the matrix is defined, the process of clustering entities is almost trivially simple. The general procedure for agglomerative clustering on a data matrix is as follows:

(1) Begin with n clusters each consisting of exactly one entity. Let the clusters are labled with the numbers 1 through N.

(2) Search the similarity matrix for the most similar pair of clusters. Let the chosen clusters be labeled p and q and let their associated similarity be s_{pq}, $p > q$.

(3) Reduce the number of clusters by 1 thorugh merger of clusters p and q . Label the product of the merger q and update the similarity matrix entities in order to reflect the revised similarities between cluster q and all other existing clusters. Delete the row and column of S pertaining to cluster p .

(4) Perform steps 2 and 3 a total of N-1 times (at which point all entities will be one cluster). At each step record the identity of the clusters which are merged and the value of similarity between them in order to have a complete record of the results.

Different agglomerative methods are implemented by varying the procedures used for defining the most similar pair at step 2 and for updating the revised similarity matrix at step 3. The similarity matrix is a given array of numbers. The numerical execution of the clustering procedures is completely independent of how the similarity values were generated or whether the entities to be clustered are variables or data units. However, it is necessary to make a direct distinction between distance-like measures (the smallest values correspond to the most similar pairs) and correlation-like measures (the largest values correspond to the most similar pairs); the essential difference is whether the search for the most similar pair involves seeking the minimum or maximum entry in the similarity matrix.

3. Single Linkage

The method of single-linkage cluster analysis is the simplest of all hierarchical techniques. At each stage, after clusters p and q have been merged, the similarity between the cluster (labeled t) and some other r is determined as follows:

1. If s_{ij} is the distance-line measure

$$s_{tr} = \min (s_{pr}, s_{qr}) \tag{37}$$

The quantity s_{tr} is the distance between the two closest members of clusters t and r. If clusters t and r were to be merged, then for any entity in the resulting cluster the distance to its nearest neighbor would be at most s_{tr}.

2. If s_{ij} is a correlation-like measure

$$s_{tr} = \max (s_{pr}, s_{qr}) \tag{38}$$

The quantity s_{tr} is the similarity between the two most similar entities in clusters t and r. If clusters t

and r were to be merged, then for any entity in the resulting cluster there would be at least one other entity in the same cluster such that the pair would have a similarity at least as large as s_{tr} .

The method is known as single linkage because clusters are joined at each stage by the single shortest or strongest link between them. Since the updating process involves choosing only the minimum or maximum single-linkage clustering is invariant to any transformation which leaves the ordering of the similarities unchanged; that is, any monotonic transformation.

4. Complete Linkage

The complete-linkage method is related to the single-linkage method and is no more difficult to execute. At each stage, after clusters p and q have been merged, the similarity between the new cluster (labeled t) and some other cluster r is determined as follows:

1. If s_{ij} is distance-like measure

$$s_{tr} = \max (s_{pr}, s_{qr}) \qquad (39)$$

The quantity s_{tr} is the distance between the most distant members of clusters t and r . If clusters t and r were merged, then every entity in the resulting cluster would be no farther than s_{tr} from every other entity in the cluster. The value of s_{tr} is the diameter of the samllest sphere which can enclose the cluster resulting from the merger of clusters t and r.

2. If s_{ij} is a correlation-like measure

$$s_{tr} = \min (s_{pr}, s_{qr}) \tag{40}$$

The quantify s_{tr} is the similarity between the two most dissimilar entities in clusters t and r. If clusters t and r were to be merged, then every entity in the resulting cluster would have a similarity of at least s_{tr} with every other entity in the cluster.

The method is called complete linkage because all entities in a cluster are linked to each other at some maximum distance or minimum similarity. Such a cluster is called a "maximally connected subgraph" in graph theory. In contrast to the single-linkage method, interpretation of the clusters can be made *only* in terms of the relationships within individual clusters; there is no particularly useful interpretation involving the differences between clusters. Like the single-linkage method, complete-linkage cluster analysis is invariant to monotonic transformations of the similarity measure. Johnson (1967) discusses this property in both single and complete linkage methods.

D. NONHIERARCHICAL CLUSTERING

Nonhierarchical clustering methods are designed to cluster data units into a single classification of g clusters, where g either is specified a priori or is determined as a part of the clustering method. The central idea in most of these methods is to choose some initial partition of the data units and then alter cluster memberships so as to obtain a better partition. The various algorighms which have been proposed differ as to what constitutes a "better partition" and what methods may be used for achieving improvements.

The broad concept for these methods is very similar to that underlying the steepest descent algorithms used for unconstrained optimization in nonlinear programming. Such algorithms begin with an initial point and then converge to a local optimum, moving one step at a time, the value of the objective function improving at each step.

The methods of nonhierarchical clustering typically may be used with much larger problems than the hierarchical methods because it is not necessary to calculate and store the similarity matrix; it is not even necessary to store the data set. In general, the data units are processed serially and can be read from tape or disk as needed. This characteristic makes it possible, at least in principle, to cluster arbitrary large collections of data units.

In this research, we consider only the partitioning method known as "K-MEANS" which was developed by MacQueen (15). He used the term "K-MEANS" to denote the process of assigning each data unit to that cluster (of k clusters) with the nearest centroid (mean vector). The cluster centroid changes with each transfer of an observation.

The decomposition of the total scatter matrix into within and between groups matrices suggests possible optimality criteria to be used in a clustering algorithm. One would like the within-groups scatter to be small relative to the between-groups scatter. Various trial clusterings could be formed using the W and B matrices as a basis for the optimality criteria which determine the best clustering. A possible choice for a criterion is to minimize trace W over all partitions into g groups. Since T is constant over all partitions, minimizing trace W is equvalent to maximizing traces B since

$$\text{trace } T = \text{trace } W + \text{trace } B \qquad (41)$$

Although trace W is invariant under an orthogonal transformation, it is not invariant under other non-singular linear transformations.

McRae (16) points out that trace W equals the total within group sum of squares, hence the "minimum variance partition" cluster solution is found by minimizing trace W .

Considerable study has been developed to alternative criteria such as those based on multivariate statistical analysis techniques, especially the methods of linear discriminant analysis and multivariate analysis of variance. Assuming the p variables are not linearly dependent, then as long as $p = N - g$, W is positive definite symmetric and so is W^{-1} . Attempts to make B and W as different as possible lead one to solving the determinantal equation:

$$|B - \lambda W| = 0 \qquad (42)$$

The solutions λ_i are the eigenvalues of the matrix $W^{-1}B$ as in discriminant analysis. There are t non-zero eigenvalues, where t is the minimum of p and $g-1$. This is a consequence of the fact that, if g is less than p, the g group means are considered in a $(g-1)$-dimensional hyperplane. When $g = 2$ the analysis is equivalent to two-group discriminant analysis. Linear discriminant analysis would take the vectors originally described in p-dimensional coordinate system and transform the basis to a t-dimensional system. Maximizing the largest of these eigenvalues is a criterion suggested by S.N. Roy and maximizing the trace of $W^{-1}B$, however is a criterion suggested by Hotelling. In both cases, large values for these statistics are sought in clustering algorithms since large values indicate large differences among (between) groups. Minimizing the ratio of determinants $|W| \div |T|$ is a criterion widely known as Wilks' lambda discussed in the discriminant analysis. Since T is the same for all partitions, this criterion is equivalent to minimizing determinant W. Both trace $W^{-1}B$ and $|T| \div |W|$ may be expressed in terms of the eignevalues of $W^{-1}B$.

$$\left|\frac{T}{W}\right| = \prod_{i=1}^{t} (1 + \lambda_i) \qquad (43)$$

$$\text{trace} \quad W^{-1}B = \sum_{i=1}^{t} \lambda_i \qquad (44)$$

where $t = \min(p, g-1)$. Therefore minimizing det W is equivalent to maximizing $\pi(1 + \lambda_i)$.

Friedman and Rubin (6) describe the advantages of the various criteria. Those based on multivariate statistical considerations (all but trace W) are invariant under changes in scale for varibles (non-singular linear transformation). In fact, they are the only invariants for W and B under such transformations. In addition, the multivariate criteria may take into account covariation among the variables.

V. ANALYSIS OF MULTIVARIATE UTILITY DATA

To illustrate hierarchical clustering we applied the technique described in the previous chapter to partition a set of twenty six attributes of a close-air support weapon system into a smaller collection of "superattributes". As part of an effort to evaluate the military utility of a proposed alternative U.S. Marine Corps air support rada system, AN-TPQ/27. Barr and Richards (4) extracted 26 attributes of the TPQ-27 and a baseline system, the AN-TPQ/10, and then had members of the Operational Test and Evaluation Team assess the utility of the TPQ/27 relative to that of the TPQ/10. In order that the additive model used to combine unidimensional relative utilities into a system relative utility be justifiable, it is necessary that the utilities satisfy certain independence properties described in Keeney and Raiffa (12).

Because those independence properties are very difficult for decision makers to verify for complex alternatives like the weapon systems under study, Professors Barr and Richards attempted instead to work with the attributes to try to generate a new collection which would likely satisfy, at least approximately, the conditions required to justify the additive model.

The original collection of 26 attributes is as follows:

1. Portability
2. Durability
3. Time to Set Up
4. Time to Take Down
5. Ease of Assigning Aircraft to Targets
6. Number of Aircraft Controlled
7. Number of Targets
8. Communications
9. Mission Flexibility
10. ASRT Survivability
11. Time to Locate and Acquire Aircraft
12. Accuracy of Tracking
13. Accuracy of Delivery
14. Range
15. Aircraft Vulnerability
16. Aircraft Attack Throughout
17. Base of Adjustment and Evaluation of Results
18. Accuracy of Feedback
19. Ease of Operation
20. Man-Machine Compatibility
21. Training Requirements
22. Reliability
23. Maintainability
24. Supportability
25. Availability
26. Documentation

where a_i represents an attribute i and

$$I(x_{ij}, x_{kj}) = \begin{cases} 1 & \text{if } x_{ij} = x_{kj} \\ 0 & \text{if } x_{ij} \neq x_{kj} \end{cases}$$

It is easy to verify that D is a metric as defined in Chapter IV. Since we will actually work with a similarity measure in the hierarchical cluster procedure, we define the similarity between two attributes a_i and a_k as

$$S(a_i , a_k) = \sum_{j=1}^{12} I(x_{ij} , x_{kj}) \tag{46}$$

One can see from this definition that the similarity between two attributes a_i and a_k is simply the number of team members who placed attributes a_i and a_k in the same partition. For example,

$$S(a_1 , a_2) = 0 + 1 + 0 + 1 + 0 + 0 + 0 + 1 + 0 + 0 + 0 + 1 + 0 = 4$$

Either S or D can be used in the computer program shown in Appendix A for hierarchical clustering. One need only indicate whether he wants a correlation-like (larger values imply more similar) measure or a distance-like measure (smaller values imply more similar). We selected to use the former method. The similarity matrix extracted from

Table II. Data Matrix

	1	2	3	4	5	6	7	8	9	10	11	12
1	6	2	1	3	2	4	3	1	3	1	1	1
2	1	2	2	3	1	3	2	1	1	5	1	2
3	6	1	1	5	2	4	3	1	6	1	2	1
4	6	1	1	5	2	5	3	1	6	1	2	1
5	2	5	3	1	3	1	1	2	2	2	3	3
6	2	7	4	1	4	1	1	2	4	2	3	3
7	2	7	4	1	4	1	1	2	4	2	3	3
8	3	5	4	7	5	6	1	3	1	3	3	6
9	2	5	4	1	3	1	1	2	4	2	3	3
10	9	6	5	6	5	7	8	3	7	3	2	4
11	2	8	6	4	3	2	1	4	4	2	3	3
12	3	8	7	4	4	2	6	5	4	6	4	3
13	3	8	7	4	4	2	6	5	4	6	4	3
14	3	8	4	4	4	2	6	5	4	2	4	3
15	7	6	5	6	5	7	7	3	7	3	7	4
16	8	8	6	1	4	1	7	4	4	2	7	3
17	8	8	8	4	3	2	5	6	5	4	5	3
18	4	8	8	4	6	2	5	6	5	6	5	3
19	4	5	3	1	3	1	1	2	2	4	3	3
20	4	5	3	3	3	3	3	1	2	4	8	3
21	5	4	9	2	3	8	4	7	2	4	6	5
22	1	3	2	3	1	3	2	7	1	5	6	2
23	1	3	9	3	1	3	2	7	1	5	6	2
24	1	3	4	3	1	3	2	7	3	5	6	2
25	1	3	2	3	1	3	2	7	1	5	6	2
26	5	4	9	2	6	8	4	7	2	4	6	5

from the data is shown in Table V-3. We present only lower triangular elements since $S(a_i , a_i) = 12$ for all i and the matrix is symmetric; i.e., $S(a_i , a_k) = S(a_k , a_i)$. Zero values are not written.

The results from the hierarchical clustering are shown in Figure 4. The numbers printed along the left hand margin refer to the attribute numbers. As you proceed to the right through the tree you will observe numbers greater than 26. These correspond to the clusterings that takes place from one step to the next. For example, the number 27 shown at the juncture of 25 and 22 means that the first attribute clustered together should be 25 and 22 (this is the most similar pair). This combination is then considered as a new attribute which is later combined with the attribute 30 (itself a combination of 23 and 24) to form the attribute 31. This is later combined with attribute 2 to form attribute 40, etc.

As discussed in Chapter IV a decision has to be made as to how many clusters (superattributes) are desired. All hierarchical methods will continue clustering until there is a single cluster. In order to decide on the number of clusters (and their composition) one need only image drawing a vertical line through the tree at various places. Each intersection of the tree with the vertical line results in a cluster. For example, teh vertical line at the point A results in the 6 clusters shown in Table V-4.

It is clear from observing the above collection that some of the attributes are highly correlated and nonredundant. If one tries to assign an importance weights to each attributes separately, there is a distinct likelihood that some of the overlapping strongly into related attributes might effectively be double or triple weighted or more producing biased result. It is an effort to prevent this from happening, Barr and Richards aksed the utility assessment team to partition the 26 attributes into a smaller collection in such a way that attributes within a group are similiar and attributes in different groups are unrelated the sense that utility assessments for attributes in one group do not depend on the amounts of attributes in any other group.

The total number of groups was not prespecified. Instead, each team member was allowed to partition the 26 attributes into any number of groups. The resulting multivariate data array is shown in Table V-2. An element x_{ij} is the number of the group into each team member j put attribute i .

Let us define a distance measure for this data array as follows:

$$D(a_i, a_k) = \sum_{j=1}^{12} (1 - I(x_{ij} , x_{kj}) \qquad (45)$$

Table III. Similarity Matrix for Superattribute Determination .

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1																										
2	4																									
3	8	1																								
4	7	1	11																							
5																										
6					8																					
7					8	12																				
8		1			3	3	3																			
9					10	10	10	3																		
10			1	1				3																		
11					6	6	6	2	7																	
12					1	3	3	3	1	2		5														
13					1	3	3	3	1	2		5	12													
14					2	5	5	2	4			5	9	9												
15								3		9																
16					3	5	5		4	1	6	3	3	4	3											
17					2	1	1	1	2		5	4	4	3		2										
18					1	1	1	1	1		5	4	4	3		1	9									
19					9	5	5	2	7		3					2	2	1								
20	2	3	1	1	5	1	1	1	3		2						2	1	8							
21					1				1		1						2		3	3						
22	2	8						1												2	2					
23	2	7						1												2	3	11				
24	3	6						1												2	3	10	11			
25	2	1																		2	2	12	11	10		
26					1												1	1	2	2	9	3	4	4	3	

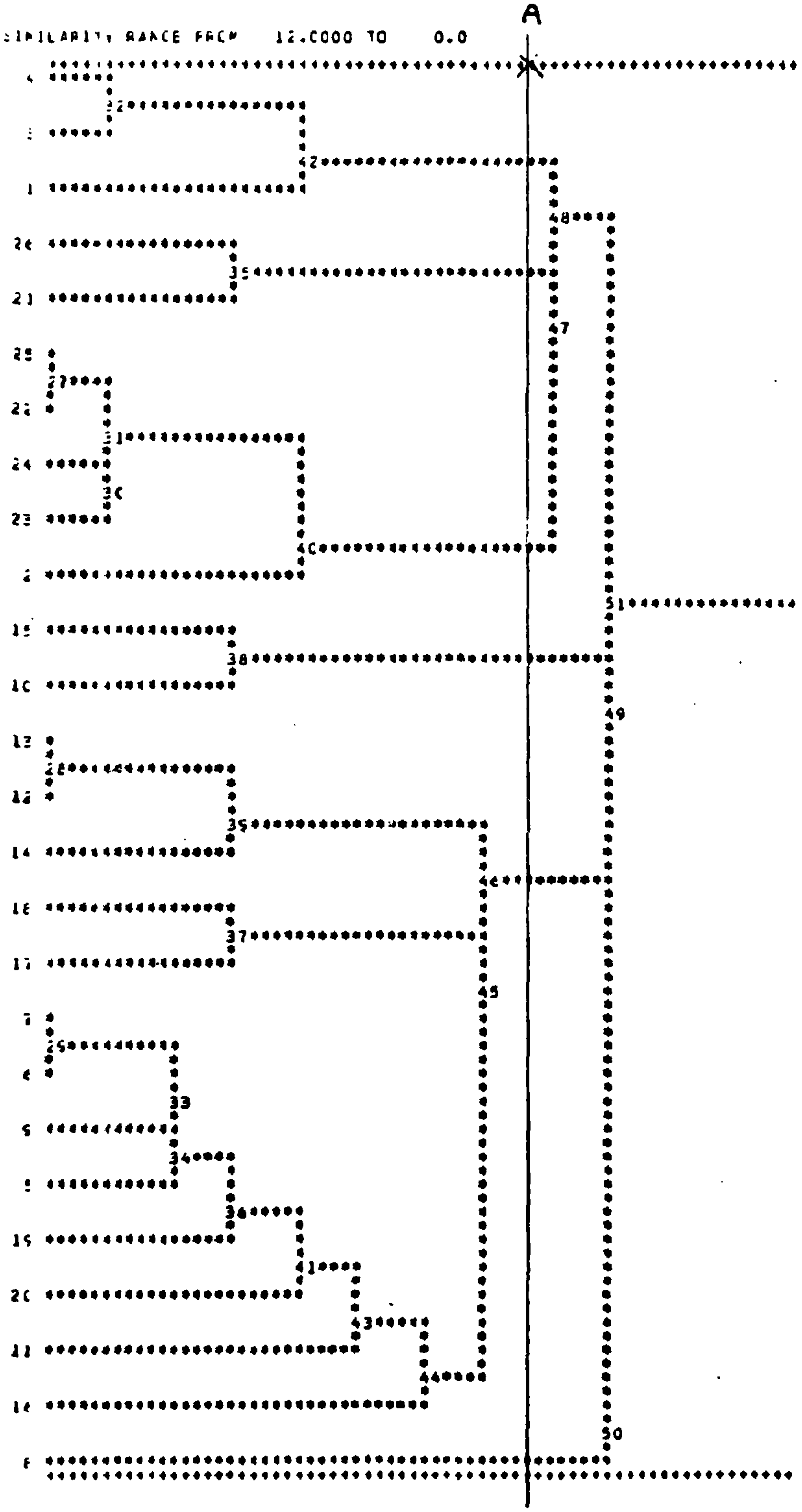


Figure 4. Tree for 26 Attributes

The superattributes used in the utility study are those sohwn in Table IV. A careful examination of the attributes which comapre the clusters shows that the results so obtained are intuitively agreeable. The names supplied to the superattribures are somewhat natural descriptions of the clusters obtained.

The listing of the computer program and sample output are given in Appendix A.

Table IV. Superattributes

Superattributes	Component Attributes
Facility of movement	1. Portability
	3. Time to set up
	4. Time to take down
Facility of Use	5. Ease of assigning aircraft to targets
(precision)	6. Number of aircraft controlled
	7. Number of targets
	9. Mission flexibility
	11. Time to locate and acquire aircraft
	16. Aircraft attack throughput
	17. Ease of adjustment
	18. Accuracy of feedback
	19. Ease of operation
	20. Man-machine compatibility
	12. Accuracy of tracking
	13. Accuracy of delivery
	14. Range
Survivability	10. ASRT Survivability
	15. Aircraft vulnerability
Learning	21. Training requirements
	26. Documentation
Readiness	2. Durability
	22. Reliability
	23. Maintainability
	24. Supportability
	25. Availability
Communications	8. Communications

VI. ANALYSIS OF ARMY TANK DATA

A. DATA STRUCTURE

In order to illustrate the nonhierarchical clustering methodology, principal components analysis, and discriminant analysis data on Army tanks from eight different countries were taken from Jane's Book of Weapon Systems (1979-80). A total of twenty-four tanks were included in the data array with observation on each of 10 variables. The 10 variables are listed below:

1. Weight (ton)
2. Length (meter)
3. Width (meter)
4. Height (meter)
5. Road Speed (kilometer per hour)
6. Trench Crossing (meter)
7. Ground Pressure (Kg/cm^2)
8. Maximum Armament (rounds)
9. Ground Clearance (meter)
10. Power to Engine Ratio (BHP/ton)

The twenty-four tanks and the associated countries are shown below:

Identification Number	Type/Name	Country
11	T-62	
12	T-54	U.S.S.R.

