AN EXAMINATION OF THE EFFECT OF ERROR PERTURBATION OF
CONSTRUCTED DATA ON FIFTEEN CLUSTERING ALGORITHMS

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

By
Glenn Wesley Milligan, B.A., M.A.

* * * * *

The Ohio State University
1978

Reading Committee:
Paul D. Isaac, Ph.D.
Robert C. MacCallum, Ph.D.
George E. Policello, II, Ph.D.
Richard J. Jagacinski, Ph.D.

Approved By:

Advisor
Department of Psychology
© Copyright by
Glenn Wesley Milligan
1978
to my parents, Donald and Wanda Milligan
ACKNOWLEDGEMENTS

The author would like to thank all of the members of the Reading Committee for their help and encouragement in the preparation of this work. Particular appreciation is extended to Dr. Paul Isaac and Dr. Robert MacCallum who have followed my interest in the field of clustering and have provided valuable guidance throughout my graduate education at The Ohio State University.

The author would also like to thank Michele Scott for providing extensive typing and proofreading services. Finally, the Instruction and Research Computer Center at The Ohio State University provided the program listing of the random number generator given in the appendix.
VITA

June 11, 1949 . . . . Born - Enid, Oklahoma

1971. . . . . . . . . . B. A., University of Southern California, Los Angeles, California

1973-1974 . . . . . . Graduate Assistant, Department of Psychology, California State University at Long Beach, Long Beach, California

1974. . . . . . . . . . M. A., California State University at Long Beach, Long Beach, California

1974-1978 . . . . . . Graduate Teaching Associate, Department of Psychology, The Ohio State University, Columbus, Ohio

1977. . . . . . . . . . Research Assistant, College of Administrative Sciences, Faculty of Marketing, The Ohio State University, Columbus, Ohio
FIELDS OF STUDY

Major Field:  Quantitative Psychology

Studies in Applied Statistics, Multidimensional Scaling, and Clustering. Professor Paul D. Isaac

Studies in Factor Analysis, Monte Carlo Simulation, and Clustering. Professor Robert C. MacCallum

Studies in Analysis of Variance and Mathematical Psychology. Professor Mari R. Jones

Studies in Mathematical Psychology and Individual Preference Behavior. Professor Thomas Nygren

Minor Field:  Mathematical Statistics

Studies in Times Series Analysis and Non-Parametric Methods. Professor Michael A. Fligner

Studies in Multivariate Analysis. Professor R. C. Srivastava

Studies in Simulation Methods and Stochastic Processes. Professor William A. T. Archambault

Studies in Multidimensional Contengency Tables. Professor G. E. Policello
# TABLE OF CONTENTS

**ACKNOWLEDGMENTS** ........................................ iii

**VITA.** ...................................................... iv

**LIST OF TABLES.** ........................................ viii

**Chapter**

I. **INTRODUCTION.** ........................................ 1

   Man's Desire to Cluster .................................. 1
   The Clustering Process ................................... 8
   The Problem and Necessity of Validating
   Clustering Procedures .................................. 24
   Validation Measures ..................................... 35
   Objectives of the Present Study ....................... 48

II. **METHODS AND MATERIALS** .......................... 54

   Data Generation and Error Perturbation
   Process .................................................. 54
   Clustering Algorithms .................................. 68
   Recovery Measures ...................................... 75

III. **RESULTS** ............................................ 82

   External Criterion Measure ............................. 82
   Internal Criterion Measures ............................ 106
   Correlation Measure of the Error
   Perturbation Process ................................. 116

IV. **DISCUSSION.** ......................................... 118

   Evaluation of Recovery Performance .................. 118
   Evaluation of Internal Criteria ...................... 128
Chapter

V. SUMMARY ................................................. 132

APPENDICES. .................................................. 135

A. Computer Program Listing for Data Generation
   Process ................................................. 135

B. Computer Program Listings for Clustering
   Algorithms .............................................. 161

BIBLIOGRAPHY ................................................. 200
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Point Distribution Across Clusters</td>
<td>58</td>
</tr>
<tr>
<td>2.</td>
<td>Hierarchical Algorithms</td>
<td>70</td>
</tr>
<tr>
<td>3.</td>
<td>Exact Recovery Frequencies</td>
<td>83</td>
</tr>
<tr>
<td>4.</td>
<td>Rank Order Performance Within Error Condition for Exact Recovery Frequencies</td>
<td>84</td>
</tr>
<tr>
<td>5.</td>
<td>Mean Recovery Rates - Hierarchical Methods</td>
<td>91</td>
</tr>
<tr>
<td>6.</td>
<td>Mean Recovery Rates - Non-Hierarchical Methods</td>
<td>92</td>
</tr>
<tr>
<td>7.</td>
<td>Rank Order Performance Within Error Condition for the Mean Recovery Rates.</td>
<td>93</td>
</tr>
<tr>
<td>8.</td>
<td>Analysis of Variance - Rand Statistic.</td>
<td>98</td>
</tr>
<tr>
<td>9.</td>
<td>Simple Main Effects Tests for Conditions Across Methods</td>
<td>100</td>
</tr>
<tr>
<td>10.</td>
<td>Simple Main Effects Tests for Methods Across Conditions</td>
<td>105</td>
</tr>
<tr>
<td>11.</td>
<td>Mean Single-Partition Cophenetic Correlations - Hierarchical Methods</td>
<td>107</td>
</tr>
<tr>
<td>12.</td>
<td>Mean Single-Partition Cophenetic Correlations - Non-Hierarchical Methods</td>
<td>108</td>
</tr>
<tr>
<td>13.</td>
<td>Correlations by Method Between the External Criterion and the Internal Criterion</td>
<td>110</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>15.</td>
<td>Mean Hierarchical Cophenetic Correlations</td>
<td>113</td>
</tr>
<tr>
<td>16.</td>
<td>Analysis of Variance Tests for the Hierarchical Cophenetic Correlation for Each Hierarchical Method</td>
<td>115</td>
</tr>
<tr>
<td>17.</td>
<td>Mean Error Perturbation Correlations</td>
<td>117</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

Man's Desire to Cluster

Classification is a basic human mental process. "None of us can communicate, or even live, unless we sort or classify the phenomena of our experience; further, we need in some way to label the classes or sets to which we assign these phenomena" (Johnson, 1968, p. 10). Virtually every academic discipline involves itself in some sort of classificatory activity. Such data groupings can be extremely useful. Relevant data groupings can provide economy of memory, predictive power, or possible theoretical interpretation. Most of the classificatory activity is carried out at a subjective level of analysis. However, with the advent of high speed computational equipment, several disciplines have been involved in the development of more or less automatic or objective algorithms for the generation of a classification of a set of objects. The biological sciences were one of the first disciplines to become involved in the development of such computer oriented clustering algorithms. However, the social sciences were also involved at an early stage in the development of a certain class of clustering procedures known as ordination methods.
Many researchers are not aware of the immense amount of activity in the field of classification. Sokal, a fairly well known figure in the area, has been quoted as asserting that no less than 1,600 scholarly articles appeared on the topic of clustering between the years 1958 to 1973 (Moss and Hendrickson, 1973). A review of more recent literature indicates that interest in the topic has continued to grow. Blashfield and Aldenderfer (1977) report that for the year 1975 alone, at least 532 articles can be identified which relate to the topic.

Despite this wide base of interest in the development and use of clustering procedures, the literature on the topic is remarkably segmented with authors in one academic discipline seemingly isolated from research in other fields. Blashfield (1976) and Blashfield and Aldenderfer (1977) have documented the surprising lack of cross reference between disciplines. Identical clustering methods are often given completely different names, even within disciplines. It is not unusual to find that identical or highly similar techniques have been simultaneously discovered or rediscovered in different fields at about the same point in time. In fact, there is little agreement as to the appropriate name of the science of classification itself. Classification, clustering, cluster analysis, numerical taxonomy, and grouping methods are but a few of the names which have been used. Perhaps the most interesting and humorous discussion as to
the appropriate name for the field was given by Johnson (1968). Johnson considered three terms: Taximetrics, taxometrics, and taxonometrics. Johnson's discussion and arguments for the adoption of one term over another was based on a detailed discussion of etymology.

However, in fairness to the applied researcher, most discussions of clustering procedures are embedded in or at least illustrated with content material specific to a given field. It is somewhat unreasonable to expect, for example, that a psychologist should have a working understanding of forested Minnesota soils, or the general theory of cladistics. Likewise, it is also unreasonable for a biological scientist to be expected to have a detailed understanding of Hull's theory of learning or various psychotherapy processes. It is clearly difficult, and perhaps undesirable in some cases, to write articles on the topic of clustering and not to relate the procedures to content issues in various disciplines.

On the other hand, the underlying theme of all of the research activity in clustering techniques appears to attempt to provide meaningful classifications of the data by partitioning the elements into homogeneous subgroups. Thus, of primary or common interest should be the concept or definition of the term cluster itself.
The Concept of a Cluster

Since clustering techniques are quantitative, a mathematical definition would seem desirable. However, no real attempts have been made in this direction. The problem may be virtually unsolvable. A truly general description would have to be invariant to the types of metric spaces in which the clusters exist and also invariant to the various structures that clusters can assume. Some researchers seem willing to accept the notion that clusters represent mixtures of multivariate normal samples (Wolfe, 1970). Certainly, the multivariate normality assumption allows the researcher to invoke a considerable amount of statistical distribution theory. However, most individuals working in the field have not adopted such a view. Some mathematical criteria have been given by Fisher and Van Ness (1971), Jardine and Sibson (1968a, 1971b), and Rubin (1967). However, these criteria do not actually define the nature of clusters. Rather, the criteria are offered as characteristics which should be satisfied by any given clustering method. Unfortunately, some researchers in the field have rejected many of the proposed criteria, especially the criteria given by Jardine and Sibson (see Williams, Lance, Dale, and Clifford, 1971).

Several verbal descriptions of the concept of a cluster have been offered. Sneath (1969) has stated that "In a broad sense clusters are thought of as collections of points which are relatively close, but which are separated by empty
regions of space from other clusters" (p. 260). Sneath goes on to comment that clusters could be overlapping and be hidden by outlier points. Everitt (1974) has suggested that "A cluster is a set of entities which are alike, and entities from different clusters are not alike" (p. 43). Wallace and Boulton (1968) also reflected this view by asserting that "The objects within a single class must essentially be equivalent at some level of discourse" (p. 185).

Gengerelli (1963) returns to a more spatial definition where a cluster is considered to be "An aggregate of points in the test space such that the distance between any two points in the cluster is less than the distance between any point in the cluster and any point not in it" (p. 458). Carmichael, George, and Julius (1968) expressed a similar spatial definition which was somewhat less restrictive. Hubert (1972), and Baker and Hubert (1975) used a definition very similar to that of Gengerelli when they defined their goodness-of-fit cluster statistic. Van Rijsbergen (1970) also offered an identical definition which was based on similarities rather than distances.

However, such definitions can be considered overly restrictive. Requiring that all elements within any given cluster to be highly similar to each other automatically rules out more elongated clusters where all immediate neighbors are similar but elements at opposite extremes of the cluster may not be highly similar. Cattell and Coulter
(1966) and Johnson (1967) recognized this aspect of cluster structure and proposed two definitions of clusters. In one type of cluster, entities are considered to be optimally compacted, whereas, in other types of clusters entities are considered to be optimally connected. Clearly, elongated clusters may be of great theoretical importance, and certainly any researcher should be warned of their presence in a data set. The more restrictive definition allowing only for compact clusters has been linked to certain variance restriction characteristics by Needham (1965), Rogers, Fleming, and Estabrook (1967), and Wishart (1969).

A less restrictive definition seems desirable. The less restrictive definitions tend to be somewhat spatial in their description. The definition of natural clusters as illustrated by Everitt (1974) is one such example. Cormack (1971) has offered a two part definition of cluster structure. Clusters should exhibit both external isolation and internal cohesion. External isolation requires that entities in one cluster should be separated from entities in another cluster by fairly empty areas of space. Internal cohesion requires that entities within a cluster should be similar to each other, at least within the local metric. Clusters may be described as being located in finite regions of space containing relatively high point densities. Such regions of high density are separated from other such regions by areas containing a relatively low point density.
A completely different conceptualization of cluster structure allows for overlapping clusters. Such clusters have often been termed cliques (Cattell and Coulter, 1966; Cormack, 1971; and Peay, 1975). However, the usefulness of such overlapping structures is probably limited to certain substantive areas. For example, there may be little benefit for a marketing researcher to discover that certain sub-samples overlap in a marketing segmentation study. Likewise, a biologist attempting to construct a phylogenetic classification could be seriously misled when the clustering results provide overlapping data structures. Overlapping structures imply that at least some proportion of two or more samples are phenotypically identical on the measured variables. However, numerous well documented cases of evolutionary mimicry have been discovered where two seemingly identical species actually belonged to not only different species and genus, but also different families (Stebbins, 1966). However, in some circumstances, the overlapping structures may be very useful as in the study of semantics or linguistics (Shepard, 1974).

The different conceptualizations of cluster structure have actually led some researchers away from offering a definition of the term. Sneath and Sokal, in one of the most important works in the field of classification, decided to "Leave the definition of clusters conveniently vague: Sets of OTU's [Operational Taxonomic Units] in phenetic
hyperspace that exhibit neither random nor regular distribution patterns and that meet one or more of various criteria imposed by a particular cluster definition" (1973, p. 194-195). Sneath and Sokal also noted that the conceptualization of cluster structure is affected by the algorithm selected. They assert "That although the end results of the clustering process is necessarily determined by the inherent structure and relations of the OTU's, it is also appreciably affected by the choice of a clustering algorithm" (1973, p. 189). Bonner (1964) also argued that no simple definition is sufficient. Bonner believes that the ultimate criterion is the value judgment of the user. All that is required is that in using the term in a given way, an answer of value to the researcher is produced.