Identification Number	Type/Name	Country
13	T-10	
14	ASU-85	
15	MK-5/Chieftain	
16	MK-3/Vickers	
17	MK-13/Centurion	U.K.
18	CVR(T)/Scorpion	
19	XM-1	U.S.A.
20	M60A2	
21	M60	
22	M48	
23	M47	
24	PZ61	SWITZERLAND
25	PZ68	
26	STRV-103	SWEDEN
27	Ikv-91	
28	TYPE61	JAPAN
29	TYPE74	
30	Leopard 2	W. GERMANY
31	Leopard	
32	TAM	
33	AMX 30	FRENCH
34	AMX 13	

We conjecture that a cluster analysis of the tank data will result in clusters corresponding to nationality since the nations may have different emphasis on the variables in the design of their tanks.

B. NONHIERARCHICAL CLUSTER ANALYSIS OF TANK DATA

1. The MIKCA Algorithm

The specific algorithm chosen for the nonhierarchical cluster analysis for the tank data is the MIKCA (Multivariate Iterative K-MEANS Clustering Algorithm) program written by Douglas J. McRae as a part of his doctoral dissertation at the University of North Carolina, Chapel Hill.

Reference to the flow chart in Figure 5 will aid the reader in following discussion of the algorithm. Inputs to program are the data matrix, an estimate for g (the number of clusters), and choice of criterion and distance functions.

In the first step, preliminary claculations are made, such as the variable means and standard deviations, as well as the cross product matrix T. The next step forms the initial cluster centers. Then each of the other observations is assigned to the nearest cluster. Euclidean distance is used for this initial phase, and the cluster centroids are recomputed after each observation is assigned to a group. The observations are considered in the same order as they were input. After all of them have been assigned to clusters, the criterion value is computed. This initial cluster-finding technique is referred to as a

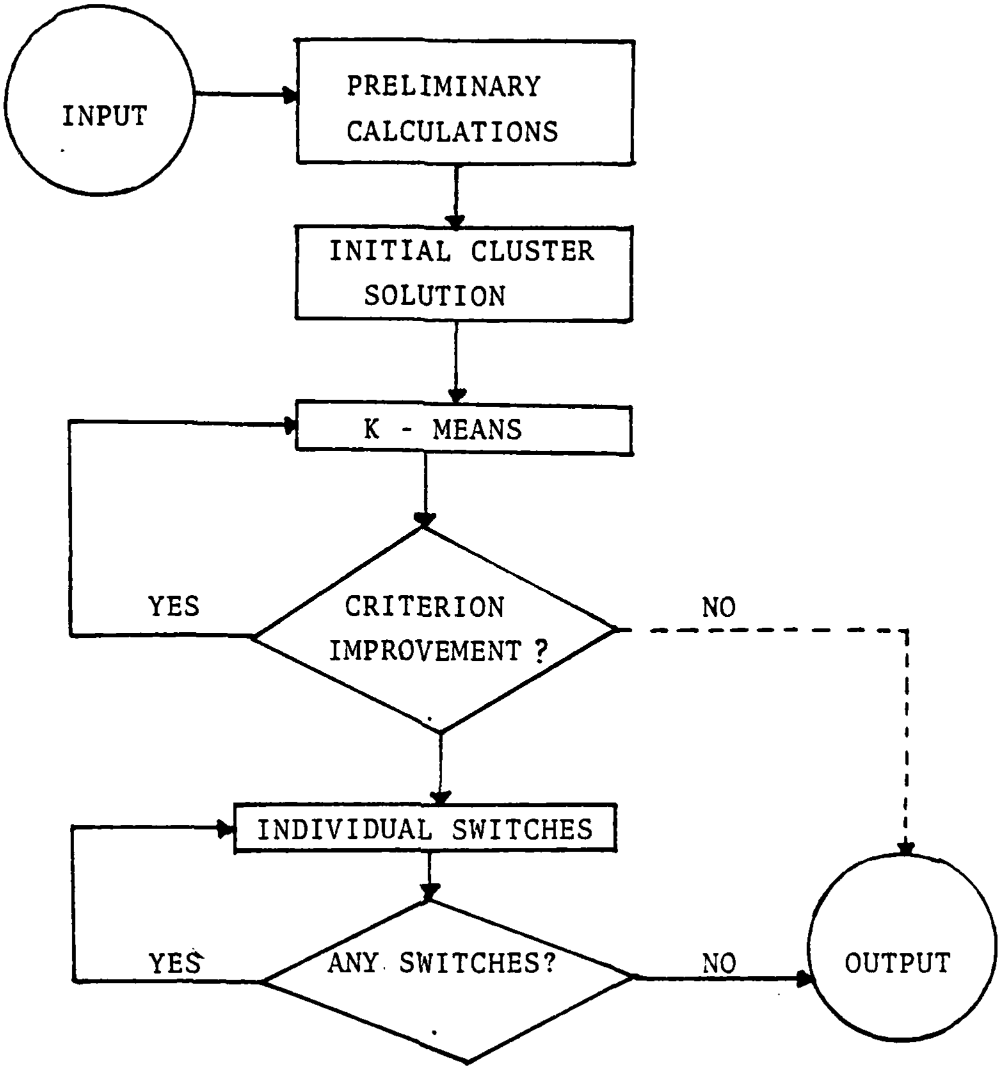


Figure 5. MIKCA Flow Chart

one-pass K-MEANS procedure. It is performed three times, and the solution which yields the best criterion value is chosen as the initial cluster solution.

After the initial solution has been found, the program advances to the iterative K-MEANS phase where the observations are again considered in the order in which they were input to the program. It is this phase where the user's choice of distance function is used. The distance from each observation to each cluster centroid is again computed, this time with the user's distance function, the assignment to the closest centroid being made and the centroid updated to reflect its new membership. After considering all n observations in this manner, the new criterion value is checked for possible improvement during the K-MEANS iteration. As long as the criterion value improves, the K-MEANS procedure is repeated; if the criterion fails to improve then the MIKCA algorithm goes to the next step, the individual switches section. Note the importance of the order of consideration of the observations. The order is important because the cluster means are recomputed after each observation is reassigned.

In the individual switches phase, consideration is given to moving each observation to every other cluster, the move being made if and only if an improvement in the value of the criterion results. An elaborate labelling procedure provides a unique order in which to consider each observation.

This procedure continues until a complete pass through the data is made with no changes in cluster membership.

The MIKCA alogorithm provides the following options for distance and criterion functions.

Criterion

1. Minimum trace W
2. Minimum determinant W
3. Maximum largest order of $|B - \lambda W| = 0$
4. Maximum sum of roots of $|B - \lambda W| = 0$

Distance

1. Euclidean
2. Weighted Euclidean
3. Mahalanobis

A complete computer program is listed in Appendix B.

2. Cluster Results for Tank Data

For clustering of the tank data we selected the minimum trace W criterion and the weighted Euclidean distance function. The algorithm automatically provides weights for the weighted Euclidean distance function. The results of the clustering with four clusters are shown in Table V.

The conjecture of clustering by nationalities is supported by the results. The three Soviet tanks make up one cluster and the two British and four of the United States tanks were found to be similar. A third cluster consists of four tanks which are very lightweight. The

Table V.

```
THE FINAL CLUSTER SOLUTION IS

CLUSTER  1
     SIZE =  11
     CENTER =      40.60        7.00         3.26         2.55        59.18
                        2.64         0.88         53.84         0.46        20.70
     THE OBSERVATIONS ARE
             16        19        24        25        26
              28        29        30        31        32
             33

CLUSTER  2
     SIZE =   6
     CENTER =      51.87        7.15         3.55         3.06        43.05
                        2.80         0.88         59.50         0.43        15.95
     THE OBSERVATIONS ARE
             15        17        20        21        22
              23

CLUSTER  3
     SIZE =   4
     CENTER =      13.08        5.40         2.77         2.21        64.37
                        2.14         0.51         45.75         0.38        20.07
     THE OBSERVATIONS ARE
             14        18        27        34

CLUSTER  4
     SIZE =   3
     CENTER =      41.00        6.85         3.50         2.35        44.33
                        2.85         0.77         37.67         0.44        15.07
     THE OBSERVATIONS ARE
             11        12        13
```

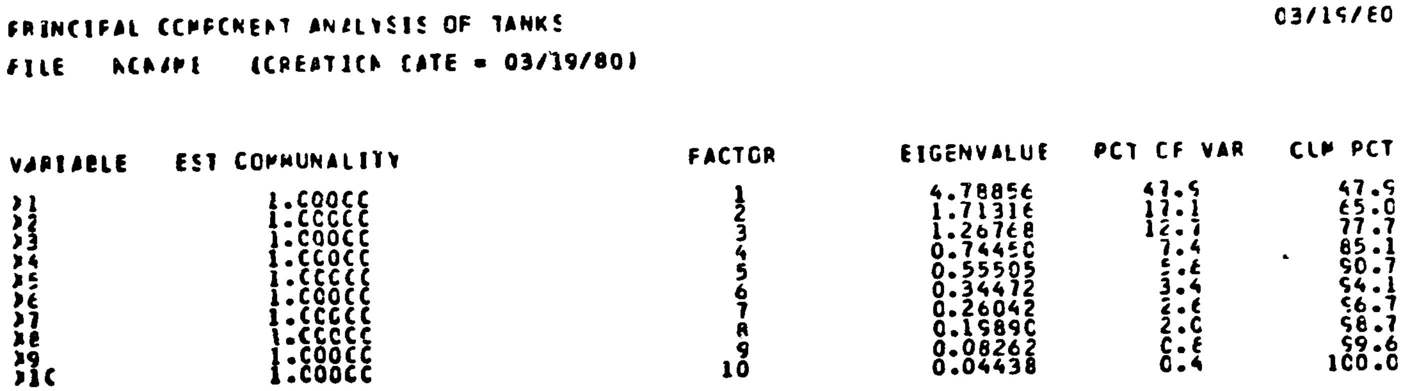
final cluster consists of the rest of the tanks, including tanks of United States allies from West Germany, France, Sweden, Switzerland and Japan.

A natural question to ask after observing the results of a cluster analysis is what variables most strongly influence the clustering that was observed. A clue is provided by the composition of the cluster containing all of the lightweight tanks. This suggests that weight is an important distinguishing feature. This is examined in the principal components analysis and the discriminant analysis in the next two section.

C. PRINCIPAL COMPONENTS ANALYSIS

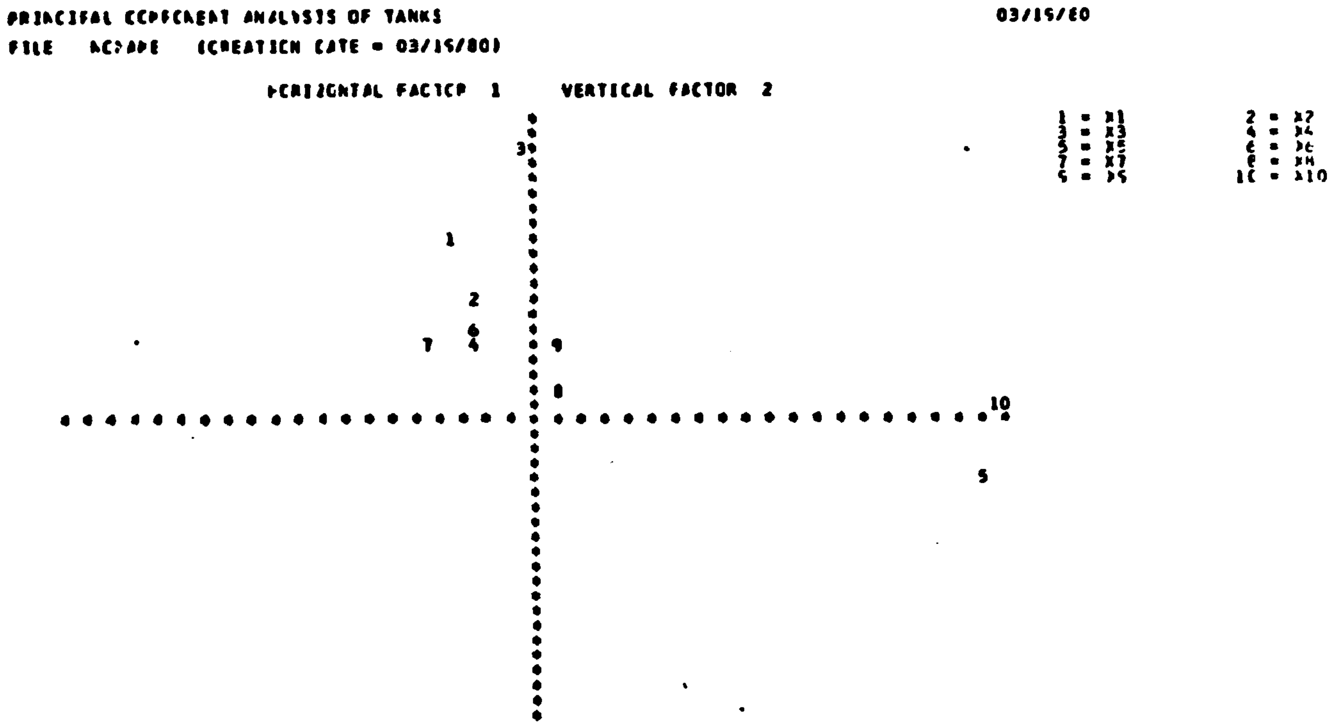
The Statistical Package for Social Sciences (SPSS) (14) subprogram FACTOR was used for the principal components analysis. It is designed both for the factor analysis and the principal components analysis. The outputs are designed to be self-explanatory. In this example, the first 5 components accoung for 90% of the variance and the remaining components account for only 10% of the variance (Figure 6).

The subprogram FACTOR provides a graphical presentation (Figure 7) for the factors that have been determined by the orthogonal rotations (in this example, variance maximization rotation). In reading the graphs, one should be attentive to following three features: (1) the relative distance of a variable from the axis, (2) the direction of a variable in relation to the axis, and finally (3) clustering of variables and their relative position to each other.



```
PRINCIPAL COMPONENT ANALYSIS OF TANKS                                           03/19/80
FILE   NONAME   (CREATION DATE = 03/19/80)

VARIABLE   EST COMMUNALITY                     FACTOR        EIGENVALUE   PCT OF VAR   CUM PCT

X1             1.00000                            1           4.78856        47.9        47.9
X2             1.00000                            2           1.71316        17.1        65.0
X3             1.00000                            3           1.26768        12.7        77.7
X4             1.00000                            4           0.74450         7.4        85.1
X5             1.00000                            5           0.55505         5.6        90.7
X6             1.00000                            6           0.34472         3.4        94.1
X7             1.00000                            7           0.26042         2.6        96.7
X8             1.00000                            8           0.19890         2.0        98.7
X9             1.00000                            9           0.08262         0.8        99.6
X10            1.00000                           10           0.04438         0.4       100.0
```

Figure 6. Summary Table of Principal Components Analysis on Tank



```
PRINCIPAL COMPONENT ANALYSIS OF TANKS                                    03/15/80
FILE   NONAME   (CREATION DATE = 03/15/80)
                  HORIZONTAL FACTOR  1     VERTICAL FACTOR  2
                                                                 1 = X1        2 = X2
                                                                 3 = X3        4 = X4
                                                                 5 = X5        6 = X6
                                                                 7 = X7        8 = X8
                                                                 9 = X9       10 = X10
```

Figure 7. Graphical Presentation

```
PRINCIPAL COMPONENT ANALYSIS OF TANKS                                                  03/15/80
FILE   NONAME   (CREATION DATE = 03/15/80)

FACTOR SCORE COEFFICIENTS

            FACTOR  1  FACTOR  2  FACTOR  3  FACTOR  4  FACTOR  5  FACTOR  6  FACTOR  7  FACTOR  8  FACTOR  9  FACTOR 10

D1            0.02644    0.17582   -0.06068   -0.02000   -0.05601   -0.02585   -0.00275   -0.08455   -0.47620    3.51646
D2            0.02234   -0.19940   -0.15643   -0.06221   -0.12437   -0.00934   -0.16675    1.88856    0.06904   -0.96578
D3           -0.02165    1.37153   -0.24574   -0.27654   -0.10554    0.01455   -0.14086   -0.45910   -0.37087   -1.41314
D4            0.07825   -0.34400    0.00699    1.26462   -0.00536   -0.21936   -0.03955   -0.08532    0.55301   -1.13218
D5            0.51004    0.00956    0.55455    0.04921   -0.07047   -0.03799    0.14451   -0.00145   -2.51081    0.26147
D6            0.08605   -0.30115    1.33188    0.00662   -0.14154    0.00258    0.06701   -0.41005   -0.27006   -0.19639
D7            0.12847   -0.14689    0.04732   -0.03347   -0.30204   -0.25737    1.55608   -0.25602    0.23897   -1.08035
D8           -0.06759    0.01688    0.00215   -0.26656    0.18410    1.17855   -0.42128   -0.01152   -0.45504    0.36444
D9           -0.08521   -0.14417   -0.15071   -0.00681    1.30032    0.15771   -0.42174   -0.27614   -0.13517   -0.04055
D10           0.06446   -0.06029    0.08156    0.09762   -0.09737   -0.07684    0.18227    0.02541    2.41735   -0.13659
```

Figure 8. Factor Score Coefficients

For example, variables 5 (road speed) and 10 (power to engine ratio) contribute heavily to the first principal component while variables 1 (wieght) and 3 (width) contributes most strongly to the second principal component. Variables 2, 4, 6, 7, 8, 9 are not as important. The weights accorded each variable in the 10 factors (principal components) are shown in Figure 8. The complete SPSS program is listed in Appendix C.

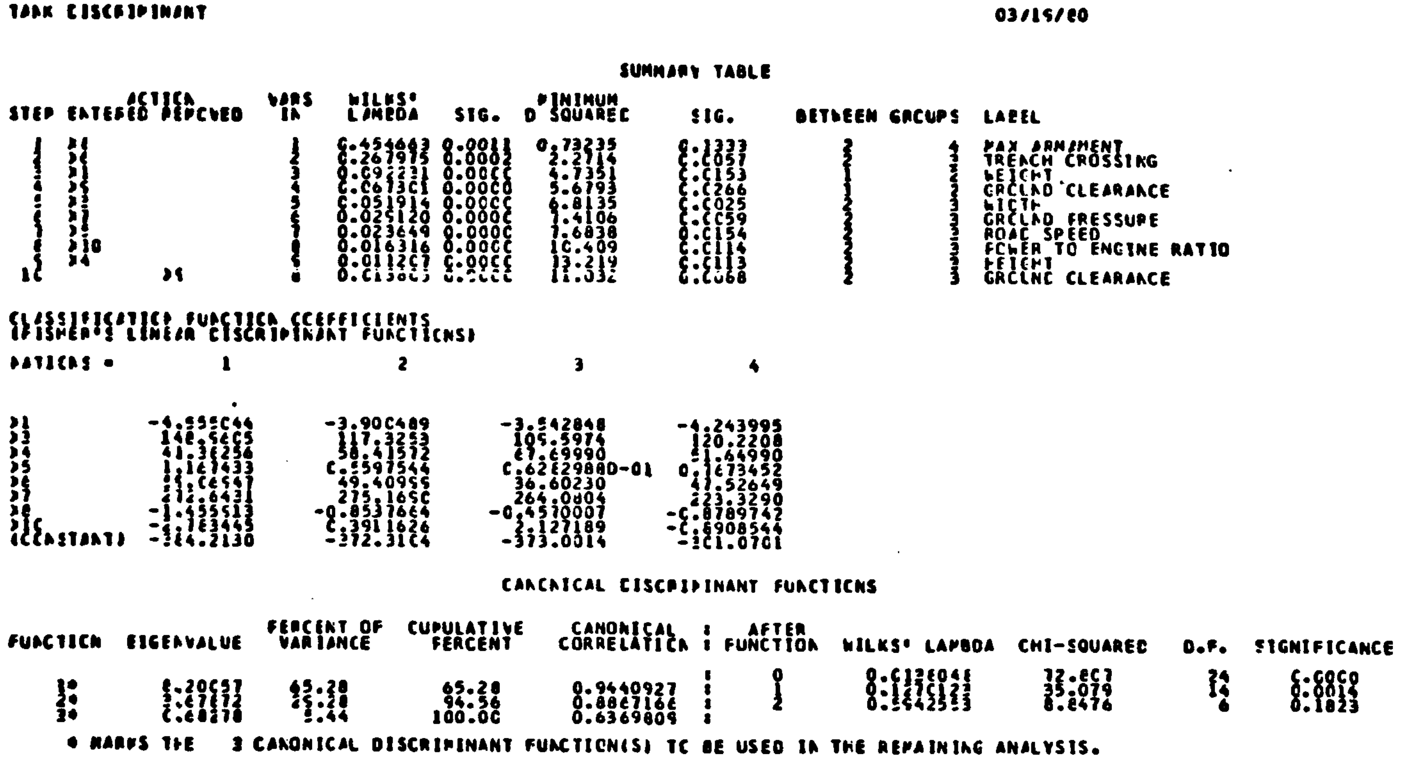
D. DISCRIMINANT ANALYSIS

The SPSS subprogram DISCRIMINANT was used to determine that function or those functions of the 10 variables that best discriminant among the four clusters determined in previous section.

The maximum number of discriminant functions to be derived is either one less than the number of groups or equal to the number of discriminating variables. This subprogram provides two measures for judging the importance of discriminant functions. One of these is the relative percentage of the eigenvalue associated with the function. It is a measure of the relative importance of the function. The sum of the eigenvalues is a measure of total variance existing in the discriminating variables. Since discriminant functions are derived in order of their importance, this process can be stopped whenever the relative percentage is judged to be too small. Of course, there is no fixed rule for deciding whatis too small. In this research, we selected arbitrary, a significance level of 0.10. The output shown in Figure 9 suggests that we therefore consider only the first two discriminant functions.

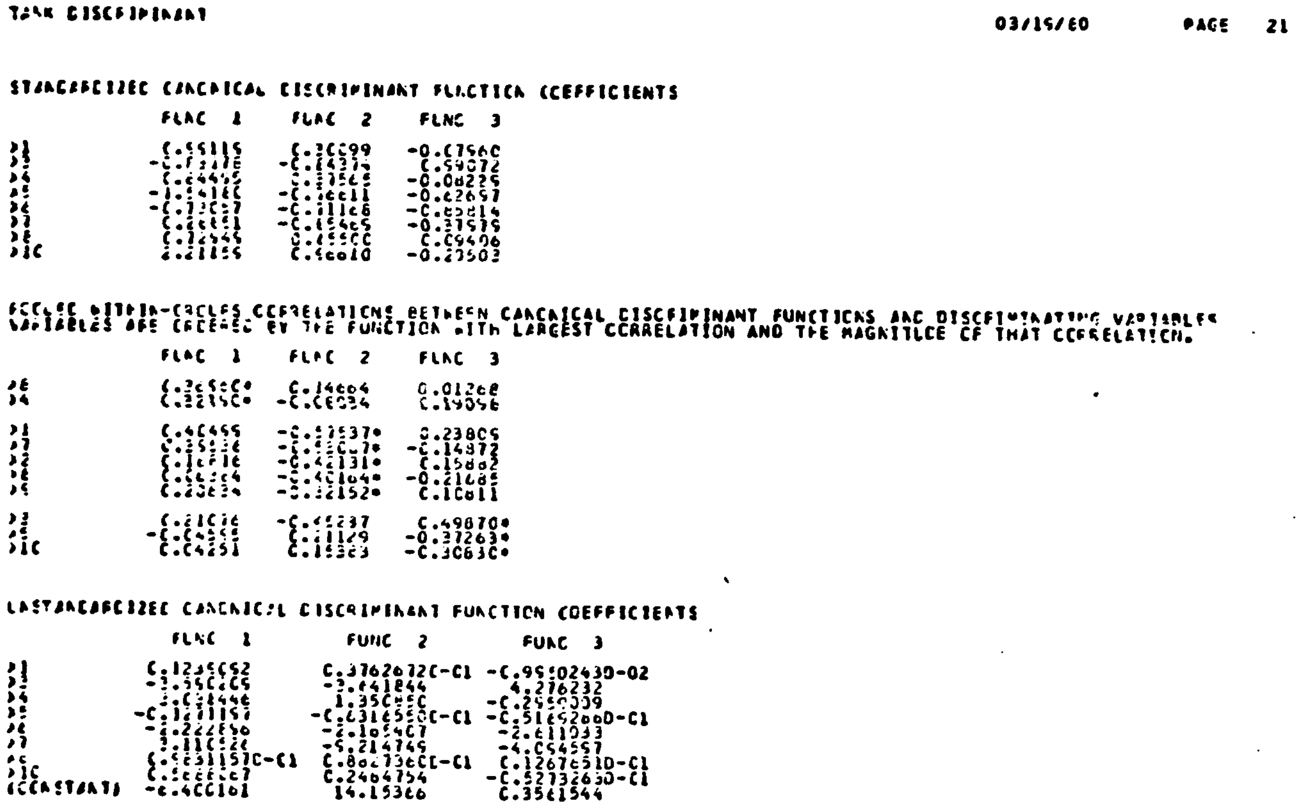
The second measure judging the importance of a discriminant function is its associated canonical correlation. The canonical correlation is a measure of association between the single discriminant function and the set of (g-1) dummy variables which define the g group memberships. It tells us how closely the function and the group variable are related, which is just another measure of the function's ability to discriminate among the groups. From Figure 10, the first two discriminant functions are each highly correlated with the groups but the third has only a moderate correlation.

The next criterion for eliminating discriminant functions is to test for the statistical significance of discriminating information not already accounted for by the earlier functions. As each function is derived, starting with no (zero) functions, Wilks' lambda is computed. Lambda is an inverse measure of the discriminating power in original variables which has not yet been removed by the discriminant functions - the larger lambda is, the less is the information remaining. Lambda can be transformed into a chi-square statistic for an easy test of statistical significance. In Figure 9, Wilks' lambda was .594 after the first two functions had been derived. This corresponds to a chi-square of 8.8476 with a probability level of .1823.



```
TANK DISCRIMINANT                                                                               03/19/80

                                                    SUMMARY TABLE

          ACTION        VARS    WILKS'               MINIMUM
STEP ENTERED REMOVED     IN     LAMBDA      SIG.  D SQUARED        SIG.      BETWEEN GROUPS   LABEL

  1   X9                  1    0.454643   0.0011   0.73235        0.1333         2         4   MAX ARMAMENT
  2   X6                  2    0.267975   0.0002   2.2714         0.0057         2         3   TRENCH CROSSING
  3   X1                  3    0.092231   0.0000   4.7351         0.0153         1         2   WEIGHT
  4   X5                  4    0.067301   0.0000   5.6793         0.0266         1         2   GROUND CLEARANCE
  5   X3                  5    0.051914   0.0000   6.8135         0.0025         2         3   WIDTH
  6   X7                  6    0.025120   0.0000   7.4106         0.0059         2         3   GROUND PRESSURE
  7   X8                  7    0.023649   0.0000   7.6838         0.0154         2         3   ROAD SPEED
  8   X10                 8    0.016316   0.0000   10.409         0.0114         2         3   POWER TO ENGINE RATIO
  9   X4                  9    0.011207   0.0000   13.219         0.0113         2         3   HEIGHT
 10          X9           8    0.013603   0.0000   11.032         0.0068         2         3   GROUND CLEARANCE

CLASSIFICATION FUNCTION COEFFICIENTS
(FISHER'S LINEAR DISCRIMINANT FUNCTIONS)

NATIONS =           1              2              3              4


X1             -4.555044      -3.900489      -3.542848      -4.243995
X3              148.5605       117.3253       105.5974       120.2208
X4              41.36256       58.41572       67.69990       51.64990
X5              1.167433       0.5597544   0.6262988D-01  0.1673452
X6              55.04547       49.40955       36.60230       47.52649
X7              212.6431       275.1650       264.0804       223.3290
X8             -1.455513      -0.8537664     -0.4570007     -0.8789742
X10            -4.763445       0.3911626      2.127189      -0.6908544
(CONSTANT)     -384.2130      -372.3104      -373.0014      -301.0701

                                          CANONICAL DISCRIMINANT FUNCTIONS

                        PERCENT OF   CUMULATIVE     CANONICAL  :   AFTER
FUNCTION   EIGENVALUE    VARIANCE      PERCENT    CORRELATION  : FUNCTION   WILKS' LAMBDA   CHI-SQUARED     D.F.   SIGNIFICANCE

                                                               :     0        0.0128041        72.807         24       0.0000
   1*        8.20057      65.28         65.28      0.9440927   :     1        0.1270123        35.079         14       0.0014
   2*        3.67672      29.28         94.56      0.8867166   :     2        0.5942553        8.8476          6       0.1823
   3*        0.68270       5.44        100.00      0.6369809   :

  * MARKS THE   3 CANONICAL DISCRIMINANT FUNCTION(S) TO BE USED IN THE REMAINING ANALYSIS.
```

Figure 9. Summary Table of Discriminant Analysis on Tank



```
TASK DISCRIMINANT                                                           03/15/80        PAGE   21

STANDARDIZED CANONICAL DISCRIMINANT FUNCTION COEFFICIENTS

              FUNC  1     FUNC  2     FUNC  3

X1            0.55115     0.30099    -0.07560
X3           -0.82478    -0.84379     0.59072
X4            0.64495     0.37568    -0.08225
X5           -1.54160    -0.26611    -0.62697
X6           -0.73027    -0.11168    -0.65814
X7            0.66651    -0.65465    -0.37979
X8            0.32645     0.45500     0.09406
X10           0.21155     0.56010    -0.23503

POOLED WITHIN-GROUPS CORRELATIONS BETWEEN CANONICAL DISCRIMINANT FUNCTIONS AND DISCRIMINATING VARIABLES
VARIABLES ARE ORDERED BY THE FUNCTION WITH LARGEST CORRELATION AND THE MAGNITUDE OF THAT CORRELATION.

              FUNC  1     FUNC  2     FUNC  3

X8            0.36560*    0.14604     0.01268
X4            0.32150*   -0.06034     0.19056

X1            0.40455    -0.57537*    0.23805
X7            0.35516    -0.53047*   -0.14872
X2            0.16716    -0.42131*    0.15802
X6            0.56264    -0.40104*   -0.21685
X5            0.20634    -0.32152*    0.10611

X3            0.21036    -0.45237     0.49870*
X9           -0.04555     0.21129    -0.37263*
X10           0.04251     0.15323    -0.30630*

UNSTANDARDIZED CANONICAL DISCRIMINANT FUNCTION COEFFICIENTS

              FUNC  1            FUNC  2            FUNC  3

X1            0.1235552          0.3762672D-01     -0.9502430D-02
X3           -3.550605          -3.641844            4.276232
X4            2.024446           1.350850           -0.2550009
X5           -0.1611157         -0.6316550D-01     -0.5169200D-01
X6           -4.222850          -2.105407           -2.611033
X7            3.110526          -5.214749           -4.054557
X8            0.5631157D-01      0.8027360D-01      0.1267651D-01
X10           0.5666667          0.2464754          -0.5273263D-01
(CONSTANT)   -6.400101          14.15366             0.3561544
```

Figure 10. Canonical Discriminant Function Coefficients

This means that a lambda of this magnitude or smaller has a .1823 probability of occurring due to the chances of sampling even if there was no further information to be accounted for by a third function in the population. Clearly, a third function is not statistically significant in this case.

The standarized discriminant function coefficients corresponding to the values of the v_{ij}'s discussed in the previous section are used to compute the discriminant score for a case (observation) in which the original discriminating variables are in standard form. The discriminant score is computed by multiplying each discriminating variable by its corresponding coefficient and adding together these products. There is a separate score for each observation on each function. The coefficients have been derived in such a way that the discriminant scores produced are in standard form.

When the sign if ignored, each standard discriminant function coefficient represents the relative contribution of its associated variable to that function. The sign merely denotes whether the variable is making a positive or negative contribution.

A graphical presentation is shown in Figure 11 using the first and the second canonical discriminant function as the axis. From this scatterplot, we can easily see that Soviet tanks (labelled 1) are well distinguished from the all of the others using only the first two discriminant

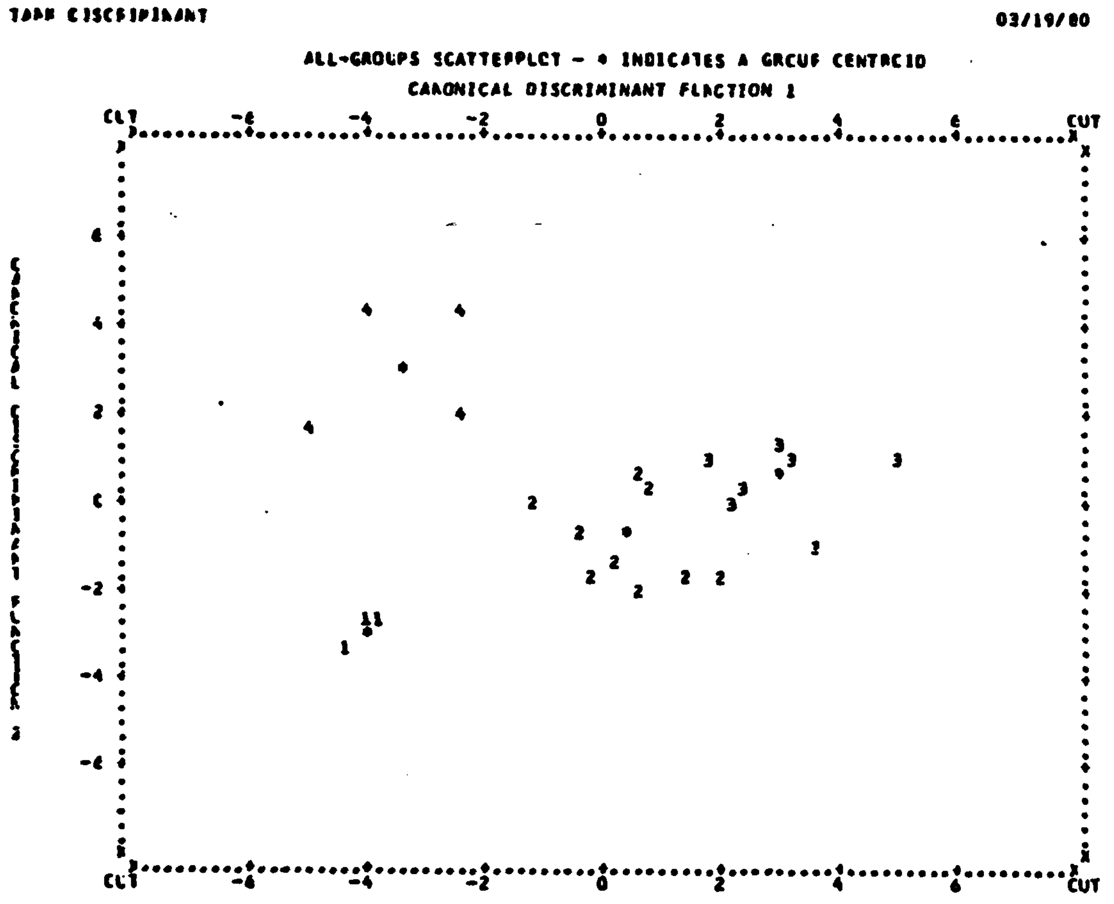


Figure 11. Scatter Plot for all Groups

functions. Also, all the lightweight tanks are clearly separated from the others. The distinction between groups 2 and 3 is also clear though not separated from each other as much as from groups 1 and 4. The complete SPSS program for the discriminant analysis is listed in Appendix D.

VII. CONCLUSION

The multivariate analysis techniques of cluster analysis, principal components analysis and discriminant analysis are useful in real world problems for examining observations on each of several dimension. Each of the techniques is related mathematically to the others, and each complements the other in explaining the data.

Computer software is readily available in many sources. The software used in this thesis for hierarchical clustering, principal components analysis, and discriminant analysis was from the IMSL package and SPSS. For nonhierarchical clustering, we used the FORTRAN program developed by McRae (16). All of this softw.re is readily available and documented at the Naval Postgraduate School.

Appendix A:

```
C
C                   HIERACHICAL CLUSTERING
C
C     THIS IS A PROGRAM FOR HIERACHOCAL CLUSTERING
C
C        VARIABLE DESCRIPTION
C        ND          INPUT NUMBER OF DATA POINTS TO BE CLUSTERED.
C                    ND-1 CLUSTERD ARE FORMED NUMBERED CONSECUTIVELY
C                    FROM ND+1 TO ND+(ND-1). ND MUST BE GREATER THAN 2.
C                    ND IN THE RANGE 2,3,...,500 IS ALLOWED
C                    (SEE REMARKS)
C        IOPT        INPUT OPTIONS VECTOR OF LENGTH 2
C                       IOPT(1)=0 IMPLIES SINGLE-LINKAGE DESIRED.
C                          OTHERWISE, COMPLETE-LINKAGE IS DESIRED
C                       IOPT(2)=0 IMPLIES THE SIMILALITIES ARE DISTANCE-
C                         LIKE (I.E. SMALLER IMPLIES CLOSER). OTHERWISE,
C                         THE SIMILARITIES ARE ASSUMED TO BE POSITIVE
C                         CORRELATION-LIKE SIMILARITIES.(SEE REMARKS)
C        XSIM        INPUT/OUTPUT VECTOR OF LENGTH ((ND+1)*ND)/2
C                       CONTAINING THE SIMILARITY MATRIX IN SYMMETRIC
C                       STORAGE MODE. FOR I GREATER THAN J,
C                       XSIM(((I-1)*I)/I+J) CONTAINS THE SIMILARITY OF
C                       THE I-TH AND J-TH DATA POINTS. ON INPUT, THE
C                       DIAGONAL ELEMENTS (SIMILARITY OF ITSELF) ARE
C                       ARBITRARY AND DO NOT NEED TO BE DEFINED.
C                       XSIM IS DESTROYED ON OUTPUT.
C        CLEVEL      OUTPUT VECTOR OF LENGTH ND-1. CLEVEL(K) CONTAINS THE
C                       SIMILARITY LEVEL AT WHICH CLUSTER ND+K WAS FORMED.
      DIMENSION XSIM(500),IOPT(2),CLEVEL(500),ICLSON(500),TITLE(20),
     1ICRSON(500),IPTR(100),IOUT(100),IND(4),CLVSK(100),NCLFST(100),
     2LEFTRT(100),STARST(100),YSIM(2)
C                    THIS WOULD BE THE INPUT VECTOR FOR SUBPROGRAM USTREE.
C        ICLSON      OUTPUT VECTOR OF LEFTSONS OF LENGTH ND-1. CLUSTER
C                       NUMBER ND+K WAS FORMED BY MERGING CLUSTER ICLSON(K)
C                       WITH CLUSTER ICRSON(K).
C                    THIS WOULD BE THE INPUT VECTOR FOR SUBPROGRAM USTREE.
C        ICRSON      OUTPUT VECTOR OF RIGHTSONS OF LENGTH ND-1. THE
C                       RIGHTSON OF CLUSTER ND+K IS CONTAINED IN ICRSON(K).
C                    THIS WOULD BE THE INPUT VECTOR FOR SUBPROGRAM USTREE.
C        IPTR        WORK VECTOR OF LENGTH ND.
C        IER         ERROR PARAMETER. (OUTPUT)
C                    TERMINAL ERROR
C                       IER =129 INDICATES ND WAS LESS THAN 3.
C        IND         INPUT VECTOR OF LENGTH 4. IND(1) CONTAINS WHEN I=1,
C                       THE HEAD NODE OF THE SUBTREE TO BE PRINTED.
C                       MUST BE BETWEEN ND AND 2*ND,EXCLUSIVELY. (REMARKS)
C                       I=2, NUMBER OF PRINTABLE SPACES PER LINE ON THE
C                       OUTPUT (PRINTER) DEVICE. IND(2) MUST EXCEED 4.
```