Although such definitions seem to dodge the issue of defining the term cluster, such vague definitions may more accurately reflect the true complexity of the clustering process.

The Clustering Process

Rather than possessing a unified strategy, the clustering process actually consists of a series of steps where each step represents a choice for the researcher (Johnson, 1968; Sneath and Sokal, 1973). The researcher is responsible for selecting the elements to be clustered along with the variables which are to be measured. The researcher must next decide among a vast array of candidates for the
similarity or distance index that is used to prepare the data for input to a clustering algorithm. The researcher must then select the actual clustering algorithm that is to be used to provide the final partition solution. Quite often the algorithm itself may have a set of adjustable parameters which require specification by the researcher. Invariably, the researcher must determine the number of clusters in the final solution. Finally, the researcher can freely choose among any number of cluster evaluation procedures, including no evaluation at all.

As it turns out, some computer packages developed for cluster analysis applications do not always give the user the ability to make all choices freely. Some programs have pre-established choices which, in essence, makes the decision for the researcher. For example, the program may always standardize the variables or automatically compute Euclidean distances. Neither of the two choices are necessarily appropriate for all data sets. Such option free procedures may reduce the number of decisions a researcher might have to make, but this is gained at the possibility of obtaining meaningless results. Another danger of such option free procedures is that a novice user will not necessarily comprehend the complexity of the analysis undertaken.

The remainder of the present section is divided into two parts. The first part surveys the topics of data selection and similarity measures. The second topic offers an
overview of the clustering algorithms themselves. Although these areas do not represent all of the choices facing the user, the topics represent the areas which have attracted the greatest attention. It should be remembered that errors of judgment with regard to the other decisions can easily lead to nonsensical or incorrect data partitions.

**Data Selection and Similarity Measures**

Biologists have been somewhat more sensitive to the problem of the selection of entities and variables than social scientists. Biologists have noted that on occasion improper sampling of entities can leave an entire species or sub-species unrepresented in a data set (Johnson, 1968). Furthermore, if a group is not properly represented in a data set, the remaining hierarchy structure may be distorted to some extent. The deletion of relevant variables can also leave a species improperly represented or poorly discriminated from another group in a data set. It is also clear that irrelevant variables should not be used. However, biologists in general have a greater theoretical basis or framework from which to work. Social scientists do not have such well developed theories of evolution and genetic inheritance. Thus, it is unlikely that any generally acceptable set of rules could be developed to guide in the selection of subjects and variables for the social sciences.
A researcher must use all available information on a case by case basis in order to decide on the variables and entities to be used.

The topic of the selection of the appropriate similarity measure has received a great deal of attention. Large portions of the standard texts on clustering are devoted to a discussion of similarity measures (Anderberg, 1973; Sokal and Sneath, 1963). The selection of the appropriate similarity measure has long been recognized as being of critical importance (Cronbach and Gleser, 1953). The similarity coefficient is of such critical importance because it determines in a large part the exact point distribution in the multivariate space defined by the variables. Each measure retains only certain aspects of the information in the original data. Euclidean distances retain a great deal of information while other measures retain only specific aspects. It seems almost certain that the choice of an inappropriate similarity measure would seriously distort the true clustering in a variable set where distinct clustering is actually present.

One type of data, dichotomous or category scale data, has been given what seems to be an almost inordinate amount of attention with regard to the development of similarity coefficients. Actually, a great deal of biological data is in the form of the presence or absence of given traits or characteristics. Comprehensive reviews of such measures can
be found in Anderberg (1973), Cheetham and Hazel (1969), and Sokal and Sneath (1963). The characteristics of each index have been well documented. Different indices enhance, reduce, or ignore certain aspects of information from the raw data. For example, the simple matching coefficient gives equal weight to both matches and mismatches, while also treating present-present matches the same as absent-absent matches. The Jaccard index, on the other hand, excludes the count for the absent-absent matches while still providing equal weighting. The Rogers and Tanimoto measure gives double weighting for unmatched pairs while including the absent-absent matches. Several other measures have also been proposed. It should be noted that some attempts have been made at examining mixed variable scale problems. That is, situations do occur where a researcher wants to use variables of differing scale type. Anderberg (1973) has provided one of the more comprehensive reviews of these techniques.

Interval scale based measures have also been extensively examined (Anderberg, 1973; Cormack, 1971; and Cronbach and Gleser, 1953). These coefficients include the simple Euclidean distance metric, the city block metric, the more general Minkowski metrics, angular separation, Pearson correlation, Cattell's $r_p$, generalized distance (Mahalanobis distance), Canberra metric, and others. Considerable care must be exercised when selecting among such indices. In
fact, care must be taken when selecting indices of the same type. Blashfield (1977b) determined that three commonly available computer programs for producing hierarchical clustering solutions used three different definitional formulas for Euclidean distance. Thus, the resulting output from supposedly the same clustering algorithm differed between all three programs. The differences between the formulas were in the normalizing factor used. Cormack (1971) gave a weighted Euclidean distance measure where the weighting scheme is user specified. This actually provides for a large number of "Euclidean" distance measures.

Again, as with the selection of data units and variables, a researcher must select the index which is felt to retain the appropriate information in the raw data and represent the metric in which the cluster configuration is considered to be distributed. It should be noted that there is no guarantee that the researcher will select the most appropriate index. The researcher's conceptualization of cluster structure may not correspond to the cluster structure in the data, if any structure exists at all.

Clustering Algorithms

Most reviews of cluster analysis techniques attempt to provide some type of overall classification scheme for the methods themselves. Some type of classification scheme is virtually a necessity since even a cursory survey of the literature indicates that there are at least, at an absolute
minimum, over 200 clustering algorithms already in existence. Hartigan's (1975) book alone gives Fortran program listings for 82 subroutines which include clustering algorithms, data preparation, and data presentation. Many of Hartigan's algorithms are for special purposes or unique situations. Anderberg (1973) provided Fortran programs for about another 15 more general clustering algorithms along with several other subroutines for data preparation and presentation. Everitt (1974) provided mailing addresses for eight different program packages, one of which contains over a dozen different algorithms. Furthermore, a computerized literature search was conducted by the present author with the aid of the Mechanized Information Center at The Ohio State University during the period 1975-1976. The search indicated that new algorithms or major modifications of existing techniques were appearing at the rate of about 1 per month in the journals devoted to the physical and social sciences.

For the purposes of the present work, four major categories are identified and discussed. Although not all algorithms clearly fall into one category or another, most methods can be classified. The four categories include ordination techniques, hierarchical methods, partitioning algorithms (non-hierarchical methods), and clumping or clique procedures (overlapping cluster algorithms).
1. **Ordination procedures.** Although most social scientists are unfamiliar with the term ordination, psychologists have been primarily responsible for its development. The term ordination is more commonly used by biologists and statisticians (see Cormack, 1971; Johnson, 1968). Ordination techniques are those methods which attempt to provide some type of multidimensional representation of an input data set. Thus, techniques such as factor analysis (R and Q mode), and metric and non-metric multidimensional scaling fall into this category. Some of the earliest clustering procedures were based on Q-mode factoring techniques (Cattell, 1944, 1952; Goodall, 1954; Tryon, 1939). Torgerson (1952, 1958) introduced metric multidimensional scaling. Gray and Curtis (1957) applied Torgerson's procedure to a biological data set. More recent advances in non-metric multidimensional scaling have become popular due to the less stringent assumptions of scale type and metric (Kruskal, 1964; Shepard, 1962). Applications of the multidimensional scaling algorithms to cluster analysis have not escaped attention (Cormack, 1971; Kruskal, 1977; Mezzich, 1975; Rohlf, 1974).

One feature of ordination techniques is that a spatial representation of the entities in the data set is usually all that is produced by the algorithms. The actual determination or detection of clusters in the data is left up to the researcher's subjective judgment. Thus, the final
clustering of the data by the same algorithm can be different for two different researchers. This strongly subjective feature is usually considered the greatest weakness of the approach. In fact, Mezzich (1975), examined inter-rater reliability between classifications produced by ordination techniques and found an average correlation of only .67 across all data sets and methods. However, some authors have argued that the subjective aspect is also a strong point for the methods. Cormack has suggested that "When the data have not been forced into clusters, the observer can assess better whether clusters exist" (1971, p. 340).

Engineers working in the field of pattern recognition have also concluded that the human observer is a far better instrument with which to detect complex patterns of clusters than existing computer algorithms (Sammon, 1969). Sammon developed an interactive clustering program which displayed various spatial representations of the data on a CRT screen. The observer alters the representation in order to detect patterns or clusters in the data. Ball and Hall (1965) also developed an interactive clustering program (ISODATA) with the same view in mind, however, ISODATA is not an ordination procedure. Although not strictly an ordination technique, the procedure known as Chernoff's faces (1973) also uses human judgment to determine the clustering of the data.
It should be noted that not all researchers agree about the usefulness of the ordination approach. In a rather humorous but somewhat scathing review of the cluster analysis literature, Johnson (1968) has asserted that "There is no reason to assume, as is often done, that simple spatial models are particularly appropriate to taxonomic ordination" (p. 15).

2. **Hierarchical clustering procedures.** Perhaps the most popular clustering algorithms ever developed have been the agglomerative hierarchical clustering routines (see Anderberg, 1973; Cormack, 1971). The hierarchical methods have been particularly appealing to biologists since the hierarchical structure can be compared to evolutionary trees or hierarchies. Some authors have argued that the hierarchical structures are not appropriate for other disciplines (Aldenderfer, 1977; Cormack, 1971; but for counter-examples see Carroll, 1976; Johnson, 1967). The methods are called agglomerative because at the start of the clustering process each element forms a single cluster. As the process continues, entities are merged together to form larger and larger clusters. The end result is one cluster which contains all elements.

Agglomerative routines probably represent the oldest automatic clustering algorithms. Williams, Lambert, and Lance (1966) cite a reference on agglomerative hierarchical techniques by Kulczynski which dates back to 1927. However,
the more common methods were derived concurrently with the
development of the availability of high speed computers.
The well known complete linkage method can be traced back to
Sorensen (1948), while the single linkage and group average
methods date from Sneath (1957) and Sokal and Michener
(1958). (Mezzich (1975) cites an earlier reference for the
single link method which dates to 1951.)

There are today at least 15 to 20 agglomerative hier-
archical methods currently in use. Cormack (1971) lists
eight methods, Anderberg (1973) gives three additional
methods not listed in Cormack, and King (1967) provides one
more. Still other variants exist (see Carlson, 1972;
Clifford and Stephenson, 1975; McQuitty, 1967; and Sneath,
1966). The characteristics of these various methods have
been discussed in detail in the above references and also in
Lance and Williams (1967a).

Some of the more heated controversies in cluster analy-
sis occurred over which of the hierarchical methods is the
best algorithm or most appropriate in general. Some dis-
cussion occurred early within the biological community as to
whether weighted or unweighted averages should be computed
(Gower, 1967; McQuitty, 1966, 1967; Sokal and Michener,
1958; Sokal and Sneath, 1963).

A more lively discussion occurred between Jardine and
Sibson, representing the "Cambridge" school (Jardine and
Sibson, 1968a, 1968b, 1971a; Sibson, 1971), and the
"Australian" school led by Lance and Williams (Lance and Williams, 1966, 1967a; Williams, Clifford, and Lance, 1971; Williams, Lance, Dale, and Clifford, 1971). An interesting discussion of the controversy, which took place mainly in the Computer Journal, can be found in Aldenderfer (1977). In short, Jardine and Sibson attempted to provide a set of mathematical or axiomatic criteria by which to judge clustering algorithms. It turns out that the single linkage procedure was the only method which satisfied the criteria. Lance and Williams replied that different considerations based solely on the usefulness of the algorithms should be used. In general, the single linkage method tends to perform "poorly" on real-life data sets in that quite often just one very large cluster is obtained. This chaining characteristic led Lance and Williams to conclude that "We submit that nearest neighbor sorting [single linkage method] should be regarded as obsolete" (1967a, p. 377). Sibson (1971) replied to this attack by asserting that the system developed by Lance and Williams (1967a) "does not define a clustering method at all" (p. 156). The only valid exception is, of course, the single linkage method.

The controversy is related in a manner to observations made by Wishart (1969), and Everitt (1974). Both authors noted that virtually all clustering methods have severe variance restriction characteristics. That is, clusters found by most algorithms are minimum varianced and have
compacted structures which tend to rule out any elongation of the clusters. The single linkage method is one of the few algorithms which is capable of properly detecting elongated or oddly shaped clusters (see also Zahn, 1971). However, the single linkage method is apparently seriously affected by intermediates or outliers and has a tendency to chain together all data points in real life data sets. Lance and Williams were apparently objecting to this last characteristic.

The other possible type of hierarchical cluster formation involves the opposite process of agglomeration. However, few divisive methods have been developed and none have seen widespread use. Most divisive methods have been based on monothetic principles (splitting the data based on one variable at a time, versus polythetic methods which use all variables at once). Association analysis developed by Lambert and Williams (1962, 1966) is an example of such a technique. A few polythetic methods have been proposed and includes dissimilarity analysis by MacNaughton-Smith, Williams, Dale, and Mockett (1964). Hartigan (1975) also presented various types of splitting algorithms which need not form entire hierarchies.