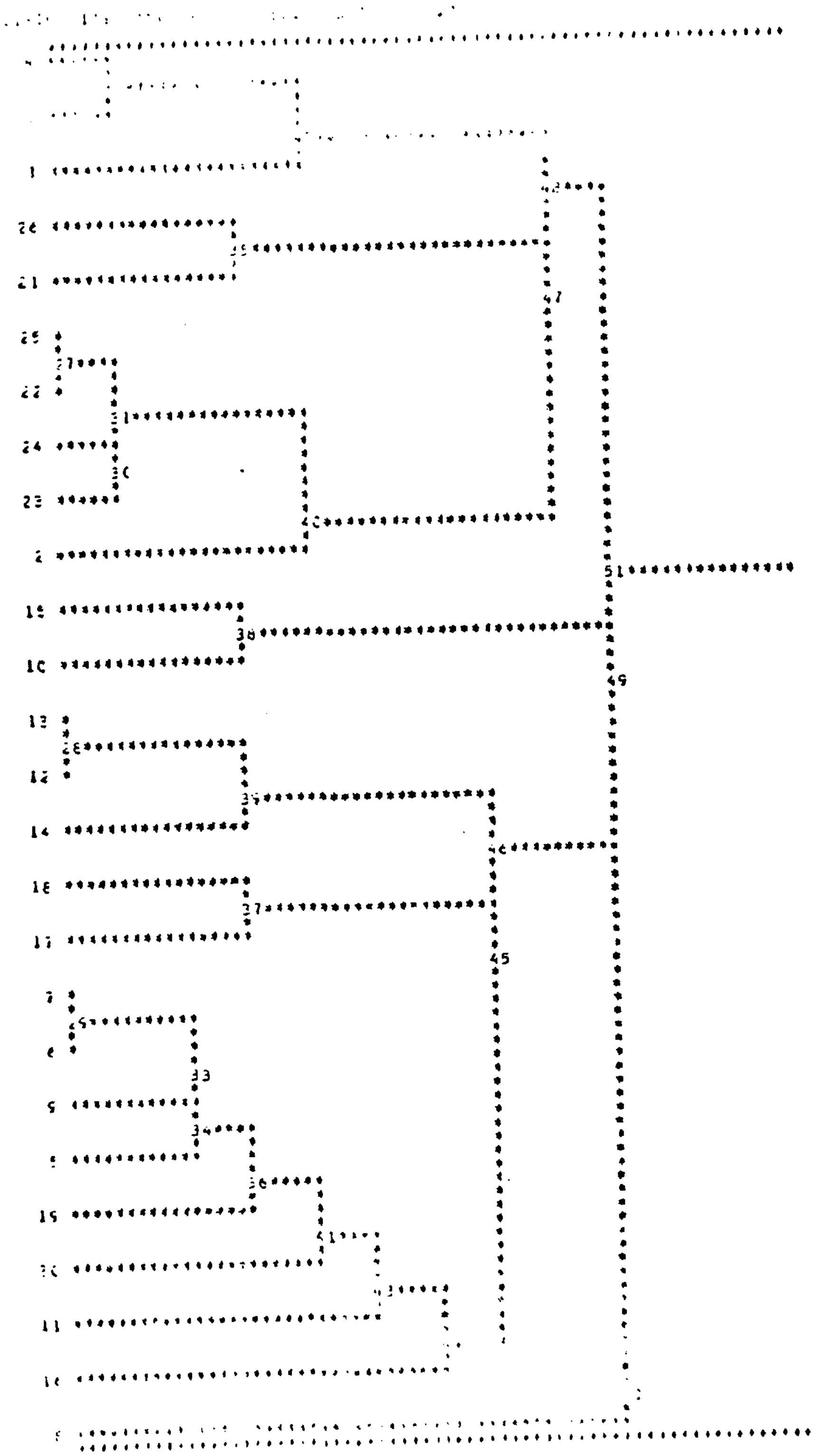
```
C                  I=3, NUMBER OF HORIZONTAL SLICES OF TREE DESIRED
C                  TO PROVIDE THE NECESSARY DETAIL.
C                  INC(3) MUST BE POSITIVE.
C                  I=4, NUMBER OF FILLER LINES PRINTED BETWEEN NODE
C                  LINES (1 USUALLY SUFFICIENT).
C     YSIM      INPUT VECTOR OF LENGTH 2 CONTAINING THE INTERVAL ON
C                  THE VERTICAL SCALE USED TO PLOT THE TREE
C                  (SEE VECTOR CLEVEL). LEVEL YSIM(1) IS WHERE THE
C                  TERMINAL NODES (FOR IMSLCLUSTERING ROUTINES, THE
C                  DATA POINTS) ARE PRINTED. THE OTHER INTERVAL
C                  ENDPOINT YSIM(2) SHOULD INCLUDE THE LEVEL FOR THE
C                  HEAD NODE (INC(1)). SEE REMARKS.
      NAMELIST /NAM1/XSIM
C       ICUT      WORK VECTOR OF LENGTH INC(2)-4.
C       CLVLSK    WORK VECTOR OF LENGTH ND.
C       NCLRST    WORK VECTOR OF LENGTH ND.
C       LEFTRT    WORK VECTOR OF LENGTH ND.
C       STARST    WORK VECTOR OF LENGTH ND.
C
C  REMARKS 1. THE DATA CLUSTERS ARE NUMBERED 1TO ND. THE ND-1 CLUSTERS
C       FORMED BY MERGING ARE NUMBERED ND+1 TO ND+(ND-1) AND DECREASE
C       IN SIMILARITY, MAKING IT EASY TO IDENTIFY THE MOST SIMILAR
C       CLUSTERS.
C
C       2. SIMILARITIES GENERALLY SHOULD BE NONNEGATIVE. RAW
C       CORRELATIONS TAKE ON VALUES R WHERE R LIES IN THE CLOSED
C       INTERVAL (-1, 1)) AND SHOULD BE MADE POSITIVE. IF R=1 AND
C       R=-1 BOTH MEAN HIGH SIMILARITY, THEN THE TRANSFORMATIONS,
C       RR EQUAL TO THE ABSOLUTE VALUE OF R, OR, RR EQUAL TO R SQUARED,
C       ARE APPROPRIATE. IF R=-1 REPRESENTE VERY LOW SIMILARITY, THEN
C       THE TRANSFORMATION, RR=1-R, BECOMES A DISTANCE-LIKE SIMILARITY.
C
C       3. NOTE THAT THE ORIGINAL DATA MATRIX OF THE USER DOES NOT
C       ENTER CCLINK, ALLOWING THE USER TO DEFINE, VIA XSIM,
C       WHATEVER MEASURE OF SIMILARITY SEEMS MOST APPROPRIATE.
C
C       4. THE TREE MAY BE TOO LARGE TO FIT IN INC(2) SPACES
C       REPRESENTING THE INTERVAL (YSIM(1),YSIM(2)). IF SO, USTREE CAN
C       PRINT THE TREE IN IND(3) SLICES OF THIS INTERVAL WHICH MAY BE
C       CUT AND TAPED TOGETHER.
C
      READ(5,110) (TITLE(I),I=1,20)
C       5. TO PRINT THE ENTIRE TREE FROM IMSL SUBROUTINE CCLINK, THE
C       HEAD NODE INC(1) =ND+(ND-1).
C
C       6. OUTPUT IS WRITTEN TO THE UNIT SPECIFIED BY IMSL ROUTINE
C       UGETIO. SEE THE UGETIO DOCUMENT FOR DETAILS.
C
```

```
C         7. REVERSALS (IER=130) MAY OCCUR IN TWO WAYS. FIRST, TWO NODES
C         MAY BE JOINED AT A LOWER LEVEL (CLOSER TO YSIM(1)). SECOND, THE
C         LEVEL OF THE HEAD NODE MAY LIE ABOVE THE INTERVAL
C         (YSIM(1), YSIM(2)).
C          8. FOR PROPER DISPLAY, THE TREE CREATED BY USTREE SHOULD BE
C         -URNED TO AN UPRIGHT POSITION.
C
      WRITE(6,111) (TITLE(I),I=1,20)
C
C           READ THE NUMBER OF OBSERVATION AND
C           TWO INPUT OPTIONS.
C
      READ(5,112) ND,IOPT(1),IOPT(2)
      WRITE(6,200) ND
      IF(ND.LE.2) GO TO 50
      M=((ND+1)*ND)/2
      IF(IOPT(1).EQ.0) GO TO 30
      WRITE(6,113)
   29 IF(IOPT(2).EQ.0) GO TO 31
      WRITE(6,114)
      GO TO 32
   30 WRITE(6,115)
      GO TO 29
   31 WRITE(6,116)
   32 WRITE(6,13) M
C
C           READ INPUT DATA MATRIX BY THE
C           NAME LIST FORMAT.
C           AN ARRY OF VECTOR FROM THE LOWER TRIANGULAR MATRIX.
C
      READ (5,NAM1)
      WRITE (6,300)
      JJJJ=1
      DO 10 I=1,ND
      JJKK=JJJJ+I-1
      WRITE (6,101) I,(XSIM(JK),JK=JJJJ,JJKK)
   10 JJJJ=JJJJ+I
      WRITE (6,100) (I,I=1,ND)
C
C           CALL IMSL SUBPROGRAM OCLINK
C
      CALL OCLINK(ND,IOPT,XSIM,CLEVEL,ICLSON,ICRSON,IPTR,IER)
      WRITE(6,1)
C
C           WRITE THE SIMILARITY LEVEL,ICLSON AND
C           ICRSON TO CONSTRUCT A TREE.
C
```

```
      WRITE(6,2) (CLEVEL(I),ICLSON(I),ICRSON(I),I=1,25)
      INC(1)=2*NC-1
      INC(2)=70
      INC(3)=1
      INC(4)=1
      YSIM(1)=CLEVEL(1)
      YSIM(2)=0.0
C
C           CALL IMSL USTREE FOR THE GRAPHICAL
C           PRESENTATION.
C
      CALL USTREE(ND,ICLSON,ICRSON,CLEVEL,IND,YSIM,ICUT,C353SM,NCLRST,
     1LEFTRT,STARST,IER)
      GO TO 60
   50 WRITE(6,900)
   60 STOP
    1 FORMAT(1H1,5X,'CLEVEL',6X,'ICLSON',6X,'ICRSON')
    2 FORMAT(2X,F10.1,2X,I7,5X,I7)
   13 FORMAT('0','NUMBER OF OBSERVATIONS IN XSIM ARE',I6)
  100 FORMAT(/,5X,26I4)
  101 FORMAT('0',I5,26F4.0)
  110 FORMAT(20A4)
  111 FORMAT('1',20A4)
  112 FORMAT(I4,2I1)
  113 FORMAT('0 COMPLETE-LINKAGE IS PERFORMED.')
  114 FORMAT('0 SIMILLARITIES ARE POSITIVE CORRELATION-LIKE.')
  115 FORMAT('0 SINGLE LINKAGE IS PERFORMED.')
  116 FORMAT('0 SIMILLARITIES ARE DISTANCE-LIKE.')
  200 FORMAT('0','NUMBER OF DATA POINTS ARE',I4)
  300 FORMAT('0','INITIAL DISTANCE MATRIX',//)
  900 FORMAT('0 YOUR DATA ARE TOO SMALL ')
      END
```

```
 (HIERARCHICAL CLUSTERING ON UTILITY DATA)
NUMBER OF DATA POINTS ARE 26
 SINGLE LINKAGE IS PERFORMED.
 SIMILARITIES ARE POSITIVE CORRELATION-LIKE.
NUMBER OF OBSERVATIONS IN XSIM ARE   351
INITIAL DISTANCE MATRIX

  1 50.
  2  4. 50.
  3  8.  1. 50.
  4  7.  1. 11. 50.
  5  0.  0.  0.  0. 50.
  6  0.  0.  0.  0.  8. 50.
  7  0.  0.  0.  0.  8. 12. 50.
  8  0.  1.  0.  0.  3.  3.  3. 50.
  9  0.  0.  0.  0. 10. 10. 10.  3. 50.
 10  0.  0.  1.  1.  0.  0.  0.  3.  0. 50.
 11  0.  0.  0.  0.  6.  6.  6.  2.  7.  0. 50.
 12  0.  0.  0.  0.  1.  3.  3.  1.  2.  0.  5. 50.
 13  0.  0.  0.  0.  1.  3.  3.  1.  2.  0.  5. 12. 50.
 14  0.  0.  0.  0.  2.  5.  5.  2.  4.  0.  5.  9.  9. 50.
 15  0.  0.  0.  0.  0.  0.  0.  3.  0.  9.  0.  0.  0.  0. 50.
 16  0.  0.  0.  0.  3.  5.  5.  0.  4.  1.  6.  3.  3.  3.  3. 50.
 17  0.  0.  0.  0.  2.  1.  1.  1.  2.  0.  5.  3.  3.  2.  0.  2. 50.
 18  0.  0.  0.  0.  1.  1.  1.  1.  1.  0.  4.  3.  4.  2.  0.  1.  9. 50.
 19  0.  0.  0.  0.  9.  5.  5.  2.  7.  0.  3.  0.  0.  0.  0.  2.  2.  1. 50.
 20  2.  3.  1.  1.  5.  1.  1.  1.  3.  0.  2.  0.  0.  0.  0.  0.  2.  1.  8. 50.
 21  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  1.  0.  0.  0.  0.  0.  2.  0.  3.  3. 50.
 22  2.  8.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  2.  2. 50.
 23  2.  7.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  2.  3. 11. 50.
 24  3.  6.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  2.  3. 10. 11. 50.
 25  2.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  2.  2. 12. 11. 10. 50.
 26  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  2.  2.  9.  3.  4.  4.  3. 50.
      1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26
```

```
CLEVEL        ICLSON        ICRSON
  12.0          25            22
  12.0          13            12
  12.0           7             6
  11.0          24            23
  11.0          27            30
  11.0           4             3
  10.0          29             9
  10.0          33             5
   9.0          26            21
   9.0          34            19
   9.0          18            17
   9.0          15            10
   9.0          28            14
   8.0          31             2
   8.0          36            20
   8.0          32             1
   7.0          41            11
   6.0          43            16
   5.0          37            44
   5.0          39            45
   4.0          35            40
   4.0          42            47
   3.0          38            46
   3.0          49             8
   3.0          48            50
```


1
26
35
21
47
42
25
27
22
31
24
30
23
40
2
51
15
38
10
49
13
28
12
39
14
46
18
37
17
45
7
6
33
9
34
5
36
19
41
20
43
11
16