3. Partitioning algorithms. Partitioning methods produce distinct non-overlapping partitions of the data. Often, the methods are known as non-hierarchical clustering procedures since only a single data partition is produced
(Anderberg, 1973; Lance and Williams, 1967b). This category is less structured than the hierarchical methods and also exhibits a greater variety of approaches. The techniques range in complexity from Hartigan's (1975) very simple leader algorithms to extremely intricate iterative reallocation methods such as Friedman and Rubin's method (1967). Blashfield (1977a) chose to discriminate between the methods using a set of five characteristics. The first characteristic involves the type of initial partition. Methods tend to vary in the choice of "seed points" or cluster centroids. Although a variety of techniques have been proposed, the K-means procedures usually use random data points as initial centroids (see Anderberg, 1973). ISODATA (Ball and Hall, 1965) selects points which are relatively distant from the overall data centroid. Wishart's CLUSTRAN allows for both of the above choices plus the ability for the user to specify a starting partition, possibly obtained from a previous data clustering (Blashfield, 1977a). Similarly, Wolfe's (1970) NORMIX program uses a partition obtained from an initial application of Ward's minimum variance hierarchical clustering algorithm.

The second dimension involves the type of pass through the data. The methods often operate iteratively on the data element assignments. The K-means procedures use nearest centroid assignment passes exclusively. That is, points are assigned to the nearest proposed cluster centroid. Other
programs such as MICKA by McRae (1971), and Friedman and Rubin (1967), use what are termed hill-climbing passes. Hill-climbing passes involve moving points from one cluster to another cluster in order to optimize a statistical criterion. Finally, the Friedman and Rubin method also uses forcing passes which change cluster membership lists regardless of the effect on the criterion statistic in an attempt to escape local minima.

The third dimension involves whether a statistical criterion is used by the algorithm. Both MICKA, and Friedman and Rubin's method allow the user to select statistical criteria such as $|W|$, trace $(W^{-1}B)$, and other possibilities. It has also been shown that the K-means procedure minimizes trace $W$, although the method was not specifically developed with this criteria in mind (Anderberg, 1973). Ball (1966) lists criteria used by several less well known algorithms.

A fourth dimension involves whether there is a fixed or variable number of clusters as the final output. Most procedures require the user to specify the exact number of clusters to be derived. At least one program (ISODATA), allows for a variable number of clusters in the final solution (Ball and Hall, 1965).
The final characteristic involves the treatment of outliers. K-means and NORMIX tend to force outliers to join the other clusters present. ISODATA and Lorr and Radhakrishnan's (1967) algorithm do allow for a residual pool of unassigned items.

Before considering the next category of clustering algorithms, it should be noted that one class of partitioning methods represents the only techniques based on statistical distribution theory. NORMIX or NORMAP developed by Wolfe (1970, 1971) are based on the assumption that the data values are distributed as a mixture of multivariate normal populations or samples.

4. **Clumping or clique formation methods.** The final category includes techniques which allow for the possibility of overlapping clusters. Some authors have argued that the often invoked non-overlapping criterion may be unusually restrictive in that some elements may be meaningfully assigned to more than one cluster (Jones and Jackson, 1967; Shepard, 1974). An early algorithm was introduced by Needham (1967) which produced overlapping structures. The algorithm was subsequently improved by Parker-Rhodes and Jackson (1969). Jardine and Sibson (1968a, 1971b; see also Cole and Wishart, 1970) provided an overlapping cluster algorithm which was based on an extension of the single linkage clustering method. Additional work on overlapping partition methods can be found in Peay (1975). Finally,
Shepard's ADCLUS (1974) generates overlapping cluster partitions using only binary input information (presence or absence data).

In general, the overlapping cluster algorithms have received the least amount of attention. Perhaps more attention should be paid to such techniques. The methods may be particularly useful in that they may possibly be able to be used as a test for the presence of a non-overlapping data structure in real life data sets. Hopefully, if such a structure really existed, then the overlapping cluster methods should logically produce non-overlapping solutions for such data. Otherwise, overlapping structures should be obtained as would be expected from less distinct structures or from random noise data.

The Problem and Necessity of Validating Clustering Procedures

Clearly, the clustering process is best represented by a series of subjective judgments which must be made by the researcher. Furthermore, there is no unified theory or commonly accepted statistical base which justifies any given approach. Of course, the lack of theory may be due primarily to a lack of a general definition of the term cluster. To complicate matters, virtually any clustering algorithm produces a partition solution for any given data set, even random noise data. Usually, the algorithm output gives no indication that the obtained solution was from a data set without structure. It is also likely that an improper
judgment by the researcher at any given step in the clustering process would produce an incorrect data partition from a data set which did possess some cluster structure. Several authors in the field of clustering have recognized these rather serious flaws in the methodology. Sneath has asserted that "It must be admitted that one of the biggest deficiencies of cluster analysis is the lack of rigorous tests for the presence of clusters and for testing the significance of those clusters that are found" (1969, p. 263). Likewise, Sammon also concurred by stating that "Perhaps the most serious deficiency involving present day clustering algorithms is that there do not exist really good ways for evaluating a resultant cluster configuration" (1969, p. 407). Unfortunately, Sneath and Sokal after raising the issue of whether a criterion could be developed for determining how successfully a group has been classified, concluded that "It is doubtful whether universally acceptable indices of optimality can be developed" (1973, p. 190). This is in line with their non-committal definition of the term cluster. Sneath and Sokal argued that certain requirements must be imposed on the nature of the classification wanted before meaningful criteria can be established.

A parallel issue is whether the methods themselves can recover true cluster structure when it is in fact present in a data set. There is no a priori guarantee that any given algorithm can recover true cluster structure under ideal
error free conditions. In addition, there is also no guarantee that the methods will be robust in regards to cluster recovery when the true structure is hidden by various types of error perturbation.

Ideally, analytic solutions to these questions would be desirable. Very few strictly mathematical comparisons of cluster analysis procedures have been reported. A few authors have focused on the properties of similarity measures alone (Cronbach and Gleser, 1953; Minkoff, 1965). When considering clustering algorithms themselves, few attempts have been made. Gower (1967) did attempt an analysis of three fairly simple methods. However, in general, the mathematical complexity becomes virtually overwhelming when the entire clustering process is examined and a variety of cluster structures are considered. For these reasons, most researchers have tended to use Monte Carlo methods. These reports have attempted to provide at least some validation information on various algorithms or procedures.

**Simulation Procedures**

A series of studies have used the multivariate normal mixture model as the basis for the conceptualization of population clusters in a multidimensional space. This class of studies generated mixtures of multivariate normal samples and analyzed the resulting data sets by a variety of clustering methods. Kuiper and Fisher (1975) examined six hierarchical clustering methods which included the complete
linkage, the single linkage, the group average method, the median method, the centroid method, and Ward's minimum variance method. The data sets contained two to five clusters each representing samples from multivariate normal populations. Each constructed set was then analyzed by the six clustering algorithms. The results indicated that Ward's minimum variance procedure had about the best recovery rate. Gross (1972) also generated multivariate normal data sets. However, only Ward's minimum variance method was tested. Nevertheless, the recovery rates correspond well to the Kuiper and Fisher results. Another study by Blashfield (1976) examined the complete linkage, the single linkage, the group average, and Ward's method with multivariate normal mixture samples. Again, the results indicated that Ward's procedure gave the best recovery rate. The complete linkage method placed a close second, the group average method placed third, while the single linkage method had the poorest recovery rate. The single linkage method in fact exhibited an extremely poor recovery rate.

Everitt (1974) used multivariate normal data sets in two dimensions in order to illustrate the recovery characteristics of a series of clustering algorithms. However, Everitt's results indicated that the single linkage method was the only technique which could correctly recover two
well separated elongated multivariate normal clusters. Thus, the recovery ability of the methods seem to be strongly affected by the types of clusters present in the data.

Blashfield (1977a) also used 20 multivariate normal data sets in a pilot study designed to examine four non-hierarchical clustering algorithms with two variations on each method. Results indicated that Rubin and Friedman's method using $|W|$ as the minimization criterion gave the best recovery rate.

Finally, Carroll and Field (1974) used a univariate normal error model in their study of 10 similarity measures. The authors varied the error level of the generated data along with the elevation, scatter, and profile shape. The authors concluded that Cattell's $r_p$ index and Osgood's and Suci's D measure were the most useful coefficients.

However, Milligan and Isaac (1978) have argued that the multivariate normal cluster model is not necessarily the most reasonable method by which to generate the constructed data sets. All of the clustering methods examined above produced distinct non-overlapping partition solutions. However, except for the data sets generated by Everitt in two dimensions, the multivariate normal data sets may have consisted of overlapping cluster structures. Hence, the data sets did not necessarily provide the most appropriate test for the non-overlapping cluster algorithms. In fact, considerable information from the studies themselves indicated that cluster overlap did occur in the constructed data sets.
Milligan and Isaac took a different approach in the construction of the test data sets. Milligan (1978b) showed that the hierarchical solution from all four of the methods examined by Blashfield (1976) invoked on the data a type of metric known as the ultrametric (see Johnson, 1967). Thus, as a test of the recovery behavior of the four hierarchical methods, Milligan and Isaac generated 672 data sets which satisfied the ultrametric inequality. The data sets were constructed in such a manner that distinct non-overlapping clusters which exhibited the properties of internal cohesion and external isolation were present.

The results from the Milligan and Isaac study produced a rank order of the recovery behavior of the algorithms which was quite different from the multivariate normal mixture studies. The group average method had the best recovery rate while the complete linkage procedure placed second. Ward's minimum variance method placed a poor third and the single linkage method placed last. An examination of the data showed a strong interaction between cluster separation level and error level. The methods generally exhibited high recovery rates in conditions of low error perturbation and wide cluster separation. High error levels combined with small between cluster separation made it particularly difficult for the methods to recover the true cluster structure. A more detailed examination of the data showed that the single linkage method was extremely sensitive to the error
perturbation of the data. The single linkage method exhibited perfect recovery in all error free data sets. However, a substantial decrement in performance was seen for fairly low error levels where the other methods were not seriously affected. A final interesting result was that Ward's method, found to be the best method in the multivariate normal mixture studies, actually exhibited a marked drop in the recovery rate with the error free data sets at the smallest level of between cluster separation. Ward's method was the only one of the four methods which failed to give perfect recovery in the error free conditions. As it turns out, the error free clusters generated at the smallest level of between cluster separation were not necessarily minimum varianced. Thus, Ward's method was using a criterion which was inappropriate for the data.

Milligan and Isaac (1978) concluded by noting that when considering all of the foregoing validation studies, the four clustering methods appeared not to be very robust with respect to differing metrics or differing cluster structures.

Other definitions, metrics, or types of cluster structures are possible. Cunningham and Ogilvie (1972) generated six example data sets based on a geometric Euclidean model. Seven hierarchical clustering methods were used to analyze each data set in a pilot study designed to illustrate the characteristics of the methods. Milligan (1978a) also used a Euclidean model in examining the recovery characteristics
of the Hubert (1972) goodness-of-fit cluster statistic. Although Everitt (1974) used a bivariate normal distribution model for his data sets, the resulting cluster configuration possesses a strong geometric interpretation. Real life data sets with clear geometric structure have also been used. Forgy (1965) used a bivariate data set from the field of astronomy to illustrate problems with elongated data sets which most clustering algorithms have. Fisher's (1936) classic iris data set also seems to possess geometric cluster structure. Friedman and Rubin (1967) and Mezzich (1975) have both used the iris data set to validate various clustering algorithms. It appears though, that more comprehensive studies using a Euclidean or geometric conceptualization of cluster structure have not been undertaken. Apparently, a major difficulty in such a study would be to define the cluster configurations in a generally acceptable manner.

Still other ways or methods of constructing data sets have not escaped attention. Bromley (1966), in a discussion of rank order cluster analysis, used Thurstone's (1947) famous box problem data set. The box data set has often been used as an illustrative data set for factor analytic procedures. Helmstadter (1957) also generated geometric figures (spheres, cylinders, tetrahedrons) in an examination of 12 different similarity measures. Another artificial data set was generated by Bartko, Straus, and Carpenter (1971). The data set consisted of 100 factitious archetypal
psychiatric patients and was used to validate three clustering algorithms. Mezzich (1975) used a similarly derived artificial psychiatric data set in his validation study. Even random noise data sets have been used to examine the properties of clustering algorithms (Sneath, 1966).

Real life data sets have also been used to validate and test various clustering procedures. Perhaps one of the most extensive studies was conducted by Mezzich (1975). Mezzich considered 10 clustering algorithms and three real life data sets. Mezzich concluded that the complete linkage method gave the best overall recovery rate.

However, one drawback to this general approach is that only a very small number of data sets are examined. Furthermore, it is unlikely that several real life data sets could be gathered which objectively manipulate such factors as the number of clusters, inter-cluster spacing, relative cluster sizes, and other variables which might be of interest.

It should also be noted that most of the research projects have focused on the recovery of error free data sets, especially for the studies using constructed data sets. Blashfield (1976) did slightly error perturb all interpoint distances before clustering the data sets. The study by Milligan and Isaac (1978) was one of the few studies which systematically varied the error perturbation of the distances.
In regards to the real life data set validation approach, researchers have examined such effects as applying different similarity measures or selecting only a subset of the variables or entities (Green and Rao, 1969; Grigal and Arneman, 1969; Mezzich, 1975). However, this raises an interesting issue. Within the field of cluster analysis, there is not just one form of error perturbation, but several different ways in which error perturbation can be introduced and examined.