Appendix B:

```
C                                                                       RRR00120
C     K-MEANS ITERATIVE CLUSTERING PROGRAM BY D.J. MCRAE                RRR00010
C                                                                       RRR00020
C     THIS PROGRAM ENABLES THE USER TO CLUSTER A DATA MATRIX            RRR00030
C     UP TO SIZE (600 X 20) INTO A MAXIMUM OF 20 CLUSTERS.              RRR00040
C     SEVERAL OPTIONS ARE AVAILABLE WITH RESPECT TO THE                 RRR00050
C     METHOD OF COMPUTATION AND CRITERIA TO ACCOMPLISH THE              RRR00060
C     CLUSTERING.  THE PROGRAM MAY BE USED ON EITHER                    RRR00070
C     THE CP/CMS TERMINALS OR BY THE CARD READER. THE APPROPRIATE       RRR00080
C     METHOD FOR EACH IS DESCRIBED BELOW.                               RRR00090
C                                                                       RRR00100
C     CP/CMS TERMINAL USE                                               RRR00110
C             THE PROGRAM SHOULD BE READ INTO CP AS IS, BUT             RRR00130
C     PRECEDED BY THE FOLLOWING CARDS, STARTING IN COLUMN ONE           RRR00140
C                                                                       RRR00150
C             CP67USERID  XXXXG                                         RRR00160
C             OFFLINE READ RRREAD FORTRAN                               RRR00170
C                                                                       RRR00180
C             THE DATA MATRIX CAN BE PLACED IN THE COMPUTER             RRR00190
C     EITHER BY USE OF THE OFFLINE READ OR BY TYPING IN THE             RRR00200
C     INFORMATION ON THE TERMINAL.  IF OFFLINE READ IS USED THE DATA    RRR00210
C     MUST BE IN THE SPECIFIED FORMAT PRECEDED BY THE FOLLOWING         RRR00220
C     TWO CARDS, STARTING IN COLUMN ONE:
C                                                                       RRR00240
C             CP67USERID  XXXXG
C             OFFLINE READ FILE FT04FYYY
C                                                                       RRR00270
C             THE PROGRAM CAN THEN BE EXECUTED BY FIRST ISSUING THE     RRR00280
C     CMS COMMAND
C                                 F RRREAD
C     FOLLOWED BY
C                                 $ RRREAD
C                                                                       RRR00330
C                                                                       RRR00340
C     CS/CARD READER USE
C                                                                       RRR00360
C             THE PROGRAM MUST CONTAIN THE FOLLOWING INFORMATION        RRR00370
C     IN THE FOLLOWING ORDER:
C                   STANDARD GREEN JOB CARD,TIME=(AS DESIRED)
C                   // EXEC FORTCLG,REGION.GO=200K                      RRR00400
C                   //FORT.SYSIN DD *                                   RRR00410
C                       MAIN PROGRAM                                    RRR00420
C                   /*                                                  RRR00430
C                   //GO.SYSIN DD *                                     RRR00440
C                       DATA DECK                                       RRR00450
C                   /*                                                  RRR00460
C                                                                       RRR00470
C                                                                       RRR00480
```

```
C     DATA  INPUT DECK                                                  RRR00490
C                                                                       RRR00500
C               THE FIRST CARD OF THE DATA DECK IS THE TITLE CARD.     RRR00510
C     IT MAY CONTAIN ANY ALPHA NUMERIC TITLE IN COLUMS 1 - 80.          RRR00520
C                                                                       RRR00530
C               THE SECOND CARD IS THE PROBLEM CARD.  IT CONTAINS      RRR00540
C     INTEGERS IN THE FIRST 13 COLLUMNS IN THE FOLLOWING MANNER:        RRR00550
C              COL 1-4:    NUMBER OF OBSERVATIONS                       RRR00560
C              COL 5-6:    NUMBER OF VARIABLES PER OBSERVATION          RRR00570
C              COL 7-8:    NUMBER OF CLUSTERS DESIRED                   RRR00580
C              COL 9:      CRITERION TO BE USED IN THE EVALUATION:      RRR00590
C                               1 = TRACE W                             RRR00600
C                               2 = DETERMINANT W                       RRR00610
C                               3 = LARGEST ROOT OF W(INVERSE)*B        RRR00620
C                               4 = TRACE W(INVERSE)*B                  RRR00630
C              COL 10:     STANDARDIZATION PARAMETER:                   RRR00640
C                               0 = THE PROGRAM WILL NOT STANDARDIZE DATA RRR00650
C                               1 = THE PROGRAM WILL STANDARDIZE DATA   RRR00660
C              COL 11:     DISTANCE PARAMETER                           RRR00670
C                               0 = EUCLIDIAN DISTANCE                  RRR00680
C                               1 = SCALED EUCLIDIAN DISTANCE           RRR00690
C                               2 = MAHALANOBIS DISTANCE                RRR00700
C              COL 12:     DATA PARAMETER                               RRR00710
C                               0 = DATA IS TO BE READ IN               RRR00720
C                               1 = NO DATA READ IN; PREVIOUS DATA IS   RRR00730
C                                   TO BE REANALYZED USING CURRENT PROBLEM RRR00740
C              COL 13:     TIMING PARAMETER                             RRR00750
C                               0 = NO OBSERVATIONS CONSIDERED IN       RRR00760
C                                   INDIVIDUAL SWITCHES                 RRR00770
C                           1 - 8 = AN INCREASING NUMBER OF OBSERVATIONS RRR00780
C                               9 = ALL OBSERVATIONS USED               RRR00790
C              NOTE: THE TIMING PARAMETER DETERMINED HOW EXTENSIVE THE  RRR00800
C                    DOUBLE CHECK OF THE CLUSTER SOLUTION IS TO BE.     RRR00810
C                    NORMALLY '9' IS USED UNLESS THE DATA MATRIX IS LARGE RRR00820
C                                                                       RRR00830
C     THIS CARD MUST BE IN (I4,2I2,5I1) FORMAT                          RRR00840
C                                                                       RRR00850
C               THE THIRD CARD IS THE VARIABLE FORMAT CARD.  THIS CARD  RRR00860
C     MUST BE PRESENT UNLESS THE DATA PARAMETER IN THE PROBLEM          RRR00870
C     CARD IS '1'.  THE FORMAT STATEMENT IS THAT TYPE WHICH THE DATA    RRR00880
C     CARDS ARE IN.  FOR EXAMPLE IF THE DATA CARD HAS DATA:             RRR00890
C                 ABCD     1.2     1.55                                 RRR00900
C     STARTING IN COL 1, THE VARIABLE FORMAT  CARD WOULD                RRR00910
C     CONTAIN THE FOLLOWING STARTING IN COL 1:                          RRR00920
C                  (1A4,4X,F3.1,4X,F4.2)                                RRR00930
C                                                                       RRR00940
C               THE NEXT CARDS ARE THE DATA CARDS.  THEY MAY CONTAIN    RRR00950
C     DATA IN COLUMNS 1 - 72.                                           RRR00960
```

```
C                                                                       RRRC0970
C            FOR REANALYSIS OF THE SAME DATA,  ADD A NEW TITLE CARD    RRRC0980
C     FOLLOWED BY A NEW PROBLEM CARD,WITH A '1' IN COLUMN 11           RRRC0990
C     OF THE NEW PROBLEM CARD.                                         RRRO1000
C            FOR ANALYSIS OF NEW DATA, SIMPLY PLACE THE NEW DATA       RRRO1010
C     SET, INCLUDING NEW TITLE CARD, PROBLEM CARD, AND FORMAT CARD,    RRRO1020
C     AFTER THE PREVIOUS SET                                           RRRO1030
C                                                                       RRRO1040
C     FOR THE PROGRAM TO EXIT NORMALLY, TWO BLANK CARDS MUST FOLLOW    RRRO1050
C     THE LAST DATA BATCH.                                             RRRO1060
C                                                                       RRRO1070
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600), RRRO1080
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20), RRRO1090
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),BT(20,20), RRRO1100
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20)          RRRO1110
C                                                                       RRRO1120
C                                                                       RRRO1130
C     DESCRIPTION OF COMMON AREA:                                      RRRO1140
C                                                                       RRRO1150
C         NOBS = NUMBER OF OBSERVATIONS                                RRRO1160
C         NVARS = NUMBER OF VARIABLES                                  RRRO1170
C         NGPS = NUMBER OF CLUSTERS                                    RRRO1180
C         ICRIT = CRITERION TO BE OPTIMIZED (SEE ABOVE)                RRRO1190
C         NOSTAN = STANDARDIZATION PARAMETER (SEE ABOVE)               RRRO1200
C         IDIST = DISTANCE PARAMETER  (SEE ABOVE)                      RRRO1210
C         IFINE = ESCAPE PARAMETER:  IF IFINE IS SET EQUAL TO '1',     RRRO1220
C                 SOMETHING IS WRONG AND THE APPROPRIATE ERROR         RRRO1230
C                 MESSAGE IS PRINTED OUT; THE PROGRAM GOES ON TO THE   RRRO1240
C                 NEXT PROBLEM                                         RRRO1250
C          KTIME = TIMING PARAMETER (SEE ABOVE)                        RRRO1260
C         IDENT = OBSERVATION IDENTIFICATIONS (A4 FORMAT):  READ FROM  RRRO1270
C                 EACH DATA CARD                                       RRRO1280
C         DATA = AREA IN WHICH THE DATA MATRIX IS STORED               RRRO1290
C         T = CROSS-PRODUCTS MATRIX                                    RRRO1300
C         B = BETWEEN-CLUSTERS MATRIX                                  RRRO1310
C         W = WITHIN-CLUSTERS MATRIX                                   RRRO1320
C         WFCT = CHOLESKY FACTOR OF THE WITHIN-CLUSTERS MATRIX         RRRO1330
C         SVCEN = CLUSTER CENTERS (MEANS)                              RRRO1340
C         IDATA = CLUSTER IDENTIFICATION FOR EACH OBSERVATION          RRRO1350
C         NISV = CLUSTER SIZES (NUMBER OF OBSERVATIONS IN EACH CLUSTER) RRRO1360
C         VMEAN = VARIABLE MEANS                                       RRRO1370
C         SD = VARIABLE STANDARD DEVIATIONS                            RRRO1380
C          XVEC = TEMPORARY STORAGE                                    RRRO1390
C          YVEC = TEMPORARY STORAGE                                    RRRO1400
      NNN=24
C         BT = TEMPORARY STORAGE FOR BETWEEN-CLUSTERS MATRIX           RRRO1410
C          NISVT, SVCENT, IDATAT:  TEMPORARY STORAGE SERVING THE       RRRO1420
C                SAME FUNCTIONS AS NISV, SVCEN, AND IDATA              RRRO1430
```

```
C           VEC = EIGENVECTORS                                          RRR0144C
C          EIG = EIGENVALUES
C                                                                       RRR0146C
C                                                                       RRR01470
  1CC CALL PRELIM (CRIT)                                                RRRC148C
C     IFINE IS ESCAPE PARAMETER                                         RRR01490
      IF (IFINE.EC.1) GC TO 400                                         RRR0150C
      IF (IFINE.EC.2) GC TC 500                                         RRR01510
      CALL RANDST (CRIT)                                                RRR01520
      IF (IFINE.EC.1) GC TO 400                                         RRR01530
  3CC CALL KMEANS (CRIT)                                                RRR01540
      IF (IFINE.EC.1) GC TO 400                                         RRRC1550
      CALL ISWTCH (CRIT)                                                RRR01560
      GC TO 100                                                         RRRC157C
C     IFINE GOT SET TO 1:  IF DATA HAS EEEN READ IN, RESET CATA ANC     RRR01580
C     GC TC NEXT FROBLEM                                                RRRC159C
  4CC CC 410 I=1,NOBS                                                   RRR01600
      CC 41C J=1,NVARS                                                  RRRC161C
      IF (NCSTAN.EQ.1) CATA(I,J) = DATA(I,J) * SC(J)
  41C CATA(I,J) = DATA(I,J) + VMEAN(J)                                  RRR0163C
      GC TO 100                                                         RRR01640
C     ALL DCNE:  WRITE CLT LAST MESSAGE AND EXIT                        RRR0165C
  5CC WRITE (6,9OC)                                                     RRR01660
  9CC FCRMAT ('0 END OF ANALYSES.')                                     RRRC1670
      STOP                                                              RRR0168C
      ENC                                                               RRRC169C
      SUBROUTINE FRELIM (CRIT)                                          RRRC17CC
C                                                                       RRR01710
C     THIS SUBROUTINE MAKES THE PRELIMINARY CALCULATIONS.               RRR01720
C     IT INPUTS THE DATA, CALCULATES THE MEANS ANC VARIANCES FOR EACH   RRR01730
C     VARIABLE, STANDARCIZES (CONVERTS TC Z-SCCRES) EACH VARIABLE IF    RRRC1740
C     RECUESTEC, AND CALCULATES THE CROSS-PRODUCTS MATRIX               RRR01750
C                                                                       RRR0176C
      CCMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,ICENT(6C0), RRR01770
     1DATA(6C0,20),T(20,20),B(20,20),W(20,20),WFCT(2C,20),SVCEN(2C,2C), RRRC1780
     2ICATA(600),NISV(2C),VMEAN(20),SC(2C),XVEC(2C),YVEC(20),ET(20,20), RRRC179C
     3 NISVT(20),SVCENT(20,20),IDATAT(6CC),VEC(20,20),EIG(2C)           RRR01800
      CIMENSICN TITLE(20),IFMT(20),VAR(2C)                              RRRC181C
C                                                                       RRR0182C
C     INFUT SECTICN:  REAC IN TITLE CARC, PROBLEM CARD, CPTIONAL CARCS, RRR01830
C     FCRMAT CARD, AND CATA CARCS                                       RRRC184C
C     WRITE OUT SCLUTION SPECIFICATIONS                                 RRR01850
C                                                                       RRRC186C
      IFINE = 0                                                         RRRC187C
      REAC (5,900) (TITLE(I),I=1,20)                                    RRR01880
      WRITE (6,903) (TITLE(I),I=1,20)                                   RRR01890
      READ (5,901) NOBS,NVARS,NGPS,ICRIT,NOSTAN,ICIST,NCCATA,KTIME      RRR01900
      IF (NOBS.EQ.0) GC TC 820                                          RRR01910
      IF (NOBS.GT.600) GO TO 815                                        RRR01920
      IF (NVARS.GT.20) GO TO 815                                        RRR01930
      IF (NGPS.GT.20) GO TO 815                                         RRR01940
      IF (ICRIT.GT.4) GO TO 815                                         RRR01950
      IF (NOSTAN.GT.1) GO TO 815                                        RRR01960
      IF (IDIST.GT.2) GO TO 815                                         RRR01970
      IF (NVARS.EQ.1) ICRIT=1                                           RRR01980
      IF (NVARS.EQ.1) IDIST=0                                           RRR01990
    4 IF (NODATA.EQ.1) GO TO 5                                          RRR02000
      READ (5,902) (IFMT(I),I=1,18)
    5 WRITE (6,904) NOBS,NVARS,NGPS                                     RRR02020
      GO TO (10,20,30,40),ICRIT                                         RRR02030
   10 WRITE (6,920)                                                     RRR02040
      GO TO 50                                                          RRR02050
   20 WRITE (6,921)                                                     RRR02060
      GO TO 50                                                          RRR02070
   30 WRITE (6,922)                                                     RRR02080
      GO TO 50                                                          RRR02090
   40 WRITE (6,923)                                                     RRR02100
   50 CONTINUE                                                          RRR02110
      IF (NOSTAN) 60,60,70                                              RRR02120
   60 WRITE (6,924)                                                     RRR02130
      GO TO 80                                                          RRR02140
   70 WRITE (6,925)                                                     RRR02150
   80 CONTINUE                                                          RRR02160
      IF (IDIST) 85,85,90                                               RRR02170
   85 WRITE (6,926)                                                     RRR02180
      GO TO 95                                                          RRR02190
   90 IF (IDIST.EQ.2) GO TO 92                                          RRR02200
      WRITE (6,936)                                                     RRR02210
      GO TO 95                                                          RRR02220
   92 WRITE (6,927)                                                     RRR02230
   95 CONTINUE                                                          RRR02240
      IF (NODATA.EQ.1) GO TO 101                                        RRR02250
      WRITE (6,905) (IFMT(I),I=1,18)
      WRITE (6,980)
      DO 100 I = 1,NOBS                                                 RRR02270
      READ (5,IFMT) IDENT(I),(DATA(I,J),J=1,NVARS)
      WRITE (6,950) IDENT(I),(DATA(I,J),J=1,NVARS)
  100 CONTINUE                                                          RRR02290
      NODATA = 1                                                        RRR02300
      GO TO 102                                                         RRR02310
  101 WRITE (6,937)                                                     RRR02320
  102 WRITE (6,943) KTIME                                               RRR02330
C                                                                       RRR02340
C     CALCULATE VARIABLE MEANS AND VARIANCES                            RRR02350
C     SUBTRACT OUT OVERALL MEAN                                         RRR02360
C                                                                       RRR02370
```

```
      DO 105 J=1,NVARS                                                     RRR02380
      VMEAN (J) = 0.0                                                      RRR02390
  105 VAR (J) = 0.0                                                        RRR02400
      DO 110 I = 1,NOBS                                                    RRR02410
      DO 110 J = 1,NVARS                                                   RRR02420
  110 VMEAN (J) = VMEAN (J) + DATA(I,J)/NOBS                               RRR02430
      DO 120 I=1,NOBS                                                      RRR02440
      DO 120 J=1,NVARS                                                     RRR02450
      DATA(I,J) = DATA(I,J)-VMEAN(J)                                       RRR02460
  120 VAR(J) = VAR(J) + (DATA(I,J)**2)/(NOBS-1)                            RRR02470
      DO 125 J = 1,NVARS                                                   RRR02480
      IF (VAR(J).LE.0.000001) MFLAG=1                                      RRR02490
  125 SD(J) = SQRT (VAR(J))                                                RRR02500
      IF (MFLAG.EQ.1) GO TO 825                                            RRR02510
C                                                                          RRR02520
C    CALCULATE T = X'X, THE CROSS-PRODUCTS MATRIX                          RRR02530
C                                                                          RRR02540
  131 DO 135 K=1,NVARS                                                     RRR02550
      DO 135 J=K,NVARS                                                     RRR02560
  135 T(K,J) = 0.0                                                         RRR02570
      DO 140 K = 1,NVARS                                                   RRR02580
      DO 140 J=K,NVARS                                                     RRR02590
      DO 140 I = 1,NOBS                                                    RRR02600
  140 T(K,J) = T(K,J) + DATA(I,K)*DATA(I,J)                                RRR02610
C                                                                          RRR02620
C    OUTPUT SECTION: WRITE OUT MEANS, STANDARD DEVIATIONS, AND CROSS-     RRR02630
C                        PRODUCTS MATRIX T                                 RRR02640
C                                                                          RRR02650
      WRITE (6,908)                                                        RRR02660
      WRITE (6,907) (VMEAN(J),J=1,NVARS)                                   RRR02670
      WRITE (6,909)                                                        RRR02680
      WRITE (6,907) (SD(J),J=1,NVARS)                                      RRR02690
      WRITE (6,911)                                                        RRR02700
      DO 810 I = 1,NVARS                                                   RRR02710
      WRITE (6,995) (T(J,I),J=1,I)
  810 CONTINUE                                                             RRR02730
      IF (NOSTAN.EQ.0) GO TO 180                                           RRR02740
C                                                                          RRR02750
C    STANDARDIZE IF REQUESTED                                              RRR02760
C                                                                          RRR02770
      DO 160 J=1,NVARS                                                     RRR02780
      DO 150 I=1,NOBS                                                      RRR02790
  150 DATA(I,J) = DATA(I,J) / SD(J)                                        RRR02800
      DO 160 K=J,NVARS                                                     RRR02810
  160 T(J,K) = (1.0/SD(J))*T(J,K)*(1.0/SD(K))                              RRR02820
      WRITE (6,702)
      DO 700 I=1,NOBS
      WRITE (6,701) (DATA(I,L),L=1,NVARS)
```

```
700 CONTINUE
701 FORMAT ('0',10(F11.3,1X))
702 FORMAT ('1','STANDARDIZED DATA MATRIX IS')
    WRITE (6,912)
    DO 170 I=1,NVARS
    WRITE (6,701) (T(J,I),J=1,I)
170 CONTINUE
912 FORMAT('0 THE STANDARIZED CROSS-PROD MATRIX IS ')
180 RETURN                                                         RRR02830
815 IFINE = 1                                                      RRR02840
    IF (NODATA.EQ.0) IFINE=2                                       RRR02850
    WRITE (6,935)                                                  RRR02860
    RETURN                                                         RRR02870
820 IFINE = 2                                                      RRR02880
    RETURN                                                         RRR02890
825 WRITE (6,945) J                                                RRR02900
    IFINE = 1                                                      RRR02910
    RETURN                                                         RRR02920
C                                                                  RRR02930
C       FORMAT STATEMENTS                                          RRR02940
C                                                                  RRR02950
900 FORMAT (20A4)                                                  RRR02960
901 FORMAT (I4,2I2,5I1)                                            RRR02970
902 FORMAT (18A4)                                                  RRR02980
903 FORMAT ('1',20A4)                                              RRR02990
904 FORMAT ('0 THE NUMBER OF OBSERVATIONS IS ',I4,/,'  THE NUMBER OF VRRR03000
   1ARIABLES IS ',I2,/,'  THE NUMBER OF GROUPS (INITIAL) IS ',I2)  RRR03010
905 FORMAT ('0 THE INPUT FORMAT IS ',18A4)                         RRR03020
907 FORMAT ('0',5(F11.3,1X),/,1X,5(F11.3,1X),///)
908 FORMAT ('1','THE VARIABLE MEANS ARE')
909 FORMAT ('0THE VARIABLE STANDARD DEVIATIONS ARE')               RRR03050
911 FORMAT ('0THE CROSS-PRODUCTS MATRIX IS ')                      RRR03060
920 FORMAT ('0 THE CRITERION IS TO MINIMIZE TRACE W')              RRR03070
921 FORMAT ('0 THE CRITERION IS TO MINIMIZE DETERMINANT W')        RRR03080
922 FORMAT ('0 THE CRITERION IS TO MAXIMIZE THE LARGEST ROOT OF W-1B')RRR03090
923 FORMAT ('0 THE CRITERION IS TO MINIMIZE THE TRACE OF W-1B')    RRR03100
924 FORMAT ('0 THE VARIABLES WILL NOT BE STANDARDIZED')            RRR03110
925 FORMAT ('0 THE VARIABLES WILL BE STANDARDIZED')                RRR03120
926 FORMAT ('0 EUCLIDIAN DISTANCE WILL BE USED')                   RRR03130
927 FORMAT ('0 MAHALANOBIS DISTANCE WILL BE USED')                 RRR03140
935 FORMAT ('1 ERROR IN PROBLEM CARD.  SKIPPING PROBLEM.')         RRR03150
936 FORMAT ('0 WEIGHTED EUCLIDIAN DISTANCE WILL BE USED.')         RRR03160
937 FORMAT ('0PREVIOUS DATA BEING USED ')                          RRR03170
943 FORMAT ('1','THE TIMING PARAMETER FOR INDIVIDUAL SWITCHES IS ',I1)
945 FORMAT ('0 WARNING:  VARIABLE ',I2,' HAS ZERO VARIANCE.')      RRR03190
950 FORMAT ('0',I9.5X,10F8.3)
980 FORMAT ('1','INITIAL DATA MATRX ',//)
999 FORMAT('0',10(F10.2,1X))
```

```
      END                                                               RRR03200
      SUBROUTINE RANDST (CRIT)                                          RRR03210
C                                                                       RRR03220
C     THIS SUBROUTINE DETERMINES THE INITIAL CLUSTER CONFIGURATION.     RRR03230
C                                                                       RRR03240
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,ICENT(600), RRR03250
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20), RRR03260
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),ET(20,20), RRR03270
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20)           RRR03280
      DIMENSION ISTRT(20)                                               RRR03290
C                                                                       RRR03300
C     ONE PASS K-MEANS TO DETERMINE INITIAL CLUSTER SOLUTION            RRR03310
C                                                                       RRR03320
C     RANDOM NUMBER STARTER                                             RRR03330
  200 IX=15739                                                          RRR03340
      NGFST=NGPS                                                        RRR03350
C    DO 260: GETTING THREE INITIAL CONFIGURATIONS; THE ONE WITH THE     RRR03360
C    BEST CRITERION VALUE WILL BE USED                                  RRR03370
      DO 260 IBSRT = 1,3                                                RRR03380
      DO 205 M = 1,NGPS                                                 RRR03390
      NISVT(M)=1                                                        RRR03400
C    FIND A RANDOM OBSERVATION TO SERVE AS INITIAL CLUSTER CENTER       RRR03410
C    FOR EACH CLUSTER                                                   RRR03420
  201 RAND = URAND(IX)                                                  RRR03430
      ISTRT(M) = RAND * NOBS                                            RRR03440
      IF (ISTRT(M).EQ.0) ISTRT(M)=1                                     RRR03450
      IF (M.EQ.1) GO TO 203                                             RRR03460
      MM1 = M-1                                                         RRR03470
      DO 204 MM=1,MM1                                                   RRR03480
      IF (ISTRT(M).EQ.ISTRT(MM)) GO TO 201                              RRR03490
  204 CONTINUE                                                          RRR03500
  203 I = ISTRT(M)                                                      RRR03510
      DO 205 J=1,NVARS                                                  RRR03520
  205 SVCENT(M,J) = DATA(I,J)                                           RRR03530
      ITEMP = IDIST                                                     RRR03540
      IDIST = 0                                                         RRR03550
C    DO FOR EACH OBSERVATION                                            RRR03560
      DO 225 I=1,NOBS                                                   RRR03570
C    FIND EUCLIDIAN DISTANCE TO EACH CLUSTER CENTER                     RRR03580
      DO 215 M=1,NGPS                                                   RRR03590
      DO 210 J=1,NVARS                                                  RRR03600
      XVEC(J)=DATA(I,J)                                                 RRR03610
  210 YVEC(J) = SVCENT(M,J)                                             RRR03620
      CALL DISTCE (XVEC,YVEC,DISTA)                                     RRR03630
      IF (IFINE.EQ.1) RETURN                                            RRR03640
      IF (M.EQ.1) GO TO 212                                             RRR03650
      IF (DISTA.GE.SMDIST) GO TO 215                                    RRR03660
  212 SMDIST = DISTA                                                    RRR03670
```

AD-A086 521 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 12/1
CLASSIFICATION TECHNIQUES FOR MULTIVARIATE DATA ANALYSIS.(U)
MAR 80 J K LEE

UNCLASSIFIED NL

2 of 2
AD
A0865[illegible]

END
DATE
FILMED
8-80
DTIC

```
      IDATAT(I) = M                                                     RRR03680
  215 CONTINUE                                                          RRR03690
C     ASSIGN OBSERVATION TO CLOSEST CLUSTER AND UPDATE THAT CLUSTER     RRR03700
C     CENTER                                                            RRR03710
      K = IDATAT(I)                                                     RRR03720
      NISVT(K) = NISVT(K) + 1                                           RRR03730
      DO 225 J = 1,NVARS                                                RRR03740
  225 SVCENT(K,J) = ((NISVT(K)-1)*SVCENT(K,J)+DATA(I,J))/NISVT(K)       RRR03750
      IDIST = ITEMP                                                     RRR03760
C     END OF DO FOR EACH OBSERVATION                                    RRR03770
C     RECALCULATE CLUSTER CENTERS TO ELIMINATE INITIAL RANDOM OBSERVA-  RRR03780
C     TIONS                                                             RRR03790
      DO 230 M=1,NGPS                                                   RRR03800
      NISVT(M) = NISVT(M)-1                                             RRR03810
      DO 230 J=1,NVARS                                                  RRR03820
  230 SVCENT(M,J)=0.                                                    RRR03830
      DO 235 I=1,NOBS                                                   RRR03840
      M = IDATAT(I)                                                     RRR03850
      DO 235 J=1,NVARS                                                  RRR03860
  235 SVCENT(M,J) = SVCENT(M,J) + DATA(I,J)/NISVT(M)                    RRR03870
C                                                                       RRR03880
C      CALCULATE THE B AND W MATRICES                                   RRR03890
C      CALCULATE THE CRITERION VALUE                                    RRR03900
C                                                                       RRR03910
      IDIR=2                                                            RRR03920
      IF (ICRIT.EQ.1) IDIR=1                                            RRR03930
      CALL WCALC (SVCENT,NISVT,NGPST,IDIR)                              RRR03940
      IF (IFINE.EQ.1) RETURN                                            RRR03950
      CALL CRITON (CRIT)                                                RRR03960
      IF (IFINE.EQ.1) RETURN                                            RRR03970
C     WHICH INITIAL CONFIGURATION                                       RRR03980
      IF (IBSRT.GT.1) GO TO 250                                         RRR03990
C     FIRST INITIAL CONFIGURATION:  BCRIT IS THE BEST CRITERION         RRR04000
  245 BCRIT = CRIT                                                      RRR04010
C     SAVE CLUSTER SIZES (NISV), CLUSTER CENTERS (SVCEN), CLUSTER       RRR04020
C     LISTS (LSTSV), AND OBSERVATION ID'S (IDATA)                       RRR04030
      DO 247 M = 1,NGPS                                                 RRR04040
      NISV(M) = NISVT(M)                                                RRR04050
      DO 247 L = 1,NVARS                                                RRR04060
  247 SVCEN(M,L) = SVCENT(M,L)                                          RRR04070
      DO 249 I=1,NOBS                                                   RRR04080
  249 IDATA(I) = IDATAT(I)                                              RRR04090
      GO TO 260                                                         RRR04100
C     SECOND OR THIRD INITIAL CONFIGURATION                             RRR04110
  250 IF (CRIT.GE.BCRIT) GO TO 260                                      RRR04120
  255 BCRIT = CRIT                                                      RRR04130
      DO 257 M = 1,NGPS                                                 RRR04140
      NISV(M) = NISVT(M)                                                RRR04150
      DO 257 L = 1,NVARS                                                RRR04160
  257 SVCEN(M,L) = SVCENT(M,L)                                          RRR04170
      DO 259 I=1,NOBS                                                   RRR04180
  259 IDATA(I) = IDATAT(I)                                              RRR04190
  260 CONTINUE                                                          RRR04200
C     ALL DONE                                                          RRR04210
      RETURN                                                            RRR04220
      END                                                               RRR04230
      SUBROUTINE KMEANS (CRIT)                                          RRR04240
C                                                                       RRR04250
C         K-MEANS.  EACH OBSERVATION IS ASSIGNED TO THE CLOSEST CLUSTER RRR04260
C         CENTER.                                                       RRR04270
C                                                                       RRR04280
C                                                                       RRR04290
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600), RRR04300
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20), RRR04310
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),B1(20,20), RRR04320
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20)           RRR04330
C     INITIALIZING FLAGS                                                RRR04340
C     AND TEMPORARY STORAGE FOR NISV, SVCEN                             RRR04350
  200 NGPST = NGPS                                                      RRR04360
      JFLAG = 0                                                         RRR04370
      ECRIT = CRIT                                                      RRR04380
      DO 201 M=1,NGPS                                                   RRR04390
      NISVT(M) = NISV(M)                                                RRR04400
      DO 201 J=1,NVARS                                                  RRR04410
  201 SVCENT(M,J) = SVCEN(M,J)                                          RRR04420
C     RECALCULATING WITHIN-CLUSTERS MATRIX                              RRR04430
      ICIR=1                                                            RRR04440
      IF (IDIST.EQ.2) ICIR=2                                            RRR04450
      CALL WCALC (SVCENT,NISVT,NGPST,ICIR)                              RRR04460
      IF (IFINE.EQ.1) RETURN                                            RRR04470
C     INITIALIZING TEMPORARY STORAGE FOR LSTSV, IDATA                   RRR04480
      DO 210 I = 1,NOBS                                                 RRR04490
  210 IDATAT(I) = IDATA(I)                                              RRR04500
C     MAJOR DO LOOP:  DO FOR EACH OBSERVATION                           RRR04510
      DO 400 I = 1,NOBS                                                 RRR04520
C                                                                       RRR04530
C      BEGIN K-MEANS SECTION                                            RRR04540
C                                                                       RRR04550
C     CALCULATE VECTOR OF DISTANCES FROM OBSERVATION TO EACH CLUSTER    RRR04560
C     CENTER                                                            RRR04570
  325 DO 335 M=1,NGPST                                                  RRR04580
      DO 330 J=1,NVARS                                                  RRR04590
      XVEC (J) = DATA (I,J)                                             RRR04600
  330 YVEC (J) = SVCENT (M,J)                                           RRR04610
      CALL DISTCE (XVEC,YVEC,DISTB)                                     RRR04620
      IF (IFINE.EQ.1) RETURN                                            RRR04630
      IF (M.EQ.1) GO TO 332                                             RRR04640
      IF (DISTB.GE.SMDIST) GO TO 335                                    RRR04650
  332 SMDIST = DISTB                                                    RRR04660
      IDGP = M                                                          RRR04670
  335 CONTINUE                                                          RRR04680
C     IS OBSERVATION ALREADY IN THE CLOSEST CLUSTER? IF YES, SKIP THIS  RRR04690
C     SECTION                                                           RRR04700
      IF (IDGP.EQ.IDATAT(I)) GO TO 360                                  RRR04710
C     IOLD IS OLD CLUSTER ASSIGNMENT                                    RRR04720
  346 IOLD = IDATAT(I)                                                  RRR04730
C     INEW IS NEW CLUSTER ASSIGNMENT                                    RRR04740
      INEW = IDGP                                                       RRR04750
      JFLAG = 1                                                         RRR04760
C     RECALCULATE CLUSTER CENTERS                                       RRR04770
      DO 348 J = 1,NVARS                                                RRR04780
      SVCENT(IOLD,J) = (NISVT(IOLD)*SVCENT(IOLD,J)-DATA(I,J))/          RRR04790
     1  (NISVT(IOLD)-1)                                                 RRR04800
  348 SVCENT(INEW,J) = (NISVT(INEW)*SVCENT(INEW,J)+DATA(I,J))/          RRR04810
     1  (NISVT(INEW)+1)                                                 RRR04820
C     ADJUST NISV                                                       RRR04830
      NISVT(IOLD) = NISVT(IOLD)-1                                       RRR04840
      NISVT(INEW) = NISVT(INEW)+1                                       RRR04850
C     ADJUST IDATA                                                      RRR04860
      IDATAT(I) = IDGP                                                  RRR04870
C     RECALCULATE WITHIN-CLUSTERS MATRIX FOR USE IN COMPUTING SCALED    RRR04880
C     EUCLIDIAN AND MAHALANOBIS DISTANCE                                RRR04890
      IF (IDIST.EQ.0) GO TO 360                                         RRR04900
      CALL WCALC (SVCENT,NISVT,NGPST,ICIR)                              RRR04910
      IF (IFINE.EQ.1) RETURN                                            RRR04920
  360 CONTINUE                                                          RRR04930
C     DONE WITH MAJOR DO LOOP                                           RRR04940
  400 CONTINUE                                                          RRR04950
C                                                                       RRR04960
C       CALCULATE THE CRITERION                                         RRR04970
C                                                                       RRR04980
C     RECALCULATE SVCEN:  ACCURACY MEASURE                              RRR04990
      DO 401 M=1,NGPST                                                  RRR05000
      DO 401 J=1,NVARS                                                  RRR05010
  401 SVCENT(M,J) = 0.                                                  RRR05020
      DO 402 I=1,NOBS                                                   RRR05030
      M = IDATAT(I)                                                     RRR05040
      DO 402 J=1,NVARS                                                  RRR05050
  402 SVCENT(M,J) = SVCENT(M,J) + DATA(I,J)/NISVT(M)                    RRR05060
C     RECALCULATE WITHIN-CLUSTERS MATRIX AND CRITERION VALUE BASED ON   RRR05070
C     NEW CLUSTER CENTER VALUES                                         RRR05080
      ICIR = 2                                                          RRR05090
      IF (ICRIT.EQ.1) ICIR = 1                                          RRR05100
      CALL WCALC (SVCENT,NISVT,NGPST,ICIR)                              RRR05110
      IF (IFINE.EQ.1) RETURN
  405 CALL CRITCN (CRIT)
      IF (IFINE.EQ.1) RETURN
C     IF CRITERION BETTER THAN BEFORE?  IF YES, THEN ANOTHER ITERATION;
C     IF NO, FINISH KMEANS
      IF (CRIT.GE.BCRIT) GO TO 535
C
C     ANOTHER ITERATION
C
C     PUT TEMPORARY VALUES INTO PERMANENT LOCATIONS
  515 NGPS=NGPST
      DO 520 M = 1,NGPS
      NISV(M) = NISVT (M)
      DO 520 J = 1,NVARS
  520 SVCEN (M,J) = SVCENT (M,J)
      DO 530 I = 1,NOBS
  530 IDATA(I) = IDATAT(I)
      GO TO 200
C
C     FINISH
C
C
C     JFLAG = 0 MEANS NO CHANGES HAVE BEEN MADE DURING THE LAST
C     ITERATION:  ITERATIONS CONVERGED
  535 IF (JFLAG.EQ.0) RETURN
C     JFLAG = 1 MEANS CHANGES HAVE BEEN MADE BUT THE CRITERION VALUE
C     GOT WORSE:  ITERATIONS NOT CONVERGING
  540 WRITE (6,940)
      WRITE (6,942) CRIT
      WRITE (6,943) BCRIT
C     RECALCULATE WITHIN-CLUSTERS MATRIX AND RESET CRIT
      ICIR=2
      IF (ICRIT.EQ.1) ICIR=1
      CALL WCALC (SVCEN,NISV,NGPS,IDIR)
      IF (IFINE.EQ.1) RETURN
      CRIT=BCRIT
      RETURN
  940 FORMAT ('0 ITERATIONS NOT CONVERGING')
  942 FORMAT ('0 THE CRITERION VALUE IS ',E12.6)
  943 FORMAT ('0 THE BEST CRITERION VALUE IS ',E12.6)
      END
      SUBROUTINE ISWTCH (CRIT)
C
C     THIS SUBROUTINE CONSIDERS SWITCHING EACH OBSERVATION TO A
C     DIFFERENT CLUSTER.  THE SWITCH IS MADE IFF A BETTER CRITERION
C     VALUE RESULTS.
C     THIS SUBROUTINE ALSO DEPENDS ON THE PARAMETER KTIME:  IT DETERMINES
C     WHICH OBSERVATIONS ARE TO BE CONSIDERED.  FOR COMPLETE EXPLANATION
C     OF THE KTIME PARAMETER, SEE THE PROGRAM DESCRIPTION.
```

```
C
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600),
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20),
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),BT(20,20),
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20)
      DIMENSION JFLAG (20)
C    SET UP THE IDIR FLAG
      IDIR=2
      IF (ICRIT.EQ.1) IDIR=1
C    KTIME = 0 MEANS SKIP THIS HEURISTIC
      IF (KTIME.EQ.0) GO TO 750
C    SET UP TEMPORARY STORAGE AREAS FOR NGPS, NISV, AND SVCEN
      NGPST=NGPS
      DO 500 M = 1,NGPS
      NISVT(M)=NISV(M)
      JFLAG(M)=0
      DO 500 J = 1,NVARS
  500 SVCENT(M,J)=SVCEN(M,J)
C    SET IFLAG = 0:  WILL BE SET = 1 IF ANY SWITCH IS MADE
  600 IFLAG=0
C    MAJOR DO LOOP:  DO 700
      DO 700 M=1,NGPS
C    JFLAG (M) INDICATES WHETHER CLUSTER M HAS BEEN ALTERED
      IF (JFLAG(M).EQ.2) GO TO 700
C    KTIME = 9 MEANS ALL OBSERVATIONS WILL BE CONSIDERED
      IF (KTIME.EQ.9) GO TO 617
C    COMPUTE ALL DISTANCES FROM OBSERVATIONS TO CLUSTER CENTERS FOR THE
C    OBSERVATIONS IN CLUSTER M
      DO 605 J=1,NVARS
  605 YVEC(J) = SVCEN(M,J)
      BIGDIS = 0.
      DO 615 I=1,NOBS
      IF (IDATA(I).NE.M) GO TO 615
      DO 610 J=1,NVARS
  610 XVEC(J) = DATA(I,J)
      CALL DISTCE (XVEC,YVEC,DISTA)
C    LOCATE OBSERVATION WITH BIGGEST DISTANCE
      IF (BIGDIS.GE.DISTA) GO TO 615
      BIGDIS = DISTA
  615 CONTINUE
C    DIVIDE BIGDIS BY TIMING PARAMETER KTIME
      TIME = KTIME
      PSME = BIGDIS / TIME
  617 MOLD = M
C    DO FOR ALL OBSERVATIONS IN CLUSTER M
      DO 691 I=1,NOBS
      IF (IDATA(I).NE.M) GO TO 691
      KFLAG = 0
```

```
C     KTIME = 9 MEANS CONSIDER ALL OBSERVATIONS                          RRR06080
        IF (KTIME.EQ.9) GO TO 618                                        RRR06090
C     DO NOT  CONSIDER SWITCHING THE OBSERVATION IF THE DISTANCE TO ITS  RRR06100
C     CURRENT CLUSTER CENTER IS LT BIGDIS / KTIME                        RRR06110
        DO 619 J=1,NVARS                                                 RRR06120
  619 XVEC(J) = DATA(I,J)                                                RRR06130
        CALL DISTCE (XVEC,YVEC,DISTA)                                    RRR06140
        IF (DISTA.LT.PSME) GO TO 691                                     RRR06150
  618 CONTINUE                                                           RRR06160
        DO 690 MNEW=1,NGPS                                               RRR06170
        IF (NISV(MOLD).EQ.1) GO TO 690                                   RRR06180
        IF (MNEW.EQ.MOLD) GO TO 690                                      RRR06190
        IF (KFLAG.EQ.1) GO TO 690                                        RRR06200
C     COMPUTE NEW SVCEN BASED ON OBSERVATION BEING SWITCHED              RRR06210
        DO 620 J=1,NVARS                                                 RRR06220
        SVCENT(MNEW,J) =  (NISV(MNEW)*SVCEN(MNEW,J)+DATA(I,J))/          RRR06230
      1   (NISV(MNEW)+1)                                                 RRR06240
        SVCENT(MOLD,J) =  (NISV(MOLD)*SVCEN(MOLD,J)-DATA(I,J))/          RRR06250
      1  (NISV(MOLD)-1)                                                  RRR06260
  620 CONTINUE                                                           RRR06270
C     ADJUST NISV                                                        RRR06280
        NISVT(MNEW) = NISV(MNEW)+1                                       RRR06290
        NISVT(MOLD) = NISV(MOLD)-1                                       RRR06300
C     COMPUTE WITHIN-CLUSTERS MATRIX AND NEW CRITERION VALUE             RRR06310
        CALL WCALC (SVCENT,NISVT,NGPST,IDIR)                             RRR06320
        IF (IFINE.EQ.1) RETURN                                           RRR06330
        CALL CRITON (TCRIT)                                              RRR06340
        IF (IFINE.EQ.1) RETURN                                           RRR06350
C     MAKE THE SWITCH IFF NEW CRIT IS BETTER                             RRR06360
        IF (TCRIT.GE.CRIT) GO TO 680                                     RRR06370
C                                                                        RRR06380
C       MAKE THE SWITCH                                                  RRR06390
C                                                                        RRR06400
C     RESET FLAGS                                                        RRR06410
        JFLAG(M) = 1                                                     RRR06420
        JFLAG(MNEW) = 1                                                  RRR06430
        IFLAG=1                                                          RRR06440
        KFLAG = 1                                                        RRR06450
C     ADJUST CRIT, IDATA, NISV, SVCEN                                    RRR06460
        CRIT=TCRIT                                                       RRR06470
        IDATA(I) = MNEW                                                  RRR06480
        NISV(MNEW) = NISVT(MNEW)                                         RRR06490
        NISV(MOLD) = NISVT(MOLD)                                         RRR06500
  629 DO 630 J=1,NVARS                                                   RRR06510
        SVCEN(MNEW,J) = SVCENT(MNEW,J)                                   RRR06520
  630 SVCEN(MOLD,J) = SVCENT(MOLD,J)                                     RRR06530
        GO TO 690                                                        RRR06540
C     SWITCH WAS NOT MADE; RESET NISVT, SVCENT, TO VALUES PRESENT        RRR06550
```

```
C     BEFORE THE SWITCH WAS CONSIDERED                                  RRR06560
  680 NISVT(MNEW) = NISV(MNEW)                                          RRR06570
      NISVT(MOLD) = NISV(MOLD)                                          RRR06580
      DO 685 J=1,NVARS                                                  RRR06590
      SVCENT(MNEW,J) = SVCEN(MNEW,J)                                    RRR06600
  685 SVCENT(MOLD,J) = SVCEN(MOLD,J)                                    RRR06610
  690 CONTINUE                                                          RRR06620
  691 CONTINUE                                                          RRR06630
C     FINISH WITH CLUSTER M:  IF NO SWITCHES HAVE BEEN MADE, SET        RRR06640
C     JFLAG(M) = 2 AND GO TO NEXT CLUSTER;  IF SWITCHES HAVE BEEN MADE, RRR06650
C     ADJUST LSTSV AND SET JFLAG(M) = 0 AND GO TO NEXT CLUSTER          RRR06660
  695 IF (JFLAG(M).EQ.0) GO TO 699                                      RRR06670
      JFLAG(M) = 0                                                      RRR06680
      GO TO 700                                                         RRR06690
  699 JFLAG(M) = 2                                                      RRR06700
  700 CONTINUE                                                          RRR06710
C     DONE WITH ALL CLUSTERS:  IFLAG = 1 MEANS SOME SWITCHES HAVE BEEN  RRR06720
C     MADE; GO BACK AND ITERATE                                         RRR06730
      IF (IFLAG.EQ.1) GO TO 600                                         RRR06740
C                                                                       RRR06750
C     ALL DONE; ACCURATELY CALCULATE MEANS AND CRITERION AND OUTPUT     RRR06760
C     THE RESULTS                                                       RRR06770
C                                                                       RRR06780
  750 WRITE (6,900)                                                     RRR06790
  900 FORMAT ('1','THE FINAL CLUSTER SOLUTION IS ')
C     RECALCULATE CLUSTER CENTERS, WITHIN-CLUSTERS MATRIX, AND CRITERION RRR06810
C     VALUE                                                             RRR06820
      DO 715 M=1,NGPS                                                   RRR06830
      DO 715 J=1,NVARS                                                  RRR06840
  715 SVCEN(M,J) = 0.                                                   RRR06850
      DO 720 I=1,NOBS                                                   RRR06860
      M = IDATA(I)                                                      RRR06870
      DO 720 J=1,NVARS                                                  RRR06880
  720 SVCEN(M,J) = SVCEN(M,J) + DATA(I,J)/NISV(M)                       RRR06890
      CALL WCALC (SVCEN,NISV,NGPS,IDIR)                                 RRR06900
      IF (IFINE.EQ.1) RETURN                                            RRR06910
      CALL CRITON (CRIT)                                                RRR06920
      IF (IFINE.EQ.1) RETURN                                            RRR06930
C     CALL THE OUTPUT ROUTINE                                           RRR06940
      CALL OUTPUT (CRIT)                                                RRR06950
      WRITE (6,901)                                                     RRR06960
  901 FORMAT ('0 END OF CLUSTER PROBLEM')                               RRR06970
      RETURN                                                            RRR06980
      END                                                               RRR06990
      SUBROUTINE OUTPUT (CRIT)                                          RRR07000
C                                                                       RRR07010
C     THIS SUBROUTINE PRINTS OUT THE CLUSTER SOLUTION                   RRR07020
C          FOR EACH CLUSTER - THE CLUSTER SIZE                          RRR07030
```

```
C                                THE CLUSTER CENTROID                        RRR07040
C                                THE OBSERVATIONS BELONGING TO THE CLUSTER   RRR07050
C          THE WITHIN CELLS MATRIX                                           RRR07060
C          THE CRITERION VALUE                                               RRR07070
C                                                                            RRR07080
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600),      RRR07090
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20),      RRR07100
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),ET(20,20),      RRR07110
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20)                RRR07120
C     UNSTANDARDIZE IF NECESSARY                                             RRR07130
      IF (NOSTAN.EQ.0) GO TO 40                                              RRR07140
      DO 20 M=1,NGPS                                                         RRR07150
      DO 20 J=1,NVARS                                                        RRR07160
   20 SVCEN(M,J) = SVCEN(M,J)*SD(J)                                          RRR07170
      DO 30 J=1,NVARS                                                        RRR07180
      DO 30 K=J,NVARS                                                        RRR07190
   30 T(J,K) = SD(J)*T(J,K)*SD(K)                                            RRR07200
      IDIR = 2                                                               RRR07210
      IF (ICRIT.EQ.1) IDIR=1                                                 RRR07220
      CALL WCALC (SVCEN,NISV,NGPS,IDIR)                                      RRR07230
      IF (IFINE.EQ.1) RETURN                                                 RRR07240
      CALL CRITON (CRIT)                                                     RRR07250
      IF (IFINE.EQ.1) RETURN                                                 RRR07260
C     ADD THE OVERALL MEAN BACK INTO EACH OBSERVATION; UNSTANDARDIZE         RRR07270
C      THE DATA IF NECESSARY                                                 RRR07280
   40 DO 80 J=1,NVARS                                                        RRR07290
      DO 50 I=1,NOBS                                                         RRR07300
      IF (NOSTAN.EQ.1) DATA(I,J) = DATA(I,J)*SD(J)                           RRR07310
   50 DATA(I,J) = DATA(I,J) + VMEAN(J)                                       RRR07320
      DO 60 M=1,NGPS                                                         RRR07330
   60 SVCEN(M,J) = SVCEN(M,J) + VMEAN(J)                                     RRR07340
   80 CONTINUE                                                               RRR07350
C     WRITE OUT NISV, SVCEN                                                  RRR07360
   95 DO 115 M=1,NGPS                                                        RRR07370
      WRITE (6,900) M                                                        RRR07380
      WRITE (6,901) NISV(M)                                                  RRR07390
      WRITE (6,902) (SVCEN(M,J),J=1,NVARS)                                   RRR07400
      WRITE(7,915)(SVCEN(M,J),J=1,NVARS)
915   FORMAT(8F8.4,/8F8.4)
      WRITE (6,903)                                                          RRR07410
      LSTNO = NISV(M)                                                        RRR07420
      K = 0                                                                  RRR07430
      DO 100 I=1,NOBS                                                        RRR07440
      IF (IDATA(I).NE.M) GO TO 100                                           RRR07450
      K = K+1                                                                RRR07460
      IDATAT(K) = IDENT(I)                                                   RRR07470
  100 CONTINUE                                                               RRR07480
C     SORT THE OBSERVATION IDENTIFICATIONS                                   RRR07490
```

```
      L=C                                                                RRR07500
  1C1 L=L+1                                                              RRR07510
      IF (L.EC.NISV(M)) GO TO 110                                        RRRC7520
  1C2 LL=L+1                                                             RRR07530
      IF (ICATAT(L).LE.ICATAT(LL)) GO TC 101                             RRRC7540
      LTEMP = IDATAT(L)                                                  RRR07550
      ICATAT(L) = IDATAT(LL)                                             RRR07560
      ICATAT(LL) = LTEMF                                                 RRR07570
      IF (L.EC.1) GO TC 102                                              RRRC7580
      L=L-1                                                              RRR07590
      GC TO 102                                                          RRRC760C
  11C CCNTINUE                                                           RRRC7610
C     WRITE OUT THE IDENTIFICATICNS                                      RRR07620
      WRITE (6,904) (IDATAT(K),K=1,LSTNC)                                RRRC763C
      WRITE(7,920)(IDATAT(K),K=1,LSTNC)
920   FCRMAT(16I5)
  115 CCNTINUE                                                           RRR07640
C     WRITE OUT THE WITHIN-CLUSTERS MATRIX                               RRR07650
  119 WRITE (6,905)                                                      RRRC766C
      DC 120 J=1,NVARS                                                   RRRC7670
      WRITE (6,906) (W(K,J),K=1,J)                                       RRR07680
  12C CCNTINUE                                                           RRRC7690
C     ICRIT IS THE CRITERION CHOSEN BY THE USER                          RRR0770C
      GC TO (15C,160,17C,17C),ICRIT                                      RRR07710
C     WRITE OUT TRACE W                                                  RRRC772C
  15C WRITE (6,907) CRIT                                                 RRR0773C
      RETURN                                                             RRR07740
C     WRITE OUT DET W                                                    RRR0775C
C     ANC LOG (DET T / DET W)                                            RRR07760
  16C CALL UPRFCT (NVARS,T,M)                                            RRRC777C
      DET = 1.                                                           RRR07780
      CC 165 J=1,NVARS                                                   RRR0779C
  165 DET = DET*T(J,J)                                                   RRR07800
      WRITE (6,912) CRIT                                                 RRRC781C
      CRIT = ALOG 10 ((CET**2)/(CRIT**2))                                RRR07820
      WRITE (6,909) CRIT                                                 RRRC783C
      RETURN                                                             RRR07840
C     FCF LARGEST ROOT ANC HCTELLING'S TRACE CRITERIA, FINC EIGENVALLES  RRRC7850
C     FOR (B-GW)=0.                                                      RRRC786C
  17C DC 175 J=1,NVARS                                                   RRR07870
      DC 175 K=J,NVARS                                                   RRRC788C
      BT(J,K) = B(J,K)                                                   RRR0789C
  175 BT(K,J) = BT(J,K)                                                  RRR0790C
      CRIT = 1.0 / CRIT                                                  RRRC791C
      CALL UTISUI (NVARS,BT,WFCT)                                        RRRC7920
      CALL EIGN (NVARS,BT,EIG,VEC,IND)                                   RRR07930
      IF (IND.NE.C) GO TO 180                                            RRRC794C
C     WRITE OUT EIGENVALLES                                              RRRC7950
```

```
      WRITE (6,910) (EIG(J),J=1,NVARS)                                  RRR07960
      IF (ICRIT.EQ.4) GO TO 178                                         RRR07970
C     WRITE OUT LARGEST ROOT                                            RRR07980
      WRITE (6,913) CRIT                                                RRR07990
      RETURN                                                            RRR08000
C     WRITE OUT SUM OF EIGENVALUES                                      RRR08010
  178 WRITE (6,914) CRIT                                                RRR08020
      RETURN                                                            RRR08030
C     ESCAPE ROUTE                                                      RRR08040
  180 IFINE = 1                                                         RRR08050
      WRITE (6,911) IND                                                 RRR08060
      RETURN                                                            RRR08070
C                                                                       RRR08080
C     FORMAT STATEMENTS                                                 RRR08090
C                                                                       RRR08100
  900 FORMAT ('OCLUSTER ',I2)                                           RRR08110
  901 FORMAT ('        SIZE = ',I3)                                     RRR08120
  902 FORMAT ('        CENTER = ',5(F9.2,1X),/,18X,5(F9.2,1X))
  903 FORMAT ('        THE OBSERVATIONS ARE ')                          RRR08140
  904 FORMAT (8X,5I7,/,9X,5I7)
  905 FORMAT ('1','THE WITHIN GROUP MATRIX IS ')
  906 FORMAT ('        ',10(F10.3,1X))                                  RRR08170
  907 FORMAT ('OTHE TRACE OF THE WITHIN-CLUSTERS MATRIX IS ',E12.6)     RRR08180
  908 FORMAT ('O DETERMINANT T IS ',E12.6)                              RRR08190
  909 FORMAT ('O LOG (DET T / DET W) = ',E12.6)                         RRR08200
  910 FORMAT ('O THE EIGENVALUES OF W(INVERSE)*B ARE ',10F8.3)          RRR08210
  911 FORMAT ('1 ILL-CONDITIONED W MATRIX -- IND IS',I3)                RRR08220
  912 FORMAT('OTHE DETERMINANT OF THE WITHIN-CLUSTERS MATRIX IS ',E12.6)RRR08230
  913 FORMAT ('OTHE LARGEST ROOT IS ',E12.6)                            RRR08240
  914 FORMAT ('OTHE TRACE OF W(INV)*B IS ',E12.6)                       RRR08250
      END                                                               RRR08260
      SUBROUTINE DISTCE (XVECS,YVECS,DIST)                              RRR08270
C                                                                       RRR08280
C     THIS SUBROUTINE CALCULATES EUCLIDIAN, WEIGHTED EUCLIDIAN, OR      RRR08290
C     MAHALANOBIS DISTANCE (DEPENDING ON IDIST).                        RRR08300
C                                                                       RRR08310
      COMMON NOBS,NVARS,NGPS,ICRIT,NCSTAN,IDIST,IFINE,KTIME,ICENT(600), RRR08320
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20), RRR08330
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),B1(20,20), RRR08340
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20)           RRR08350
      DIMENSION X(20),XVECS(1),YVECS(1)                                 RRR08360
      IDIR = IDIST+1                                                    RRR08370
      GO TO (100,200,300),IDIR                                          RRR08380
      WRITE (6,900)                                                     RRR08390
  900 FORMAT ('1 ERROR IN DISTANCE CODE - COLUMN 10 OF PROBLEM CARD')   RRR08400
      IFINE = 1                                                         RRR08410
      RETURN                                                            RRR08420