**Types of Error Perturbation**

Certainly, it seems necessary to show that any proposed clustering method can recover structure in error free data sets. However, Baker (1972) has argued that it is also necessary to verify that the methods have the ability to extract a known data structure when the structure is hidden by noise. Kanal (1972) has argued that outliers or intermediates should be removed before clustering the data, although no method for identifying the outliers is given for general application. Thus, one type of error perturbation in a clustering situation is the presence of outliers in a data set. Lange, Stenhouse, and Offler (1965) have indicated that small percentage changes in the similarity values can affect the final cluster structure greatly. This, of course, is the same as the error perturbation of the distances. Baron and Fraser (1968), and Gower (1969) have argued that the researcher should remove noisy or irrelevant
variables from the analysis before the similarity coefficients are computed. Hence, the introduction of a random noise dimension represents another form of error perturbation.

Baker (1972) has noted that most clustering algorithms seem to be sensitive to the similarity coefficient used. Blashfield (1977b), Grigal and Arneman (1969), and Mezzich (1975) have also noted this result. Thus, the use of various types of similarity coefficients, especially when they are not appropriate for the data set and its cluster structure, represent another form of error perturbation. Related to the issue of similarity coefficients is the problem which a researcher often faces in regards to the standardization of the variables in a data set. If the variables in a data set have differing means and variances, then the variables will receive differential weighting when computing most similarity coefficients. Such variable weighting may be inappropriate. The common solution used is to standardize the variables to standard score form before computing the similarity coefficient. However, both Fleiss and Zubin (1969) and Anderberg (1973) have noted that the procedure can distort the true cluster structure. The standardization should be carried out within cluster, but of course, cluster membership is not known on an a priori basis. Again, it appears that variable standardization reflects another source of error perturbation.
Of course, this does not exhaust the list of possibilities. Another source of error perturbation can occur when a researcher specifies that an incorrect number of clusters are to be recovered. The researcher may also use a clustering method which is not suited to the type of cluster structure present.

Thus, the overall clustering process represents a very complicated methodology. The potential types of error perturbation which can result are also numerous and complex.

**Validation Measures**

In general, validation measures can be divided into two overall categories or approaches as proposed by Sneath (1969). Validation implies that some type of criterion is invoked as a standard of judgment. The first procedure has been labeled internal criterion analysis. The internal criterion approach involves using only the information available from the input data matrix and the resulting partition solution. Typically, some statistic or index is computed between the input data and the proposed cluster partition. Thus, the proposed indices tend to be measures of the goodness-of-fit of the cluster configuration to the structure of the original data. It should be noted that a clustering algorithm can produce a partition solution which fits the input data well, but there still need be no distinct cluster structure present in reality.
The second procedure, called external criterion analysis, involves using a source of information separate from the input data and the partition solution. The additional source of information is often an a priori clustering of the data. Again, some type of index or statistic is computed between the criterion definition of cluster structure and the obtained clustering of the data. The external criterion approach has the potential of providing possibly the strongest test information in regards to the ultimate usefulness of any clustering procedure. If the input data set consists of constructed (artificial) data with known cluster structure, then the criterion clusters can be defined as the known item assignment in the generated clusters. If the sets are constructed in such a manner that virtually any researcher would agree that any given clustering method should recover the true cluster structure present, then the consistent failure of the algorithm would indicate that the method should probably not be used in general applications. It should also be noted that the external criterion approach can be used to evaluate the effects of error perturbation of constructed data sets in terms of cluster recovery for any given algorithm. Since internal criterion analysis utilizes the input data as the criterion, any internal index would indicate the goodness-of-fit of the solution partition to
the error perturbed data, not the true data structure hidden by the error. The external criterion would measure only the fit to the underlying cluster structure.

Several researchers have argued that significance tests for cluster analysis results should be developed (Goodall, 1966; Sneath, 1967). The procedure would be designed to test for the presence of cluster structure against the null hypothesis of no cluster structure. There are currently no widely accepted testing procedures. If a satisfactory significance testing procedure is ever developed, the procedure will probably be based on some type of internal criterion. Tests based on external criteria can be developed if such criteria are available (Hansen and Milligan, 1978). The Hansen and Milligan procedure assumes that an appropriate external criterion can be collected. For example, in marketing research, one useful way to validate clusters generated by demographic data is to use as a criterion the yearly income of the family units in the study. If the clustering of the data is to be useful for marketing purposes, then variation in the average income should be detected.

However, it is fairly uncommon to have a strong external criterion. The advantage of the internal criterion approach is that it requires no additional information besides the input data. A few attempts have been made to develop appropriate test statistics. Engelman and Hartigan (1969) provided tables for testing $F$-type ratio statistics
for a monothetic (one variable) clustering algorithm which produced binary splits only. Hartigan (1975) proposed the F-test as a test for the K-means clustering procedures. Beale (1969) also proposed the F-test on the assumption that clusters are spherical normal. Baker and Hubert (1975) proposed using the Goodman and Kruskal (1954) $\gamma$ statistic as a single partition recovery measure. However, the use of the $\gamma$ statistic as a recovery measure has been critized by Milligan (1978a). For algorithms which produce hierarchical solutions, some attempts have been made to provide testing procedures based on random graphs (Ling, 1973). Finally, Rohlf and Fisher (1968) generated null distribution values for the cophenetetic correlation coefficient. However, few of the testing procedures are generally applicable and none have been shown to possess satisfactory operation characteristics.

**Internal Criterion Measures**

An amazingly large number of internal criterion indices have been proposed in the clustering literature. Rohlf (1974) provided a listing of many, but not all, of the measures. Perhaps the earliest index proposed was the so-called cophenetetic correlation developed by Sokal and Rohlf (1962). The index is based on the Pearson product moment correlation coefficient. The correlation between the input data matrix and a dissimilarity matrix derived from the clustering process is computed. The derived matrix can
be defined in a number of ways. The most common definition has involved hierarchical clustering procedures where the dissimilarity matrix entries are obtained from the level values of the hierarchical classification. When a single partition is of interest, a 0/1 dissimilarity matrix has been used to indicate whether any specific pair of items are clustered together in the algorithm solution. Finally, the derived matrix may actually consist of distances between points in a configuration obtained from an ordination procedure. The resulting coefficient would correspond to the square root of the index of metric determinacy as used in the multidimensional scaling literature (Isaac and Poor, 1974). Farris (1969) has criticized the cophenetic correlation and provided simple illustrations of shortcomings of the index. However, the situations where Farris argues that the index fails are not necessarily common in real life data sets. Farris does show that the cophenetic correlation is maximized by the group average clustering method.

Rank order variants of the cophenetic correlation have not escaped attention. In fact, Johnson (1967) suggested that a hierarchical routine should be developed which maximizes the rank order correlation between the input data and the hierarchical classification. The rank order correlation is appealing since it allows the researcher to drop the constraint that a strictly linear relationship exists between the input data and the cluster solution matrix. Also,
social science researchers are often hard put to argue that
their measurements possess anything higher than ordinal
scale properties. This issue of scale type has not been a
serious problem for the biological sciences since much of
their data consists of physical measurements and hence of
interval or ratio scale type. When biologists deal with nom-
inal scale data (presence-absence data), they pay particular
attention to the selection of the similarity coefficient.

Baker and Hubert in a series of articles (Baker, 1974;
Hubert, 1974) used the Goodman and Kruskal (1954) $\gamma$ statistic
as a rank order hierarchical cophenetic correlation coeffi-
cient in their evaluation studies of the complete and single
linkage algorithms. Baker and Hubert (1975) used the $\gamma$ sta-
tistic as a single partition cophenetic coefficient. As it
turns out, when $\gamma$ is defined as a single partition measure,
the statistic shares a simple mathematical relationship to
Hubert's (1972) goodness-of-fit cluster statistic. As noted
above, when the $\gamma$ statistic is used as a single partition
index, the measure has poor operating characteristics in
regards to the detection of true cluster structure (Milligan,
1978a).

Jardine and Sibson (1968a, 1971b) have introduced a
number of coefficients for the comparison of two dissimilar-
ity matrices. One series is based on taking some measure of
the actual distance between the entries in the two matrices.
The indices range from taking the sum of the absolute
differences, to computing a Euclidean distance measure, to computing a normalized distance measure. The indices are not suited for use as single partition (non-hierarchical) measures. Hartigan (1967) has also suggested the use of a weighted Euclidean distance measure as an internal criterion. However, Hartigan does leave open the question of what weighting scheme or formula to use. Anderberg (1973) has argued in favor of an equal weighting formula. However, an equal weighting scheme reduces the Hartigan index to one of the Jardine and Sibson measures. Sammon (1969), on the other hand, suggested using a weighting function proportional to the size of the distance involved. Finally, Jardine and Sibson have also proposed the use of at least two additional indices which are not strict distance measures. One index is the simple angular separation coefficient between vectors while the other measure is a log function coefficient which is based on maximum distance ratios for the two solutions. It is interesting to note that Jardine and Sibson (1971b) acknowledged the existence of the Pearson based cophenetic correlation. They assert that the index is ad hoc, though they do admit that it is widely used.

Besides the variety of coefficients developed specifically for the assessment of goodness-of-fit, several clustering methods themselves utilize criteria that are also possible alternative internal criterion measures. The most common index is a measure of the within groups sum of
squares. The Edwards and Cavalli-Sforza (1965) clustering algorithm specifically attempted to minimize the within group sum of squares by considering all possible partitions in a data set, a formidable task indeed. A less ambitious scheme was introduced by Ward (1963) which attempted to minimize the increase in the within group sum of squares at each step in the hierarchy. Perhaps the most elaborate procedure is the algorithm developed by Friedman and Rubin (1967) which attempted to minimize or maximize various forms of the B matrix (between groups sum of squares and cross products matrix) and the W matrix (within groups sum of squares and cross products matrix). Finally, at least one group of researchers (Jardine and Sibson, 1971b; Zahn, 1971) have argued that the minimum spanning tree solution produced by the single linkage algorithm is the most appropriate criterion.

**External Criterion Measures**

Before considering the indices proposed as external criteria, the various sources of external information should be noted. The information can range from very reliable and extremely accurate to incorrect and unreliable. A "strong" source of information would be the cluster membership lists from constructed data sets with known cluster structure. Several studies noted earlier in the paper have used this source of information for testing purposes (Blashfield, 1976, 1977a; Everitt, 1974, Gross, 1972; Kuiper and Fisher, 1975;
Milligan and Isaac, 1978). Weaker forms of external information has involved the use of "expert" judgments of the cluster structure as in real life data sets. The primary difficulty with the expert judgment approach is that the experts may be wrong about the cluster structure. There may, in fact, be no cluster structure present in the data set under examination, or it may be incorrectly perceived. Nevertheless, the expert judgment approach has been one of the most popular procedures. A review of the clustering literature shows that the introduction of just about every clustering algorithm has been accompanied with an illustrative application on a real life data set. This often serves as the only validation information. For examples, see Bromley (1966), Boyce (1969), Freidman and Rubin (1967), Goldstein and Linden (1969), Hodson, Sneath, Doran (1966), Johnson (1967), Lorr and Radhakrishnan (1967), and Ward and Hook (1963). However, the use of expert judgments may be more appropriate for fields such as the biological sciences than for the social sciences. Biologists can bring to a clustering situation a considerable amount of valid a priori information (evolutionary theory, dissection information, breeding results). Social scientists generally do not have such well developed data theories.

Other types of external criteria have also been proposed. One popular strategy has been to use the output of one clustering algorithm as the criterion for another
algorithm, either on the same data set, or on split portions of a data set (Anderberg, 1973; Blashfield, 1977a; Cormack, 1971; Grigal and Arneman, 1969; Mezzich, 1975). Such cross validation procedures yield limited information. Even if the algorithms agree, there is no guarantee that a correct clustering of the data has been obtained from either algorithm. If the algorithms disagree, there is no way of telling which algorithm produced an incorrect partition.

Assuming that a criterion partition is provided by some source, a number of external criterion statistics are possible. One broad class of external criterion measures are based on a contingency table approach. The basic matrix for the measures consists of a cross-classification table where the row categories correspond to the criterion clusters and the column categories to the obtained clusters. Entities are classified according to which partition they fall in for both the criterion partition and the obtained solution partition. All of the proposed measures were originally developed for the analysis of cross-classified data and have been adopted for clustering applications. None of the indices were developed for the purpose of cluster analysis validation. Thus, many of these indices have serious flaws when used for clustering applications. Although several papers have suggested the indices (Anderberg, 1973; Borko, Blankenship, and Burket, 1968; Mezzich, 1975), none of the measures have seen widespread use. The index which has been reported
the most often appears to be the simple percentage of concordance between the criterion and obtained clustering partition.

Several chi-square based measures have been suggested, including the chi-square test of independence. However, the independence test is only indirectly related to the assessment of correct cluster recovery. A chi-square value which is clearly within the null distribution would indicate that the resulting cluster classification did not exceed chance assignment levels. However, a highly significant chi-square may or may not indicate correct cluster recovery. A significant value can easily be obtained in ways which do not correspond to a diagonal pattern that would be obtained with a perfect match between the criterion and cluster partition. This problem also plagues the other chi-square based measures. These alternative measures include Pearson's coefficient of contingency, Cramer's V statistic, and Goodman and Kruskal's $\lambda$ coefficient (see Anderberg, 1973). Pearson's index has a range of 0.0 to 1.0, but the index cannot always achieve its upper bound. In fact, for a value of 1.00 to be obtained, the number of clusters in both partitions must be infinite (a rare event indeed). Furthermore, Goodman and Kruskal (1954) argued that the values obtained from the Pearson index cannot necessarily be compared from one table to the next. Cramer's V statistic is very similar to Pearson's except for the use of a different normalizing factor.
The Goodman and Kruskal $\lambda$ statistic also has a range of 0.0 to 1.0, and has the advantage of being interpretable as the proportional reduction in error when predicting column membership from row membership. Here again, large index values can be obtained for tables which actually indicate a poor match between the criterion and the cluster solution partition.