C                                                                       RRR08430
```

```
C      CALCULATE EUCLIDIAN DISTANCE                                       RRR08440
C                                                                         RRR08450
  100 DIST=0.                                                             RRR08460
      DO 150 J=1,NVARS                                                    RRR08470
  150 DIST = DIST + (XVECS(J)-YVECS(J))**2                                RRR08480
      RETURN                                                              RRR08490
C                                                                         RRR08500
C      CALCULATE EUCLIDIAN DISTANCE FOR STANDARDIZED VARIABLES            RRR08510
C                                                                         RRR08520
  200 DIST=0.                                                             RRR08530
      DO 205 J=1,NVARS                                                    RRR08540
  205 DIST = DIST + (XVECS(J)-YVECS(J))*(1.0/W(J,J))*(XVECS(J)-YVECS(J))RRR08550
      RETURN                                                              RRR08560
C                                                                         RRR08570
C     CALCULATE MAHALANOBIS DISTANCE AS THE ELEMENTS OF                   RRR08580
C     L'(INV)(XVEC-YVEC) SQUARED WHERE L IS THE CHOLESKY FACTOR OF W      RRR08590
C                                                                         RRR08600
  300 DIST=0.                                                             RRR08610
      DO 305 J=1,NVARS                                                    RRR08620
  305 X(J) = XVECS(J) - YVECS(J)                                          RRR08630
      CALL UTIRT (NVARS,1,WFCT,X)                                         RRR08640
      DO 310 J=1,NVARS                                                    RRR08650
  310 DIST = DIST + X(J)**2                                               RRR08660
      RETURN                                                              RRR08670
      END                                                                 RRR08680
      SUBROUTINE CRITON (CRIT)                                            RRR08690
C                                                                         RRR08700
C      THIS SUBROUTINE CALCULATES THE CRITERIA VALUES                     RRR08710
C                                                                         RRR08720
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600), RRR08730
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20), RRR08740
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),E1(20,20), RRR08750
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20)            RRR08760
      GO TO (100,200,300,400),ICRIT                                       RRR08770
      WRITE (6,900)                                                       RRR08780
  900 FORMAT ('1 ERROR IN CRITERION CODE, COLUMN 8 IN PROBLEM CARD.')    RRR08790
      IFINE =1                                                            RRR08800
      RETURN                                                              RRR08810
C                                                                         RRR08820
C      CALCULATE TRACE W                                                  RRR08830
C                                                                         RRR08840
  100 CRIT = 0.                                                           RRR08850
      DO 105 J=1,NVARS                                                    RRR08860
  105 CRIT = CRIT + W(J,J)                                                RRR08870
      RETURN                                                              RRR08880
C                                                                         RRR08890
C      CALCULATE DET W AS THE PRODUCT OF DIAGONAL ELEMENTS OF THE         RRR08900
C      CHOSESKY FACTOR OF W                                               RRR08910
```

```
C                                                                       RRR08920
  200 DET = 1.                                                          RRR08930
      DO 205 J=1,NVARS                                                  RRR08940
  205 DET = DET*WFCT(J,J)                                               RRR08950
      CRIT = DET                                                        RRR08960
      RETURN                                                            RRR08970
C                                                                       RRR08980
C     CALCULATE LARGEST ROOT AS L(INV)*B*L'(INV) WHERE L IS THE         RRR08990
C     CHOLESKY FACTOR OF W                                              RRR09000
C     CRITERION IS RECIPROCAL OF LARGEST ROOT                           RRR09010
C                                                                       RRR09020
  300 DO 305 J=1,NVARS                                                  RRR09030
      DO 305 K=J,NVARS                                                  RRR09040
      BT(J,K) = B(J,K)                                                  RRR09050
  305 BT(K,J) = BT(J,K)                                                 RRR09060
      CALL LTISUI (NVARS,BT,WFCT)                                       RRR09070
      CALL EIGN (NVARS,BT,EIG,VEC,IND)                                  RRR09080
      CRIT = 1.0 / EIG(1)                                               RRR09090
      RETURN                                                            RRR09100
C                                                                       RRR09110
C     CALCULATE HOTELLING'S TRACE AS SUM OF DIAGONAL ELEMENTS OF        RRR09120
C     L(INV)*B*L'(INV)                                                  RRR09130
C     CRITERION IS RECIPROCAL OF THIS SUM                               RRR09140
C                                                                       RRR09150
  400 DO 405 J=1,NVARS                                                  RRR09160
      DO 405 K=J,NVARS                                                  RRR09170
      BT(J,K) = B(J,K)                                                  RRR09180
  405 BT(K,J) = BT(J,K)                                                 RRR09190
      CALL UTISUI (NVARS,BT,WFCT)                                       RRR09200
      CRIT = 0.                                                         RRR09210
      DO 410 J=1,NVARS                                                  RRR09220
  410 CRIT = CRIT+BT(J,J)                                               RRR09230
      CRIT=1.0/ CRIT                                                    RRR09240
      RETURN                                                            RRR09250
      END                                                               RRR09260
      SUBROUTINE WCALC (SV,NI,NGP,IDIR)                                 RRR09270
C                                                                       RRR09280
C     THIS SUBROUTINE CALCULATES THE WITHIN-CELLS MATRIX AND, IF        RRR09290
C     NECESSARY, THE CHOLESKY FACTOR OF THE WITHIN-CELLS MATRIX         RRR09300
C                                                                       RRR09310
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,ICENT(600), RRR09320
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20), RRR09330
     2IDATA(600),NISV(20),VMEAN(20),SC(20),XVEC(20),YVEC(20),BT(20,20), RRR09340
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20)          RRR09350
      DIMENSION SV(20,20),NI(20)                                        RRR09360
C                                                                       RRR09370
C     CALCULATE B AND THEN W = T - B                                    RRR09380
C     B = SUM  NISV(M)*SVCEN(M)**2                                      RRR09390
```

```
C                                                                       RRR09400
      DO 100 J=1,NVARS                                                  RRR09410
      DO 100 K=J,NVARS                                                  RRR09420
  100 B(J,K) = 0.                                                       RRR09430
      DO 105 M=1,NGP                                                    RRR09440
      DO 105 J=1,NVARS                                                  RRR09450
      DO 105 K=J,NVARS                                                  RRR09460
      BT(J,K) = SV(M,J) * SV(M,K)                                       RRR09470
  105 B(J,K) = B(J,K) + BT(J,K) * NI(M)                                 RRR09480
      DO 110 J=1,NVARS                                                  RRR09490
      DO 110 K=J,NVARS                                                  RRR09500
  110 W(J,K) = T(J,K) - B(J,K)                                          RRR09510
C                                                                       RRR09520
C    CALCULATE CHOLESKY FACTOR OF W IF ICIR = 2                         RRR09530
C                                                                       RRR09540
      GO TO (120,115),ICIR                                              RRR09550
  115 DO 116 J=1,NVARS                                                  RRR09560
      DO 116 K=J,NVARS                                                  RRR09570
  116 WFCT(J,K) = W(J,K)                                                RRR09580
      CALL UPFFCT (NVARS,WFCT,M)                                        RRR09590
      IF (M.NE.0) GO TO 125                                             RRR09600
  120 RETURN                                                            RRR09610
  125 IFINE=1                                                           RRR09620
      WRITE (6,900)                                                     RRR09630
  900 FORMAT ('1 THE WITHIN GROUPS MATRIX IS SINGULAR ')                RRR09640
      RETURN                                                            RRR09650
      END                                                               RRR09660
      FUNCTION URAND(IRAND)                                             RRR09670
C                                                                       RRR09680
C     THIS FUNCTION CALCULATES UNIFORMLY DISTRIBUTED RANDOM NUMBERS.    RRR09690
C     BETWEEN 0 AND 1                                                   RRR09700
C     3**19 CONGRUENTIAL UNIFORM RANDOM NUMBER GENERATOR                RRR09710
C                                                                       RRR09720
      IRAND = IRAND*1162261467                                          RRR09730
      IF (IRAND.GT.0) GO TO 3                                           RRR09740
1     IRAND = -IRAND                                                    RRR09750
3     URAND = FLOAT(IRAND)*0.4656612873E-9                              RRR09760
      RETURN                                                            RRR09770
      END                                                               RRR09780
      SUBROUTINE UPRFCT(N,A,M)                                          RRR09790
C                                                                       RRR09800
C     REPLACE UPPER TRIANGLE OF A SQUARE POSITIVE DEFINITE MATRIX A     RRR09810
C     BY ITS CHOLESKI FACTOR                                            RRR09820
C                                                                       RRR09830
      DIMENSION A(20,20)                                                RRR09840
C     CLEAR THE ERROR INDICATOR                                         RRR09850
      M=0                                                               RRR09860
      N1=N-1                                                            RRR09870
      IF(N1) 23C,1CC,1CC                                                   RRR09E8C
  1CC DC 220 K=1,N                                                         RRR09890
      AKK=A(K,K)                                                           RRRC99OC
      IF(K .EC. 1) GO TC 12C                                               RRR09910
      DC 110 J=2,K                                                         RRRC992C
  11C AKK=AKK-A(J-1,K)**2                                                  RRRC9930
  12C IF(A(K,K)) 14C,14C,13C                                               RRRC9940
C     IN THE CASE OF A CCVARIANCE MATRIX AKK/A(K,K) IS 1-R**2 WHERE        RRRC9950
C     R IS MULT CCRRELATICN OF VARIABLE K WITH ALL FRECEECING VARIABLES.RRR09960
  13C IF(AKK/A(K,K) .GE.   .001) GO TC 150                                 RRRC997C
  14C N=K                                                                  RRR09980
      AKK=0.                                                               RRRC999C
  15C AKK=SCRT(AKK)                                                        RRR1CC00
      A(K,K)=AKK                                                           RRR1CC1C
      IF(K .EC. N) GO TC 23C                                               RRR10020
C                                                                          RRR1CC30
      CC 220 I=K,N1                                                        RRR1004C
      AKI=A(K,I+1)                                                         RRR1CC50
      IF(K .EC. 1) GO TC 190                                               RRR10060
      CC 180 J=2,K                                                         RRR10070
  18C AKI=AKI-A(J-1,K)*A(J-1,I+1)                                          RRR10C80
  19C IF(AKK) 210,200,210                                                  RRR10090
  2CC A(K,I+1)= 0.0                                                        RRR10100
      GC TO 220                                                            RRR1C110
  21C A(K,I+1)=AKI/AKK                                                     RRR1012C
  22C CCNTINUE                                                             RRR10130
  23C RETURN                                                               RRR1C14C
      ENC                                                                  RRR1C15C
      SUBRCUTINE LTIRT(M,N,S,B)                                            RRR1C160
C                                                                          RRR1C17C
C     INVERSE CF UPPER (S) TRANSPOSED TIMES RECTANGLE B TRANSPCSEC.        RRR1018C
C                                                                          RRR1C19C
      DIMENSICN S(20,20),B(1,20)                                           RRR10200
      CC 130 J=1,N                                                         RRR1C21C
      CC 130 I=1,M                                                         RRR10220
      SUM = 0.0                                                            RRR1023C
      IF (S(I,I) ) 90,130,90                                               RRR1C24C
   9C SUM = B(J,I)                                                         RRR10250
      IM1 = I-1                                                            RRR1C26C
      IF(IM1) 120,120,1C0                                                  RRR1C27C
  1CC CC 110 K=1,IM1                                                       RRR1C28C
  11C SUM = SUM-S(K,I)*B(J,K)                                              RRR1C290
  12C SUM = SUM/S(I,I)                                                     RRR1C30C
  13C B(J,I) = SUM                                                         RRR1031C
      RETURN                                                               RRR1C32C
      ENC                                                                  RRR10330
      SUBROUTINE LTISUI (N,A,B)                                            RRR1C34C
C                                                                          RRR1C350
```

```
C     UPPER (B) TRANSPOSE INVERSE TIMES A TIMES UPPER (B) INVERSE.
C
      DIMENSION A(20,20),B(20,20)
C     NOTE THAT POSTMULT IS CARRIED OUT ON FINAL VALUES LEFT BY PREMULT
      DO 200 I=1,N
      I1 = I-1
C     NOTE THAT PREMULT ON RIGHT HALF OF ROW IS SAME AS POSTMULT
C     ON LOWER HALF OF COLUMN - EXCEPT FOR DIAG TERM WHICH IS THE
C     FINAL VALUE LEFT BY PREMULT BEFORE POSTMULT
      DO 130 J=I,N
      IF(I1) 120,120,100
  100 DO 110 K=1,I1
  110 A(J,I) = A(J,I) - B(K,I)*A(J,K)
  120 A(J,I) = A(J,I)/B(I,I)
  130 IF (B(I,I) .EQ. 0.0) A(J,I) = 0.0
C     NOTE THAT ELEMENTS IN LEFT HALF OF ROW ARE FINAL FOR PREMULT
C     NOTE THAT DIAG ELEMENT WAS PREVIOUSLY THE FINAL RESULT OF
C     PREMULT.  NOW WE MAKE THE FINAL RESULT OF POSTMULT.
      DO 200 J=1,I
      J1 = J-1
      IF(J1) 160,160,140
C     HORIZONTAL BRANCH OF INNER PRODUCT EXCLUDING DIAG TERM
  140 DO 150 K=1,J1
  150 A(I,J) = A(I,J) - B(K,I)*A(J,K)
  160 IF(I1-J) 190,170,170
C     VERTICAL BRANCH OF INNER PRODUCT INCLUDING DIAG TERM
  170 DO 180 K=J,I1
  180 A(I,J) = A(I,J) - B(K,I)*A(K,J)
  190 A(I,J) = A(I,J)/B(I,I)
  200 IF (B(I,I) .EQ. 0.0) A(I,J) = 0.0
      RETURN
      END
      SUBROUTINE EIGN(NN,A,EIG,VEC,IND)
C         NN= SIZE OF MATRIX
C         A = MATRIX (ONLY LOWER TRIANGLE IS USED + THIS IS DESTROYED)
C         EIG = RETURNED EIGENVALUES IN ALGEBRAIC DESCENDING ORDER
C         VEC = RETURNED EIGENVECTORS IN COLUMNS
C         IND = ERROR RETURN INDICATOR
C               0 FOR NORMAL RETURN
C               1 SUM OF EIGENVALUES NOT EQUAL TO TRACE
C               2 SUM OF EIGENVALUES SQUARED NOT EQUAL TO NORM
C               3 BOTH OF THESE ERRORS
      DIMENSION A(20,20),GAMMA(20),BETA(20),BETASQ(20),EIG(20)
      DIMENSION W(20),VEC(20,20)
C     THE FOLLOWING DIMENSIONED VARIABLES ARE EQUIVALENCED
      DIMENSION P(19),Q(19)
      EQUIVALENCE (P(1),BETA(1)),(Q(1),BETA(1))
      DIMENSION IPOSV(20),IVPOS(20),IORD(20)
      EQUIVALENCE (IPOSV(1),GAMMA(1)),(IVPOS(1),BETA(1)),                RRR10840
     1(ICRD(1),BETASQ(1))                                                RRR10850
      N=NN                                                               RRR10860
C     RESET ERROR RETURN INDICATOR                                       RRR10870
      IND=0                                                              RRR10880
      IF(N .EQ. 0) GO TO 560                                             RRR10890
      N1=N-1                                                             RRR10900
      N2=N-2                                                             RRR10910
C     COMPUTE THE TRACE AND EUCLIDIAN NORM OF THE INPUT MATRIX           RRR10920
C     LATER CHECK AGAINST SUM AND SUM OF SQUARES OF EIGENVALUES          RRR10930
      ENORM=0.
      TRACE=0.                                                           RRR10950
      DO 110 J=1,N                                                       RRR10960
      DO 100 I=J,N                                                       RRR10970
  100 ENORM=ENORM+A(I,J)**2                                              RRR10980
      TRACE=TRACE+A(J,J)                                                 RRR10990
  110 ENORM=ENORM-.5*A(J,J)**2                                           RRR11000
      ENORM=ENORM+ENORM                                                  RRR11010
      GAMMA(1)=A(1,1)                                                    RRR11020
      IF(N2) 280,270,120                                                 RRR11030
  120 DO 260 NR=1,N2                                                     RRR11040
      B=A(NR+1,NR)                                                       RRR11050
      S=0.                                                               RRR11060
      DO 130 I=NR,N2                                                     RRR11070
  130 S=S+A(I+2,NR)**2                                                   RRR11080
C     PREPARE FOR POSSIBLE BYPASS OF TRANSFORMATION                      RRR11090
      A(NR+1,NR)=0.                                                      RRR11100
      IF(S) 250,250,140                                                  RRR11110
  140 S=S+B*B                                                            RRR11120
      SGN=+1.                                                            RRR11130
      IF(B) 150,160,160                                                  RRR11140
  150 SGN=-1.                                                            RRR11150
  160 SQRTS=SQRT(S)                                                      RRR11160
      D=SGN/(SQRTS+SQRTS)                                                RRR11170
      TEMP=SQRT(.5+B*D)                                                  RRR11180
      W(NR)=TEMP                                                         RRR11190
      A(NR+1,NR)=TEMP                                                    RRR11200
      D=D/TEMP                                                           RRR11210
      B=-SGN*SQRTS                                                       RRR11220
C     D IS FACTOR OF PROPORTIONALITY. NOW COMPUTE AND SAVE W VECTOR.     RRR11230
C     EXTRA SINGLY SUBSCRIPTED W VECTOR USED FOR SPEED.                  RRR11240
      DO 170 I=NR,N2                                                     RRR11250
      TEMP=D*A(I+2,NR)                                                   RRR11260
      W(I+1)=TEMP                                                        RRR11270
  170 A(I+2,NR)=TEMP                                                     RRR11280
C     PREMULTIPLY VECTOR W BY MATRIX A TO OBTAIN P VECTOR.               RRR11290
C     SIMULTANEOUSLY ACCUMULATE DOT PRODUCT WP,(THE SCALAR K)            RRR11300
      WTAW=0.                                                            RRR11310
      DO 220 I=NR,N1                                                    RRR11320
      SUM=0.                                                            RRR11330
      DO 180 J=NR,I                                                     RRR11340
  180 SUM=SUM+A(I+1,J+1)*W(J)                                           RRR11350
      I1=I+1                                                            RRR11360
      IF(N1-I1) 210,190,190                                             RRR11370
  190 DO 200 J=I1,N1                                                    RRR11380
  200 SUM=SUM+A(J+1,I+1)*W(J)                                           RRR11390
  210 P(I)=SUM                                                          RRR11400
  220 WTAW=WTAW+SUM*W(I)                                                RRR11410
C     P VECTOR AND SCALAR K NOW STORED. NEXT COMPUTE Q VECTOR           RRR11420
      DO 230 I=NR,N1                                                    RRR11430
  230 Q(I)=P(I)-WTAW*W(I)                                               RRR11440
C     NOW FORM PAP MATRIX, REQUIRED PART                                RRR11450
      DO 240 J=NR,N1                                                    RRR11460
      QJ=Q(J)                                                           RRR11470
      WJ=W(J)                                                           RRR11480
      DO 240 I=J,N1                                                     RRR11490
  240 A(I+1,J+1)=A(I+1,J+1)-2.*(W(I)*QJ+WJ*Q(I))                        RRR11500
  250 BETA(NR)=B                                                        RRR11510
      BETASQ(NR)=B*B                                                    RRR11520
  260 GAMMA(NR+1)=A(NR+1,NR+1)                                          RRR11530
  270 B=A(N,N-1)                                                        RRR11540
      BETA(N-1)=B                                                       RRR11550
      BETASQ(N-1)=B*B                                                   RRR11560
      GAMMA(N)=A(N,N)                                                   RRR11570
  280 BETASQ(N)=0.                                                      RRR11580
C     ADJOIN AN IDENTITY MATRIX TO BE POSTMULTIPLIED BY ROTATIONS.      RRR11590
      DO 300 I=1,N                                                      RRR11600
      DO 290 J=1,N                                                      RRR11610
  290 VEC(I,J)=0.                                                       RRR11620
  300 VEC(I,I)=1.                                                       RRR11630
      M=N                                                               RRR11640
      SUM=0.                                                            RRR11650
      NPAS=1                                                            RRR11660
      GO TO 400                                                         RRR11670
  310 SUM=SUM+SHIFT                                                     RRR11680
      COSA=1.                                                           RRR11690
      G=GAMMA(1)-SHIFT                                                  RRR11700
      PP=G                                                              RRR11710
      PPBS=PP*PP+BETASQ(1)                                              RRR11720
      PPBR=SQRT(PPBS)                                                   RRR11730
      DO 370 J=1,M                                                      RRR11740
      COSAP=COSA                                                        RRR11750
      IF(PPBS .NE. 0.) GO TO 320                                        RRR11760
      SINA=0.                                                           RRR11770
      SINA2=0.                                                          RRR11780
      COSA=1.                                                           RRR11790
      GO TO 350                                                         RRR11800
  320 SINA=BETA(J)/PPBR                                                 RRR11810
      SINA2=BETASQ(J)/PPBS                                              RRR11820
      COSA=PP/PPBR                                                      RRR11830
C     POSTMULTIPLY IDENTITY BY P-TRANSPOSE MATRIX                       RRR11840
      NT=J+NPAS                                                         RRR11850
      IF(NT .GE. N) NT=N                                                RRR11860
  330 DO 340 I=1,NT                                                     RRR11870
      TEMP=COSA*VEC(I,J)+SINA*VEC(I,J+1)                                RRR11880
      VEC(I,J+1)=-SINA*VEC(I,J)+COSA*VEC(I,J+1)                         RRR11890
  340 VEC(I,J)=TEMP                                                     RRR11900
  350 DIA=GAMMA(J+1)-SHIFT                                              RRR11910
      U=SINA2*(G+DIA)                                                   RRR11920
      GAMMA(J)=G+U                                                      RRR11930
      G=DIA-U                                                           RRR11940
      PP=DIA*COSA-SINA*COSAP*BETA(J)                                    RRR11950
      IF(J .NE. M) GO TO 360                                            RRR11960
      BETA(J)=SINA*PP                                                   RRR11970
      BETASQ(J)=SINA2*PP*PP                                             RRR11980
      GO TO 380                                                         RRR11990
  360 PPBS=PP*PP+BETASQ(J+1)                                            RRR12000
      PPBR=SQRT(PPBS)                                                   RRR12010
      BETA(J)=SINA*PPBR                                                 RRR12020
  370 BETASQ(J)=SINA2*PPBS                                              RRR12030
  380 GAMMA(M+1)=G                                                      RRR12040
C     TEST FOR CONVERGENCE OF LAST DIAGONAL ELEMENT                     RRR12050
      NPAS=NPAS+1                                                       RRR12060
      IF(BETASQ(M) .GT. 1.E-21) GO TO 410                               RRR12070
  390 EIG(M+1)=GAMMA(M+1)+SUM                                           RRR12080
  400 BETA(M)=0.                                                        RRR12090
      BETASQ(M)=0.                                                      RRR12100
      M=M-1                                                             RRR12110
      IF(M .EQ. 0) GO TO 430                                            RRR12120
      IF(BETASQ(M) .LE. 1.E-21) GO TO 390                               RRR12130
C     TAKE ROOT OF CORNER 2 BY 2 NEAREST TO LOWER DIAGONAL IN VALUE     RRR12140
C     AS ESTIMATE OF EIGENVALUE TO USE FOR SHIFT                        RRR12150
  410 A2=GAMMA(M+1)                                                     RRR12160
      R2=.5*A2                                                          RRR12170
      R1=.5*GAMMA(M)                                                    RRR12180
      R12=R1+R2                                                         RRR12190
      DIF=R1-R2                                                         RRR12200
      TEMP=SQRT(DIF*DIF+BETASQ(M))                                      RRR12210
      R1=R12+TEMP                                                       RRR12220
      R2=R12-TEMP                                                       RRR12230
      DIF=ABS(A2-R1)-ABS(A2-R2)                                         RRR12240
      IF(DIF .LT. 0.) GO TO 420                                         RRR12250
      SHIFT=R2                                                          RRR12260
      GO TO 310                                                         RRR12270
  420 SHIFT=R1                                                           RRR12280
      GO TO 310                                                          RRR12290
  430 EIG(I)=GAMMA(I)+SUM                                                RRR12300
C     INITIALIZE AUXILIARY TABLES REQUIRED FOR REARRANGING THE VECTORS   RRR12310
      DO 440 J=1,N                                                       RRR12320
      IPOSV(J)=J                                                         RRR12330
      IVPOS(J)=J                                                         RRR12340
  440 IORD(J)=J                                                          RRR12350
C     USE A TRANSPOSITION SORT TO ORDER THE EIGENVALUES                  RRR12360
      M=N                                                                RRR12370
      GO TO 470                                                          RRR12380
  450 DO 460 J=1,M                                                       RRR12390
      IF(EIG(J) .GE. EIG(J+1)) GO TO 460                                 RRR12400
      TEMP=EIG(J)                                                        RRR12410
      EIG(J)=EIG(J+1)                                                    RRR12420
      EIG(J+1)=TEMP                                                      RRR12430
      ITEMP=IORD(J)                                                      RRR12440
      IORD(J)=IORD(J+1)                                                  RRR12450
      IORD(J+1)=ITEMP                                                    RRR12460
  460 CONTINUE                                                           RRR12470
  470 M=M-1                                                              RRR12480
      IF(M .NE. 0) GO TO 450                                             RRR12490
      IF(N1 .EQ. 0) GO TO 500                                            RRR12500
      DO 490 L=1,N1                                                      RRR12510
      NV=IORD(L)                                                         RRR12520
      NP=IPOSV(NV)                                                       RRR12530
      IF(NP .EQ. L) GO TO 490                                            RRR12540
      LV=IVPOS(L)                                                        RRR12550
      IVPOS(NP)=LV                                                       RRR12560
      IPOSV(LV)=NP                                                       RRR12570
      DO 480 I=1,N                                                       RRR12580
      TEMP=VEC(I,L)                                                      RRR12590
      VEC(I,L)=VEC(I,NP)                                                 RRR12600
  480 VEC(I,NP)=TEMP                                                     RRR12610
  490 CONTINUE                                                           RRR12620
  500 ESUM=0.                                                            RRR12630
      ESSQ=0.                                                            RRR12640
C     BACK TRANSFORM THE VECTORS OF THE TRIPLE DIAGONAL MATRIX           RRR12650
      DO 550 NRR=1,N                                                     RRR12660
      K=N1                                                               RRR12670
  510 K=K-1                                                              RRR12680
      IF(K .LE. 0) GO TO 540                                             RRR12690
      SUM=0.                                                             RRR12700
      DO 520 I=K,N1                                                      RRR12710
  520 SUM=SUM+VEC(I+1,NRR)*A(I+1,K)                                      RRR12720
      SUM=SUM+SUM                                                        RRR12730
      DO 530 I=K,N1                                                      RRR12740
  530 VEC(I+1,NRR)=VEC(I+1,NRR)-SUM*A(I+1,K)                             RRR12750
```

```
      GO TO 510                                                         RRR12760
  540 ESUM=ESUM+EIG(NRR)                                                RRR12770
  550 ESSQ=ESSQ+EIG(NRR)**2                                             RRR12780
      TEMP=ABS(512.*TRACE)                                              RRR12790
      IF((ABS(TRACE-ESUM)+TEMP)-TEMP .NE. 0.) INC=INC+1                 RRR12800
      TEMP=1024.*ENORM                                                  RRR12810
      IF((ABS(ENORM-ESSQ)+TEMP)-TEMP .NE. 0.) INC=INC+2                 RRR12820
  560 RETURN                                                            RRR12830
      END                                                               RRR12840
```