A final contingency table based index should be noted. Cohen's (1960) kappa statistic has been used by Blashfield (1976, 1977a) in his validation research. Kappa was designed as a measure of the agreement between two nominal scale variables. The measure has an upper limit of 1.0. Kappa is basically an indication of the deviation of the expected proportions in the diagonal cells from the observed proportions. However, ambiguity exists as to how to determine which cells are the appropriate "Diagonal" entries. This, of course, stems from the fact that there is no objective way of matching the cluster partitions from the criterion set to the solution set. Thus, the kappa statistic is not unique. For a two cluster solution, two values are possible for the index. For a three cluster solution, six alternative values can be obtained. In general, there are $N!$ possible values where $N$ is the number of clusters. Blashfield chose to take the largest numerical value when he used
the index for his clustering validation studies. Blashfield does admit that the procedure could tend to overestimate the recovery ability of the algorithms.

A completely different conceptualization for an external criterion index was developed simultaneously by Green and Rao (1969) and Rand (1969, 1971). The Green and Rao measure is simply the complement function of the Rand statistic. The index is one of the few measures which have been directly developed for use in clustering situations. The index is quite logical, and in fact, was once independently derived by the present author before he was aware of the previous research. The statistic is based on considering the joint classification of pairs of points in the criterion definition of cluster structure and the obtained solution partition. A square matrix is defined whose rows and columns correspond to the elements which have been clustered. All entities in the matrix are either 0.0 or 1.0. If two data units are assigned to the same cluster in both the criterion solution and the obtained partition, then the corresponding entry in the matrix is assigned a value of 1.00. A value of 1.00 is also assigned if the two data units are assigned to different clusters in both partition solutions. A value of 0.0 is assigned otherwise. Thus, a value of 1.0 indicates agreement between the criterion definition of cluster structure and the obtained solution. A zero entry indicates disagreement. The computation of the
statistic involves simply summing over the entries in the matrix and dividing by the total number of elements in the matrix. (Actually, only a lower or upper half matrix need be computed. The diagonal is left undefined.)

The Rand index has several distinct advantages over the other proposed external criterion measures. The index has a range of values from 0.0 to 1.0 with a value of 1.00 occurring only when the observed cluster partition exactly matches the criterion definition. The index is unique for any given criterion and obtained solution. Thus, the index will not overestimate the agreement between partitions. Finally, the statistic also can be used to determine the number of pairwise misclassifications obtained in any solution by simply working backwards and solving for the numerator value.

**Objectives of the Present Study**

The basic aim of the present study was to evaluate the recovery behavior of a set of clustering algorithms over a series of constructed data sets which have been subjected to various types of error perturbation. The study was designed to simulate the actual usage of the algorithms rather than attempting to modify the procedures. The clustering algorithms selected produced distinct nonoverlapping partitions in an objective manner. The data sets were generated in a geometric space where each dimension was assumed to be a continuous variable of interval scale type. The clusters
contained in the constructed sets exhibited the properties of internal cohesion and external isolation. The location of each point was determined with the aid of a random number generator. For purposes of simplicity in data preparation and for ease of presentation, the Euclidean distance metric was used as the basic distance measure defining the sets.

Six different aspects of error perturbation were examined in the study. Each type of error perturbation was examined separately. That is, in general the perturbation conditions were not compounded in the same data set, although variable standardization did occur in more than one condition. The first three conditions represented inherent error states of the data over which the experimenter usually has no control. The three conditions were:

1. **Pure data with no error present.** This condition provided a test of the clustering algorithms in an ideal error free situation in order to determine if they could recover the true cluster structure in the data. A high failure rate in this condition for any given method would indicate a rather serious deficiency of the algorithm in regards to its general application. The zero-error data sets generated in this condition formed the basic parent data sets used for the remaining five conditions.

2. **Data sets with outliers added.** Each of the parent data sets had a number of additional points added which fell outside existing cluster boundaries. The data sets
allowed for an examination of the recovery performance of
the algorithms when intermediates or outliers diffused the
true cluster structure.

3. Error perturbation of the distances. Each of the
true interpoint distances in the parent data sets were error
perturbed to form a new distance matrix. The condition
corresponded to the situation where a true cluster structure
existed but the measuring instruments were imperfect.

The last three conditions represented the introduction
of error into the data which can occur when the researcher
makes certain errors of judgment when selecting and prepar-
ing the data matrix for analysis. The conditions were:

4. Introduction of random error dimensions. Each of
the parent data sets had one or more pure error dimensions
added to the existing set of variables which defined the
cluster structure. Since the researcher usually has full
control over the variables which are entered into the dis-
tance computation, the introduction of a pure error dimen-
sion simulated those situations where a variable is used in
the clustering process but is irrelevant to the true clus-
tering in the data.

5. Computation of the distances with a non-Euclidean
index. Instead of computing the distances with the Euclid-
ean distance formula, two different coefficients were used.
Cattell's (1949) \( r_p \) index was used as one of the coeffi-
cients. The index retains both profile shape and height
information. The Pearson correlation coefficient was used as the other index. The correlation coefficient retains only profile shape information. It should be noted that both indices standardize the variables either between or within subjects before or during computation of the coefficient. The two similarity indices were converted to dissimilarity coefficients for the study. The use of an incorrect dissimilarity measure simulated a possible error which a researcher might make when reducing a set of variable for input to a clustering algorithm.

6. Standardization of the variables. Each dimension of each parent data set was separately standardized to have a mean of zero and a variance of 1.00 before computation of the Euclidean distances. The process simulated the situation where a researcher is often required to standardize a data set when the measured variables have differing means and variances.

The primary measure of cluster recovery was the Rand (1971) statistic. The index is an external criterion measure which can be computed for both hierarchical and non-hierarchical clustering methods. The index provided a comparable unit of measurement of cluster recovery across data sets and methods. The statistic was computed at the appropriate partition level for the hierarchical algorithms. In order to provide base line rates of the statistics for each method, sets of pure random noise data were generated and
analyzed by each of the clustering procedures. Values for the Rand statistic were computed by defining arbitrary partitions in these random data sets. Thus, it was possible to determine if the recovery rate of any given method exceeded that which would have been expected by chance alone.

An internal criterion measure of cluster recovery was also taken. The Pearson correlation was computed between the input distance matrix and a dissimilarity matrix whose entries were either zero if the corresponding elements belonged to the same cluster as determined by the algorithm or one otherwise. The 0-1 dissimilarity matrix was defined at the appropriate partition level for the hierarchical methods. The measure represented one form of the so-called cophenetic correlation coefficient. The index provided a comparable measure across data sets and across both hierarchical and non-hierarchical clustering methods. The correlation reflected the goodness-of-fit of the cluster partition as determined by the algorithm to the input distance data. This form of the cophenetic correlation considers only one partition level when a hierarchical solution is present.

For the hierarchical clustering methods an additional internal criterion index was measured. The Pearson correlation was computed between the input data matrix and the distance matrix derived from the hierarchical classification.
The statistic is the traditional form of the cophenetic correlation coefficient. This form of the index provided a measure of the recovery of the entire hierarchy structure rather than just focusing on a specific level in the hierarchy. A comparable cophenetic correlation obviously could not be computed for the non-hierarchical methods.

Base line or null distribution values for both types of cophenetic correlations were computed from the random noise data sets. Significance tests and other comparisons were conducted to determine if either statistic could be used to detect correct cluster recovery.

Finally, as a measure of the degree of distortion produced by the various error conditions, the Pearson correlation between the error free parent data matrices and the resulting error perturbed matrices was computed. The coefficient allowed for a comparison across conditions as to the relative severity of the error perturbation introduced into the data.
II. METHODS AND MATERIALS

Date Generation and Error Perturbation Process

Date Generation

Random number generator. The random number generator used throughout the study was the LLRANDOM generator developed by the Naval Postgraduate School at Monterey, California (see Learmonth and Lewis, 1973). The generator was extensively tested by Dudewicz (1974) at the Ohio State University Instructional and Research Computer Center along with eight other random number generators. The research by Dudewicz indicated that the LLRANDOM generator produced pseudo-random numbers of fairly high quality. The generator was also one of the fastest in terms of central processing unit time. The generator is written in Assembler with two Fortran subroutines and a listing of the program with sample output is given in Appendix A.

Defining the clusters in the multivariate space. The constructed data sets which were generated for the present study were defined in a N-dimensional geometric space. The distance metric used was the Euclidean distance function. The constructed data sets contained clusters which exhibited properties of external isolation and internal cohesion. The cluster boundaries were determined separately for each
dimension of the space in such a manner that cluster overlap was not possible on the first dimension. For the remaining N-1 dimensions, the cluster boundaries were determined in such a manner that cluster overlap was possible. Thus, the cluster generation process ensured that all cluster boundaries closed off finite regions of space and that the clusters exhibited at least some minimal level of separation. The cluster generation process also allowed for a fairly wide variety of cluster configurations in the N-dimensional space. Although all cluster boundaries were parallel to the dimensions of the space, it should be noted that Euclidean distances are invariant to rigid rotations of the dimensions. The generation procedure was adopted for two reasons. First, data sets constructed in such a manner provided a fairly clear-cut test of the ability of the clustering algorithms in recovering true cluster structure. Secondly, the procedure offered no major programming difficulties.

**Characteristics of the constructed data sets.** One hundred eight data sets were generated as the basic error free parent data sets for Condition 1. The number of objects to be clustered was set at 50 for all data sets. The sample size was chosen as being somewhat representative of real life applications. There was also no reason to believe that the clustering methods which were selected for the study were sensitive to differential sample sizes.
For the purposes of the present study, the absolute cluster boundaries were determined with the aid of a uniform distribution random number generator. The uniform generator provided a convenient procedure which created clusters of arbitrary volumes and more importantly guaranteed that a specific level of between cluster separation was achieved. The actual location of the data points within clusters were determined with the aid of a truncated multivariate normal distribution generator. A separate multivariate distribution was used for each cluster and the centroid of the distribution was located at the center of the cluster. The procedure generated a point distribution within clusters which was considered intuitively acceptable. That is, the center of each cluster possessed the greatest point density while the point density fell off towards the boundaries of the cluster.

A factorial design was used to control the generation of three aspects of cluster structure. The three factors were completely crossed and were included in order to provide some generalizability over the characteristics in the overall experiment. Thus, a comprehensive analysis of the effects on the algorithms of these factors was not undertaken. The first of the factors involved the number of clusters in the data sets. The number of clusters was set as either 2, 3, 4, or 5 in each constructed data set. The levels were selected as representing a parsimonious
reduction of the total sample size and researchers often seek solutions with a very few number of clusters. The second factor involved selecting the number of dimensions of the space from the set of 4, 6, or 8. The values are representative of the number of variables which are often used in real life research settings. Also, the larger dimension sizes chosen could not be visually inspected with ease. Finally, three levels of relative cluster sizes were chosen to represent the distribution of the elements across the clusters. One level involved creating equal size clusters, or at least as close to equality as could be established. The second level involved placing 60% of the points in one cluster and distributing the rest equally among the remaining clusters. The third possible condition placed 10% of the data points in one cluster and the remainder were again equally distributed across the rest of the clusters in the set. The resulting point distribution is presented in Table 1 for the various combinations produced by the number of clusters and point distribution conditions. The 10% distribution condition allowed for a fairly discrepant distribution of points in the smaller number of clusters but only a slight discrepancy for the larger number of clusters. The 60% condition had a reversed effect as seen in the table.