Appendix C:

```
RUN NAME         PRINCIPAL COMPONENT ANALYSIS OF TANKS
VARIABLE LIST    NATIONS,X1 TO X10
INPUT MEDIUM     CARD
INPUT FORMAT     FIXED(1X,F2.0,5X,8F8.3/8X,2F8.3)
N OF CASES       24
VAR LABELS       NATIONS TYPE OF TANK/
                 X1 WEIGHT/X2 LENGTH/X3 WIDTH/X4 HEIGHT/X5 ROAD SPEED/
                 X6 TRENCH CROSSING/X7 GROUND PRESSURE/X8 MAX ARMAMENT/
                 X9 GROUND CLEARANCE/X10 POWER TO ENGINE RATIO/
FACTOR           VARIABLES=X1 TO X10/TYPE=PA1/NFACTORS=10/
OPTIONS          8,11
STATISTICS       ALL
READ INPUT DATA
   DATA    DECK
       .
       .
       .
       .
       .
FINISH
```

Appendix D:

```
RUN NAME         TANK DISCRIMINANT
VARIABLE LIST    NATIONS,X1 TC X10
N CF CASES       24
INPLT MECIUM     CAFD
INPLT FCRMAT     FI)ED(1X,F2.C,5X,8F8.3/6X,2F8.3)
RECCCE           NATIONS(11 THRU 13=1)(15,16,17,24,25,26,3C,31,32,33=2)
                 (19,20,21,22,23,28,29,=3)(14,18,27,34=4)
VAR LABELS       NATIONS TYPE OF TANK/
                 X1 WEIGHT/X2 LENGTH/X3 WICTH/X4 HEIGHT/X5 RCAD SPEEC/
                 X6 TRENCH CROSSING/X7 GRCUND PRESSLRE/X8 MAX ARMAMENT/
                 X9 GROUNC CLEARANCE/X10 FCWER TO ENGINE RATIC/
CISCRIMINANT     GRCUPS=NATICNS(1,4)/VARIABLES=X1 TC X10/
                 ANALYSIS=X1 TO X10/METHCC=MAHAL/
CPTICNS          5,6,7,9,11,12
STATISTICS       ALL
REAC INPLT DATA
  CATA     DECK

      .
      .
      .

      .

      .
FINISH
```

BIBLIOGRAPHY

1. Aiken, J. W., "Development of Cluster Analysis for Student Opinion Data", Master's Thesis, Naval Postgraduate School, Monterey, CA, 1979.

2. Anderson, T. W., An Introduction to Multivariate Statistical Analysis, Wiley, 1958.

3. Anderberg, M. R., Cluster Analysis for Applications, Academic Press, 1973.

4. Barr, D. R., and Richards, F. R., Utility Assessment Methodology (Report on Relative Utility Score of the AN-TPQ/27). System Exploration Inc., Monterey, CA, 1980.

5. Eisenheis, R. A., and Avery, R. B., Discriminant Analysis and Classification Procedures, Lexington Books, 1972.

6. Friedman, H. P., and Rubin, J., "On Some Invariant Criteria for Grouping Data", Journal of the American Statistical Association, Vol. 62: 320, Dec. 1967.

7. Giri, N. C., Multivariate Statistical Inference, Academic Press, 1977.

8. Gnanadesikan, R., Methods for Statistical Data Analysis of Multivariate Observations, Wiley, 1977.

9. Green, P. E., and Carroll, J. D., Mathematical Tools for Applied Multivariate Analysis, Academic Press, 1976.

10. Hartigan, J. A., Clustering Algorithms, Wiley, 1975.

11. Kandell, M. G., The Advanced Theory of Statistics, Vol. III, Charles Griffin and Company, 1947.

12. Keeney, R. L., and Raiffa, H., Decisions with Multiple Objectives, Wiley, 1976.

13. Klecka, W. R., "Discriminant Analysis", SPSS (Statistical Package for the Social Sciences), McGraw Hill, 1975.

14. Kim, J. O., "Factor Analysis", SPSS (Statistical Package for the Social Sciences), McGral Hill, 1975.

15. MacQueen, J., "Some Methods for Classification and Analysis of Multivariate Observations", Paper presented at Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California, 1965-1966.

16. McRae, D. J., Clustering Multivariate Observations, Doctoral Dissertation, University of North Carolina, Chapel Hill, N.C., 1973.

17. Morrison, D. F., Multivariate Analysis: Technique for Educational and Psychological Research, Wiley, 1971.

18. The IMSL Library, Vol. 3, IMSL, Inc., 1979.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center (DTIC) Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	4
4.	Professor F. R. Richards, Code 55Rh (Thesis Advisor) Department of Operations Research Naval Postgraduate School Monterey, California 93940	2
5.	Professor D. R. Barr, Code 55Bn (Second Reader) Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
6.	LTC Jin Ki Lee, ROK Army (Student) 6-601, Gun-In Apt, Dong-Bing-Go Dong Yong-San, Seoul Korea	5

LMED

-8