Thus, the three factors generated a total of 36 cells. Three observations per cell were collected to give a total sample size of 108 data sets.
Table 1

Point Distribution Across Clusters

<table>
<thead>
<tr>
<th>Number of Clusters</th>
<th>Density Level</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Equal Density</td>
<td>10% Density</td>
<td>60% Density</td>
</tr>
<tr>
<td>2</td>
<td>25-25</td>
<td>5-45</td>
<td>30-20</td>
</tr>
<tr>
<td></td>
<td>(5-5)</td>
<td>(1-9)</td>
<td>(6-4)</td>
</tr>
<tr>
<td></td>
<td>(10-10)</td>
<td>(2-18)</td>
<td>(12-8)</td>
</tr>
<tr>
<td>3</td>
<td>16-17-17</td>
<td>5-22-23</td>
<td>30-10-10</td>
</tr>
<tr>
<td></td>
<td>(3-3-4)</td>
<td>(1-4-5)</td>
<td>(6-2-2)</td>
</tr>
<tr>
<td></td>
<td>(6-6-8)</td>
<td>(2-8-10)</td>
<td>(12-4-4)</td>
</tr>
<tr>
<td>4</td>
<td>12-12-13-13</td>
<td>5-15-15-15</td>
<td>30-6-7-7</td>
</tr>
<tr>
<td></td>
<td>(2-2-3-3)</td>
<td>(1-3-3-3)</td>
<td>(6-1-1-2)</td>
</tr>
<tr>
<td></td>
<td>(4-4-6-6)</td>
<td>(2-6-6-6)</td>
<td>(12-2-2-2)</td>
</tr>
<tr>
<td>5</td>
<td>10-10-10-10-10</td>
<td>5-11-11-11-12</td>
<td>30-5-5-5-5</td>
</tr>
<tr>
<td></td>
<td>(2-2-2-2-2)</td>
<td>(1-2-2-2-3)</td>
<td>(6-1-1-1-1)</td>
</tr>
<tr>
<td></td>
<td>(4-4-4-4-4)</td>
<td>(2-4-4-4-6)</td>
<td>(12-2-2-2-2)</td>
</tr>
</tbody>
</table>

NOTE: The parenthetical entries indicate the number of outliers assigned to each cluster for Condition 2. The first line gives the number of added elements for the 20% outlier level while the second line gives the number for the 40% level.
The maximum range of each cluster on each dimension was chosen randomly from the interval of 10 to 40 units. Cluster overlap was not permitted on the first dimension. Overlap was permitted on the remaining dimensions of the space. The length of a cluster on each dimension was used to define the standard deviation of the points to be generated within the cluster for the dimension. Specifically, the length of the cluster on each dimension was defined to be three standard deviations. In order to generate the actual location of the data points within the clusters, a truncated multivariate normal distribution model was used. The centroid of the multivariate normal distribution was located at the geometric center of the cluster in the N-dimensional space. The variance-covariance matrix was defined as a diagonal matrix where all off diagonal elements were zero. The variances on the diagonal were set equal to the squared standard deviations computed for each dimension of the cluster. A truncated distribution was used to ensure that all cluster points fell within the cluster boundaries. Any point which had coordinate values which exceeded +/- 1.5 standard deviations on any dimension was rejected and another random location was examined. Also, since all covariances were zero, the observed values on all dimensions were independent. Thus, a single univariate normal distribution random number generator was actually used to generate the coordinates on all dimensions.
On the dimension where cluster overlap was not permitted, the clusters were separated by at least the sum of the values of one-fourth standard deviations from each of the neighboring clusters. (Note that the clusters need not have had the same variances.) The absolute boundaries of the clusters were used to define the separation between clusters. That is, at 1.5 standard deviations from the cluster centroid the next cluster boundary which was also 1.5 standard deviation from its respective centroid could not occur for at least the distance of the sum of one-fourth standard deviations from each cluster. The actual distance between the closest points from each cluster was usually greater than the sum since the points would not generally fall exactly on the cluster boundaries. The maximum separation was the sum of .75 standard deviations from each cluster. The actual values used in each case were selected randomly from the interval of the sum of .25 to .75 standard deviations from each cluster (uniform distribution). The separation parameter levels were chosen through empirical testing for acceptable recovery behavior by the algorithms.

For the dimensions where overlap of cluster boundaries was permitted, the maximum allowable range for each dimension was set equal to 2/3 of the range of the first dimension where overlap was not permitted. Within this range, individual cluster lengths of 10 to 40 units were chosen. However, the actual location of the cluster boundaries
within the range were chosen randomly and without regards for the location of the other clusters present. It should be noted that this method of cluster construction ensured that the constructed data sets contained clusters which exhibited external isolation. The actual ordering of the clusters on the variables was chosen randomly for each dimension. Thus, a variety of cluster configurations was possible.

In terms of the shape of the generated clusters, on the average the cluster boundaries formed a hypercube with all dimensional lengths equal. However, the possible range of the dimensional lengths allowed a maximum ratio of lengths of four to one between any pair of dimensions. Thus, at least some limited elongation of cluster structure could occur. The point density within cluster was greatest near the geometric centroid of the cluster. The point density gradually fell off moving away from the centroid and followed the contours of a hyperellipsoid, thus leaving the extreme corners of the cluster boundaries less dense than any other region of the cluster. Of course, no cluster point was allowed to fall outside the cluster boundaries.

The major characteristics of each constructed data set such as the number of clusters, the density of the clusters, and the maximum numerical range of the clusters on the first dimension were saved and used for the construction of a
random noise data set. The information was also utilized for the subsequent computation of the base line rates for the criterion statistics.

A listing of the computer programs used for the generation of the constructed data sets is given in Appendix A. The programs were written in Fortran IV.

Random noise data sets. One hundred eight random noise sets were also generated in order to provide null distribution values for the criterion statistics. Each of the parent data sets from Condition 1 provided the maximum range for each dimension for the generation of one random noise set. The coordinates of each point were selected randomly from the range of each dimension (uniform distribution). Euclidean distances between points were then computed to form the matrices which were used for all conditions.

Data Base

1. Error free cluster structure. The error free parent data sets were generated directly from the cluster construction program. Since the sets were not error perturbed, they were analyzed directly by each of the 15 clustering algorithms.

2. Data sets with outliers added. Condition 2 was created by adding intermediates or outliers to the parent data sets from Condition 1. Two levels were used which represented adding either an additional 20% or 40% of the total number of elements as outliers. Thus, 108 data sets
were generated where each set consisted of 60 points with 10 points as outliers. Another 108 data sets were generated with 70 points each where 20 points were outliers. The added points had coordinates which fell outside the cluster boundaries on at least one dimension. In order to generate the outliers, a modification was made to the multivariate normal random number generator which was used to generate the points within clusters. The diagonal entries of the variance-covariance matrix for each cluster were multiplied by a factor of nine (representing a threefold increase in the standard deviations). Outliers were generated for each cluster using the modified variance-covariance matrix and the normal distribution generator. The outliers were accepted with the restriction that each generated point must fall outside the boundaries of every cluster in the space. The number of outliers added to each cluster are given in Table 1. The number of outliers assigned to each cluster were set to be proportional to the relative size of each cluster.

3. Error perturbation of distances. Condition 3 represented the error perturbation of the distances in the Condition 1 data sets. Two levels of error perturbation each generated two sets of 108 data matrices. Since the between cluster separation was defined in terms of the inherent within cluster standard deviation, the error perturbation process involved adding error to the dimension-wise measurements proportional to the within cluster standard
deviations. Each interpoint distance in the original error free matrix was separately error perturbed. Specifically, the interpoint distances were recomputed as:

$$\sqrt{\Sigma(A_{ij} - A_{ik} + \gamma \epsilon_{ijk})^2}.$$ 

The $A_{ij}$ entry represents the coordinate value for point $j$ and dimension $i$. The $\epsilon_{ijk}$ entry is the error perturbation value on dimension $i$ for points $j$ and $k$. The $\epsilon_{ijk}$ values are obtained from a univariate normal distribution generator with a mean of zero. The standard deviation of the generator for each dimension was obtained by taking the average of the two within cluster standard deviations from the clusters which contained points $j$ and $k$. Finally, $\gamma$ is a multiplication factor which defined the error level. The parameter was set equal to 1.00 for the low error level and was reset to 2.00 for the high error condition. The low error condition generally ensured that the resulting perturbed clusters would not overlap common regions of space. However, the guarantee of external isolation was lost. The high error condition did allow the clusters to overlap common regions of space to some extent. However, the extent of overlap could not have been very severe.

Due to the variable spacing between clusters, the exact extent of cluster overlap cannot be specified. However, it is worth considering the worst possible case. That is, consider the situation where the minimum possible cluster separation occurred between two clusters. For simplicity,
assume that the clusters have equal standard deviations on all dimensions. Thus, on the dimension where overlap was not permitted, the clusters are separated by one-half standard deviation. On the remaining dimensions where cluster overlap was permitted, it is assumed that perfect overlap occurred on each dimension. (It should be noted that perfect overlap on any of the dimensions would be a rare event indeed. In fact, clusters need not have overlapped at all on the dimensions.) Thus, the only relevant dimension is the first dimension where the clusters were separated by .5 standard deviations. Hence, a point located on the neighboring cluster boundaries would be .5 standard deviations away from the next cluster while a point situated at the centroid would be 2 standard deviations distant. A point located at the opposite (or outside) cluster boundary would be 3.5 standard deviations away from the neighboring cluster boundary.

The low error condition involved adding normal error with a mean of zero and a standard deviation of 1.00. Although points near the cluster boundaries could move into the other cluster with a probability of about .30, it would be a fairly rare event for a point located at the centroid to move into the other cluster (probability of .04). It would be even more unlikely for a point located at the opposite cluster boundary to move into the other cluster (probability of less than .001).
The high error condition involved adding error with a standard deviation of 2.00. Thus, a point located at the neighboring cluster boundary would have a probability of .40 of moving into the other cluster. Points located at the centroid could move into the other cluster with a probability of .16. Finally, a point located at the opposite cluster boundary would have a probability of only about .04 of falling inside the other cluster.

It should be noted that the error perturbation process involved temporarily moving each point to a different location for the computation of each interpoint distance. Thus, the error model used allowed the error in the data to have a higher dimensionality than the space which defined the true cluster structure. The error model is similar, but not identical to various error models which have been used in the study of multidimensional scaling algorithms (Isaac and Poor, 1974).

4. Introduction of Random noise dimensions. The error free parent data sets from Condition 1 had one or two random error dimensions added to the existing set of defining dimensions. The error dimensions had a maximum range equal to the range of the dimension where cluster overlap was not permitted. The coordinates for each point in the space were distributed randomly over the computed interval (uniform distribution). One condition of 108 data sets was generated by adding one error dimension to the existing set. Another
set of 108 data matrices was generated by adding two error dimensions. The Euclidean distance function was applied to the augmented set of dimensions.

5. **Computation of the distances with a non-Euclidean index.** Condition 5 involved using a non-Euclidean distance metric on the original parent data matrices. Two different distance measures were computed. Each measure gave rise to 108 data matrices. Cattell's (1949) $r_p$ measure was computed as one index. The index is a similarity measure with a range of values identical to that of a correlation coefficient. It should be noted that the Cattell measure requires standardization of the data before computing the index. In order to convert the index to a dissimilarity measure, the following formula was used:

$$d_{r_p} = 1.00 - r_p$$

The transformation gave the index a range of values of zero to two with a value of zero occurring if and only if two points had the same location in the variable space. A value of two occurred only if two points were an infinite distance apart.

The Pearson correlation was used as the other similarity index. The same transformation as used for the $r_p$ index was also used for the Pearson coefficient.
6. **Standardization of the variables**. The last 108 data sets which were generated involved taking the parent data sets from Condition 1 and standardizing each variable separately to have a mean of zero and a variance of one. The Euclidean distance function was then computed from the data.

**Clustering Algorithms**

The 15 clustering algorithms examined in the present study included 11 agglomerative hierarchical methods and four non-hierarchical centroid sorting procedures. The algorithms selected represented some of the more commonly available clustering programs. The algorithms produce distinct non-overlapping partition structures and allow the researcher to select the similarity coefficient. All of the procedures determine the cluster solutions in a more or less objective manner. Hence, various types of ordination techniques were not considered since these procedures require considerable subjective evaluation by the researcher. The clustering algorithms were modified to compute the criterion statistics with the analysis of each input matrix. The programs were written in Fortran IV and listings are given in Appendix B.

Eight of the 11 hierarchical clustering procedures can be represented by their respective coefficients for the cluster similarity index developed by Lance and Williams (1967a). The recurrence formula gives the updated distance entry between a point (or cluster) labeled k and the newly
formed cluster \((ij)\) which was created out of the merger of groups \(i\) and \(j\). The formula in terms of distances is:

\[
d_k(ij) = \alpha_i d_{ki} + \alpha_j d_{kj} + \beta d_{ij} + \gamma |d_{ki} - d_{kj}|.
\]

The \(\alpha, \beta,\) and \(\gamma\) parameters uniquely define the clustering algorithm. The parameter values for the eight methods are given in Table 2. The \(n_i\) values used in the table represent the number of elements in cluster \(i\).

Both the single linkage and the complete linkage methods are discussed in detail in Johnson (1967). These are the only two known clustering methods which are invariant to monotone transformations of the input data matrix. Both methods are known by at least four other names. The single linkage method is also known as the nearest neighbor method (Lance and Williams, 1967a), the minimum method or the connectedness method (Johnson, 1967), or as linkage analysis (McQuitty, 1967). The complete linkage procedure is also referred to as the furthest neighbor method (Lance and Williams, 1967a), the maximum method or the diameter method (Johnson, 1967), or as rank order typal analysis (McQuitty, 1967).

Both the single linkage and the complete linkage methods are dependent on only a single value from the data matrix for the determination of the next merger at each level in the hierarchy. Since the data values which are chosen fall at the extremes of the range of the data entries, it is quite possible that the entry could represent a data
### Table 2

**Hierarchical Algorithms**

<table>
<thead>
<tr>
<th>Method</th>
<th>(a_i)</th>
<th>(a_j)</th>
<th>(\beta)</th>
<th>(\gamma)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Linkage</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>-1/2</td>
</tr>
<tr>
<td>Complete Linkage</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td>Group Average</td>
<td>(n_i/(n_i+n_j))</td>
<td>(n_j/(n_i+n_j))</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Centroid</td>
<td>(n_i/(n_i+n_j))</td>
<td>(n_j/(n_i+n_j))</td>
<td>(-n_i n_j/(n_i+n_j)^2)</td>
<td>0</td>
</tr>
<tr>
<td>Median</td>
<td>1/2</td>
<td>1/2</td>
<td>(-1/4)</td>
<td>0</td>
</tr>
<tr>
<td>Ward's Minimum Variance</td>
<td>((n_i+n_k)/(n_i+n_j+n_k))</td>
<td>((n_j+n_k)/(n_i+n_j+n_k))</td>
<td>(-n_k/(n_i+n_j+n_k))</td>
<td>0</td>
</tr>
<tr>
<td>Flexible</td>
<td>1/2(1-(\beta))</td>
<td>1/2(1-(\beta))</td>
<td>(\beta(&lt;1))</td>
<td>0</td>
</tr>
</tbody>
</table>
point which is subject to extreme error. For this reason a series of hierarchical methods based on some measure of the central location of the clusters have been proposed. The methods are known as the group average method (Sokal and Michener, 1958), the weighted average method (McQuitty, 1966), the centroid method (Gower, 1967), and the median method (Gower, 1967). Of the four methods the group average procedure has apparently been the most popular (Sneath, 1969).

Ward's minimum variance procedure attempts to find that merger which gives the minimum increase in the total within group sum of squares. The method was introduced by Ward (1963) and is a fairly popular technique. On the other hand, the flexible method has received little use since its introduction by Lance and Williams (1967a). The method possesses an additional parameter \( \beta \) which must be set by the researcher and has a range of \(-1 \leq \beta < 1\). Values of \( \beta \) greater than zero causes the method to have an increasing chaining characteristic similar to the single linkage method. Values below zero produce clusters with properties akin to the complete linkage method. When \( \beta \) equals zero, the method is identical to the weighted average method. For the present study, \( \beta \) was set at \(-.25\) as suggested by Lance and Williams (1967a).

The last three hierarchical algorithms cannot be represented by the Lance and Williams recurrence formula. Lance and Williams (1967a) state that any procedure not
satisfying the recurrence formula has the characteristic that an additional data matrix must be maintained throughout the clustering process in addition to the matrix which is updated during the clustering process. The three additional hierarchical methods do have such a programming characteristic.

One of the three methods is related to the group of procedures based on averaging cluster distances. The method is known as the average linkage within the new cluster (Anderberg, 1973). The algorithm attempts to find those clusters which minimize the average linkage within the newly formed cluster.

The last two methods are related to the Ward procedure and have been called by Anderberg (1973) minimum total within group sum of squares in the new cluster and minimum average within group sum of squares in the new cluster. It should be noted that these three algorithms are actually using different objective criteria to determine which clusters are merged at the next step in the hierarchy. Ward's method selects that partition which gives rise to the minimum increase in the total within group sum of squares. The other two methods select those cluster mergers which either give the minimum total error sum of squares or the minimum average error sum of squares across the clusters. Anderberg notes that the last two methods tend to produce
clusters with approximately equal within cluster variances. The minimum average method has a particularly strong tendency to form the equal variance clusters.

The four non-hierarchical programs examined are all variants of the original K-means algorithms developed by MacQueen (1967). The methods are MacQueen's original technique, Forgy's (1965) method, Jancey's (1966) variant, and the convergent K-means algorithm (Anderberg, 1973). Initially, the researcher specifies that K clusters are to be extracted from the data. All four methods make two or more passes through the data set and assign each data point to the nearest proposed cluster centroid. Specifically, MacQueen's original method utilized the first K data points as the proposed set of K cluster centroids. During the first pass through the data set, the program assigns each point to its nearest cluster centroid. The cluster centroids are updated to reflect the current cluster membership after each point is assigned. A final reassignment pass is made through the data set. However, the cluster centroids remain fixed during the final pass. The resulting cluster assignments form the final K-means solution partition. Anderberg (1973) discusses the convergence properties of these four algorithms.

A generalized approach to selecting the starting configuration would involve selecting K points at random from the data set. This approach represents a reasonable
strategy which an applied researcher might adopt. The approach was used for all four non-hierarchical clustering algorithms in the present study.

The convergent K-means procedure is similar to the basic MacQueen algorithm. In fact, in the present study the methods are identical up to and including the first pass through the data. However, during the second pass through the data the convergent method updates the centroids after each point is reassigned rather than holding the centroids fixed. Thus, the second pass is identical in execution to the first pass. Additional reassignment passes are made through the data until a pass is made where no data points change cluster assignment. The resulting cluster assignments become the convergent K-means solution partition.

Forgy's method starts out with the initial proposed cluster centroids and assigns each point to its nearest cluster centroid. However, the cluster centroids are not updated until a complete pass is made through the data. Additional reassignment passes are made in exactly the same manner until a pass is made where no data units change cluster membership.

Jancey's procedure is identical to Forgy's algorithm except for the computation of the updated cluster centroids at the end of each pass through the data. Jancey's method reflects the old centroid through the new centroid as computed by Forgy's procedure. That is, a vector is
constructed in the space from the old cluster centroid to the centroid which would be computed for the Forgy method. However, the vector is extended beyond the Forgy centroid for an equivalent length and in the same direction. The end point of the vector becomes the updated cluster centroid for the Jancey procedure. The process overshoots the updated Forgy centroid in an attempt to accelerate convergence and possibly lead to a better partitioning of the data by avoiding entrapment in local minima.

All four of the non-hierarchical programs were set to produce the same number of clusters as were present in the constructed data. The process exactly matches that procedure whereby the correct partition level of the hierarchy was examined for the hierarchical clustering methods.

Recovery Measures

External Criterion Measure

The Rand (1971) statistic served as the primary measure of cluster recovery. The statistic is based on a square matrix whose rows and columns correspond to the entities to be clustered. The matrix entries are $\delta_{ij}$ values which are either 0 or 1. $\delta_{ij}$ equals 1 if items i and j are clustered together in both the criterion definition of the clusters and the obtained cluster solution. $\delta_{ij}$ also equals 1 if items i and j are not clustered together in both solutions. Thus, a value of 1 indicates agreement between the criterion definition of the cluster structure and the obtained
solution. A value of zero is assigned to a $\delta_{ij}$ entry if in one solution the items are clustered together while in the other partition the items are not clustered together. Thus, a 0 entry represents a failure of the clustering algorithm to correctly classify the pair of items. Formally, when $n$ is the number of elements in the data set, the statistic is computed as:

$$\frac{\sum_{j=1}^{n} \left( \sum_{i>j}^{n} \delta_{ij} \right)}{n(n-1)/2}$$

The denominator is simply the total number of possible comparisons. The measure has fixed limits of 0.0 to 1.0 with a value of 1.0 indicating perfect cluster recovery. As noted in Chapter 1, the index has several distinct advantages over the other proposed external criterion measures.

The index was computed for each clustering method in all six error conditions. Computation of the statistic for all error conditions except Condition 2 was straightforward and presented no complications since the criterion solution and the obtained solution had the same number of items in strict one-to-one correspondence. However, Condition 2 had more points in the cluster solution than in the criterion definition of the cluster structure since intermediates or outliers had been added to the set. The procedure which was used for the computation of the index value for Condition 2 involved simply ignoring the extra points in the set. The index was computed on only those elements which defined the
criterion clusters. The statistic would achieve a value of 1.00 only if the algorithm produced a partition of the data in which the criterion points were perfectly clustered. Ignoring the outliers also left the statistic unaffected for comparison purposes. In Condition 2, the statistic was computed for the partition level which corresponded to the earliest level at which perfect clustering of the data could occur. Hence, the outliers were allowed to remain unassigned single point clusters. The procedure follows closely the rationale developed by Edelbrock (1977).

Finally, base line or null distribution values for the Rand statistic were collected from the random noise data sets for each clustering method. Arbitrary partitions were defined as the criterion clusters and index values were computed from the algorithm solutions for each data set. The number and size of the arbitrary criterion clusters matched those values represented in the set of basic parent data matrices generated for Condition 1.

Internal Criterion Measures

*Single partition cophenetic correlation coefficient.*

As an internal criterion measure of cluster recovery, a Pearson based cophenetic correlation coefficient was computed at the appropriate partition level for the hierarchical methods or from the solution partition produced by the non-hierarchical methods. The correlation was between the input distance matrix and a 0-1 dissimilarity matrix which was
defined by the partition solution from a given clustering method. A value of 0 was assigned to a matrix entry if the corresponding elements had been clustered together by the algorithm. A value of 1 was entered otherwise. Hence, the index was a point-biserial correlation and was used as a goodness-of-fit measure of cluster recovery in terms of the input distances rather than in terms of an external criterion.

The computation of the coefficient was straightforward in all conditions except Condition 2. Condition 2 had a number of points added to the data sets which did not define the underlying cluster partition structure. If the index was computed on all of the points in the data set, then the measure would have indicated the goodness-of-fit of the method in fitting all distance values including the outliers. However, since the correct K-cluster partition was being sought, the outliers were allowed to be unassigned single point clusters. Hence, a correlation based on all data values would have reflected this source of distortion even if the criterion clusters were perfectly recovered. Basing the correlation on the reduced data set which consisted only of the points in the true clusters removed the distortion due to the outliers. The correlation computed from the reduced matrix would also be comparable to the correlations computed for the other conditions. If the resulting cluster solution of a data matrix from Condition 2 perfectly
recovered the criterion clusters, then the correlation based on the reduced matrix would exactly equal the correlation computed from a perfect clustering of the parent matrix in Condition 1. The cophenetic correlation based on the reduced data set seemed more reasonable in terms of isolating the effects which were being examined. Thus, the reduced set correlation was computed for each method in Condition 2.

Baseline rates were also collected for the statistic for each method from the random noise data sets. Since even random noise data possesses some inherent structure, the resulting correlations were expected to be greater than zero. However, for the correlation to be a useful index of cluster recovery, it would be necessary for the null distribution values to be significantly different from the correlations generated from data sets with true cluster structure present.

Hierarchical cophenetic correlation coefficient. An alternative conceptualization for the cophenetic correlation coefficient is in terms of the recovery of the entire hierarchy structure rather than a single partition. Of course, the index is meaningfully defined only for the hierarchical clustering algorithms. Thus, the coefficient was computed only for the 11 hierarchical methods examined in the present study.

A Pearson correlation was computed between the input data matrix and the distance matrix induced by the hierarchical classification. The induced distance matrix was obtained
from the hierarchical solution by taking the dissimilarity value produced at each level of the hierarchy and filling in the appropriate entries in the matrix. The appropriate entries corresponded to those pairwise point combinations which were involved in the cluster merger at the given hierarchy level.

The same solution for the problem associated with the Condition 2 data sets which was used for the single partition cophenetic correlation was used for the hierarchical correlation coefficient. Base line rates were also collected for the hierarchical correlation from the random noise data sets. Here again, it was expected that the null distribution correlations would be greater than zero.

**Correlation Measure of the Error Perturbation Process**

As a measure of the degree of distortion of the error perturbation process, a Pearson correlation was computed between the error free parent data matrices from Condition 1 and the resulting error perturbed data matrices from Conditions 3, 4, 5, and 6. The coefficient was computed from the distances or other dissimilarities appropriate for the given condition. Unfortunately, a meaningful correlation could not be computed for the Condition 2 data sets. It would have been possible to compute the correlation by ignoring the added intermediates or outliers. However, since the original elements which defined the clusters were not disturbed, the correlation between the sets would have always
been 1.00. It also seemed inappropriate to ignore the added
elements since these actually represented the error pertur-
bation.
III. RESULTS

External Criterion Measure

Exact Recovery

Descriptive statistics. An exact recovery was defined as perfect cluster recovery for the purpose of the study. Thus, frequency counts were tabulated for which a Rand statistic value of 1.00 was obtained for each method and condition combination. Exact recovery was considered to be a relevant measure of cluster recovery particularly for the error free parent data sets and the mild error conditions. The recovery frequencies are presented in Table 3. As an aid in studying the table, the rank order performance across methods within error condition is given in Table 4.

1. Error free data. The highest exact recovery was obtained in the error free condition for each of the first 11 methods. These methods also corresponded to the 11 hierarchical clustering algorithms. Interestingly, slightly higher recovery was obtained in some of the error conditions for three of the four non-hierarchical methods. Other than a random chance explanation, the effect may be due to slight improvements in the random starting seed points which were used for the non-hierarchical methods.
<table>
<thead>
<tr>
<th>Method</th>
<th>Error Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Error Free</td>
</tr>
<tr>
<td>Single Link</td>
<td>93</td>
</tr>
<tr>
<td>Complete Link</td>
<td>102</td>
</tr>
<tr>
<td>Group Average</td>
<td>104</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>103</td>
</tr>
<tr>
<td>Centroid Method</td>
<td>100</td>
</tr>
<tr>
<td>Median Method</td>
<td>94</td>
</tr>
<tr>
<td>Ward’s Min. Variance</td>
<td>101</td>
</tr>
<tr>
<td>Beta-Flexible</td>
<td>102</td>
</tr>
<tr>
<td>Ave. Link in New Cluster</td>
<td>96</td>
</tr>
<tr>
<td>Min. Total Sum of Square</td>
<td>83</td>
</tr>
<tr>
<td>Min. Ave. Sum of Square</td>
<td>96</td>
</tr>
<tr>
<td>MacQueen’s Method</td>
<td>37</td>
</tr>
<tr>
<td>Forgy’s Method</td>
<td>66</td>
</tr>
<tr>
<td>Jancey’s Method</td>
<td>67</td>
</tr>
<tr>
<td>Convergent K-means</td>
<td>54</td>
</tr>
</tbody>
</table>

**NOTE:** The maximum score in any cell is 108.
<table>
<thead>
<tr>
<th>Method</th>
<th>Error Free</th>
<th>20% Outliers</th>
<th>40% Outliers</th>
<th>Low Error</th>
<th>High Error</th>
<th>1-dim. Error</th>
<th>2-dim. Error</th>
<th>r</th>
<th>Std. Scores</th>
<th>Std. p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Link</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>10</td>
<td>13.5</td>
<td>1</td>
<td>4.5</td>
<td>10</td>
<td>7</td>
<td>6.5</td>
</tr>
<tr>
<td>Complete Link</td>
<td>3.5</td>
<td>10</td>
<td>12.5</td>
<td>5</td>
<td>6</td>
<td>9.5</td>
<td>7</td>
<td>6.5</td>
<td>4.5</td>
<td>5</td>
</tr>
<tr>
<td>Group Average</td>
<td>1</td>
<td>5</td>
<td>7.5</td>
<td>1.5</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3.5</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>2</td>
<td>9</td>
<td>9</td>
<td>1.5</td>
<td>2.5</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4.5</td>
<td>3.5</td>
</tr>
<tr>
<td>Centroid Method</td>
<td>6</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>11</td>
<td>13</td>
<td>15</td>
<td>4</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Median Method</td>
<td>8</td>
<td>6</td>
<td>7.5</td>
<td>7</td>
<td>7.5</td>
<td>14</td>
<td>14</td>
<td>6.5</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>Ward's Min. Variance</td>
<td>5</td>
<td>14.5</td>
<td>12.5</td>
<td>4</td>
<td>2.5</td>
<td>6</td>
<td>4.5</td>
<td>5</td>
<td>1.5</td>
<td>2</td>
</tr>
<tr>
<td>Beta-Flexible</td>
<td>3.5</td>
<td>11</td>
<td>12.5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3.5</td>
<td>1</td>
</tr>
<tr>
<td>Ave. Link in New Cluster</td>
<td>10</td>
<td>12</td>
<td>12.5</td>
<td>9</td>
<td>7.5</td>
<td>8</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Min. Total Sum of Squares</td>
<td>11</td>
<td>14.5</td>
<td>12.5</td>
<td>11</td>
<td>12</td>
<td>12</td>
<td>12.5</td>
<td>8</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>Min. Ave. Sum of Squares</td>
<td>7</td>
<td>13</td>
<td>12.5</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>9</td>
<td>6</td>
<td>6.5</td>
</tr>
<tr>
<td>MacQueen's Method</td>
<td>15</td>
<td>7</td>
<td>4</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>12.5</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Forgy's Method</td>
<td>13</td>
<td>2</td>
<td>3</td>
<td>13</td>
<td>13.5</td>
<td>11</td>
<td>10.5</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>Lancey's Method</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>12</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>Convergent K-means</td>
<td>14</td>
<td>4</td>
<td>2</td>
<td>14</td>
<td>10</td>
<td>9.5</td>
<td>10.5</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
</tbody>
</table>

**Note:** Tied rates were assigned averaged ranks.
In the error free condition, recovery can be considered fairly good for the hierarchical methods. Six of the 11 methods recovered perfectly 100 or more of the 108 possible data sets. Three of the remaining methods had recovery frequencies above 90 while the last two methods had frequencies above 80. The four non-hierarchical methods produced distinctly lower recovery with the highest frequencies below 70. Apparently, the four K-means algorithms performed rather poorly in terms of exact recovery in the error free condition where distinct clustering was present. Overall, the group average hierarchical procedure gave the best exact recovery. Since the non-hierarchical methods were preset to recover the correct number of clusters and the hierarchical methods were examined at the appropriate partition level, neither approach had an advantage or bias in regards to correct cluster recovery.

2. Outlier conditions. A fairly strong decrement in recovery performance was found for the 20% and 40% outlier conditions for the hierarchical procedures. In several cases the frequencies fell to zero. Interestingly, when compared to their respective error free frequencies the four K-means procedures were virtually unaffected by the addition of outliers and intermediates to the configuration. All four of the non-hierarchical methods gave better recovery than any of the 11 hierarchical methods. Jancey's method gave the best recovery for any of the 15 methods studied.
It should be noted that the single linkage method was not nearly as affected by the addition of outliers as might have been expected. The complete link and Ward's minimum variance method did noticeably worse in terms of recovery.

3. **Distance perturbation conditions.** For the two conditions involving the error perturbation of the distances, slight to moderate decrements were found for all methods. This, of course, implies that the K-means procedures gave lower frequencies than the hierarchical methods. The group average method gave the highest recovery frequency with the weighted average and Ward's method either tying or placing a close second. It should be noted that the single linkage method gave the lowest or next to lowest recovery for any of the hierarchical methods. This result is in accord with the findings of the studies by D'Andrade (1978), Blashfield (1976), Edelbrock (1977), and Milligan and Isaac (1978). However, the lack of superiority of the Ward's method over the group average method is not consistent with the multivariate normal mixture studies such as Blashfield's (1976) and Kuiper and Fisher (1975).

4. **Random noise dimension conditions.** The error conditions in which one or two dimensions of random noise were added to the basic set of defining dimensions produced a fairly strong decrement in performance for all methods. The group average and weighted average methods usually gave the best recovery. However, the single linkage method actually
gave the highest recovery frequency for the one-dimensional error condition. The recovery performance for the single link method falls off noticeably for the two-dimensional condition. This may indicate that the higher recovery obtained from the one-dimensional condition could possibly have been sampling error in the Monte Carlo process.

5. **Non-Euclidean indices and variable standardization conditions.** The last three error conditions represented various types of error perturbation which can take place when preparing the raw data for input to the clustering algorithms. A fairly slight drop in recovery performance was obtained for all three conditions when compared to the error free recovery frequencies. Only slight to moderate differences were observed between the hierarchical methods. The beta-flexible method gave the best recovery in all three conditions, although it did tie for first place with Ward's method for the standard scores error condition. Actually, within any of the three conditions, the difference between the first and fifth ranking methods was at most only six data sets. The group average method ranked second in two conditions and third in the other condition. Interestingly, three of the four K-means procedures gave their highest recovery in these three conditions. However, the differences when compared to the error free condition is very slight with the maximum discrepancy being five data sets.
Overall, the hierarchical methods performed better than the non-hierarchical methods in eight out of ten error conditions. The non-hierarchical methods performed better only in the conditions involving outliers. Among the non-hierarchical procedures, Jancey's method gave the highest recovery in each error condition. It should be noted that the group average method ranked first in recovery in four of the error conditions, second in three conditions and third in one condition. Except for the outlier conditions, the group average method seems to give fairly good recovery performance.

**Hypothesis testing.** Since the data consisted of observed frequency counts, an appropriate data analysis procedure would be a contingency table approach. In the present situation, three classification variables are involved. The variables are: (a) methods (15 levels), (b) error conditions (10 levels), and (c) recovery (2 levels). Thus, a three dimensional contingency table analysis was undertaken using the procedures presented by Bishop, Fienberg, and Holland (1975). (It should be noted that strict independence cannot be assumed since the same 108 data sets were analyzed by each of the 15 methods and since the same 108 data sets were subjected to all 10 levels of error perturbation. Although the independence assumption was violated, the descriptive value of the test statistics remains useful, if not the inferential procedure itself.)
A test of the three-way interaction produced a chi-square value of 1,779.05 with 126 degrees of freedom. The observed value is significant well beyond the .001 level. Thus, it seems reasonable to conclude that recovery is strongly dependent on error condition and method type. An examination of the deviation of the expected to observed frequencies (residuals of prediction), and an examination of the u-terms in the saturated model (analogous to beta weights in multiple regression) indicated two distinct patterns in the data. One pattern was found for the hierarchical methods and another pattern was obtained from the non-hierarchical methods. Thus, the original table was broken into two separate tables, one table for the hierarchical methods and the other for the non-hierarchical procedures.

The results for the table consisting of the hierarchical methods indicated that again, a significant three-way interaction existed in the data (chi-square value of 546.63, 90 degrees of freedom, p < .001). Thus, the dependency of recovery on error condition and method persists.

However, the test for the three factor interaction for the non-hierarchical procedures produced null results (chi-square value of 19.78, 27 degrees of freedom, p > .5). Thus, the performance of these methods is not dependent on the error condition. This is consistent with the observed result that Jancey's method always produced the best recovery for the non-hierarchical methods. When considering the
lower order interactions, two of the two-way interactions were found significant. Methods were found to differ in terms of recovery (chi-square value of 179.3, 30 degrees of freedom, \( p < .001 \)). The error conditions also had a differential effect on recovery (chi-square value of 253.61, 36 degrees of freedom, \( p < .001 \)). No interaction was found between error condition and method (chi-square value of 27.99, 54 degrees of freedom, \( p > .5 \)). This last result seems reasonable since the response variable, recovery, was not a factor in the test.

**Mean Recovery Rate - Rand Index**

Although exact recovery is interesting and particularly useful for the very mild error conditions, exact recovery is a very stringent criterion. For applied situations, it would be useful to also determine whether the cluster recovery was close to a perfect recovery, or whether recovery was at least above chance levels. Thus, an analysis of mean recovery rates is appropriate.

**Descriptive statistics.** The mean recovery rates along with standard deviations are presented in Table 5 for the hierarchical methods and in Table 6 for the non-hierarchical methods. Again, for convenience, the rank order performance within each error condition is given in Table 7.
<table>
<thead>
<tr>
<th>Method</th>
<th>Error Condition</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>20% Outlier</td>
<td>60% Outlier</td>
<td>Low Error</td>
<td>High Error</td>
<td>1-dim. Error</td>
<td>2-dim. Error</td>
<td>r</td>
<td>Stl. Sources</td>
<td>r</td>
<td>Random Noise</td>
</tr>
<tr>
<td>Single Link</td>
<td></td>
<td>.974</td>
<td>.970</td>
<td>.960</td>
<td>.903</td>
<td>.777</td>
<td>.899</td>
<td>.843</td>
<td>.928</td>
<td>.922</td>
<td>.922</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.087)</td>
<td>(.085)</td>
<td>(.081)</td>
<td>(.173)</td>
<td>(.204)</td>
<td>(.145)</td>
<td>(.163)</td>
<td>(.142)</td>
<td>(.155)</td>
<td>(.135)</td>
</tr>
<tr>
<td>Complete Link</td>
<td></td>
<td>.995</td>
<td>.883</td>
<td>.799</td>
<td>.570</td>
<td>.880</td>
<td>.989</td>
<td>.927</td>
<td>.961</td>
<td>.981</td>
<td>.984</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.031)</td>
<td>(.158)</td>
<td>(.127)</td>
<td>(.104)</td>
<td>(.161)</td>
<td>(.122)</td>
<td>(.144)</td>
<td>(.099)</td>
<td>(.056)</td>
<td>(.048)</td>
</tr>
<tr>
<td>Group Average</td>
<td></td>
<td>.998</td>
<td>.963</td>
<td>.914</td>
<td>.997</td>
<td>.948</td>
<td>.970</td>
<td>.903</td>
<td>.969</td>
<td>.986</td>
<td>.986</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.010)</td>
<td>(.061)</td>
<td>(.086)</td>
<td>(.012)</td>
<td>(.094)</td>
<td>(.092)</td>
<td>(.120)</td>
<td>(.086)</td>
<td>(.044)</td>
<td>(.042)</td>
</tr>
<tr>
<td>Weighted Average</td>
<td></td>
<td>.904</td>
<td>.912</td>
<td>.899</td>
<td>.997</td>
<td>.934</td>
<td>.917</td>
<td>.985</td>
<td>.966</td>
<td>.984</td>
<td>.985</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.049)</td>
<td>(.085)</td>
<td>(.107)</td>
<td>(.017)</td>
<td>(.119)</td>
<td>(.107)</td>
<td>(.125)</td>
<td>(.090)</td>
<td>(.064)</td>
<td>(.044)</td>
</tr>
<tr>
<td>Centroid Method</td>
<td></td>
<td>.993</td>
<td>.990</td>
<td>.968</td>
<td>.948</td>
<td>.810</td>
<td>.808</td>
<td>.616</td>
<td>.967</td>
<td>.903</td>
<td>.866</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.085)</td>
<td>(.071)</td>
<td>(.070)</td>
<td>(.120)</td>
<td>(.205)</td>
<td>(.156)</td>
<td>(.137)</td>
<td>(.086)</td>
<td>(.174)</td>
<td>(.201)</td>
</tr>
<tr>
<td>Median Method</td>
<td></td>
<td>.976</td>
<td>.964</td>
<td>.946</td>
<td>.960</td>
<td>.831</td>
<td>.808</td>
<td>.661</td>
<td>.958</td>
<td>.925</td>
<td>.963</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.081)</td>
<td>(.078)</td>
<td>(.077)</td>
<td>(.090)</td>
<td>(.188)</td>
<td>(.146)</td>
<td>(.193)</td>
<td>(.104)</td>
<td>(.147)</td>
<td>(.197)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.068)</td>
<td>(.151)</td>
<td>(.147)</td>
<td>(.050)</td>
<td>(.111)</td>
<td>(.143)</td>
<td>(.161)</td>
<td>(.845)</td>
<td>(.061)</td>
<td>(.076)</td>
</tr>
<tr>
<td>Beta-Flexible</td>
<td></td>
<td>.997</td>
<td>.836</td>
<td>.768</td>
<td>.994</td>
<td>.945</td>
<td>.904</td>
<td>.863</td>
<td>.972</td>
<td>.987</td>
<td>.983</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.019)</td>
<td>(.135)</td>
<td>(.137)</td>
<td>(.032)</td>
<td>(.105)</td>
<td>(.121)</td>
<td>(.137)</td>
<td>(.081)</td>
<td>(.047)</td>
<td>(.063)</td>
</tr>
<tr>
<td>Ave. Link in the New Cluster</td>
<td></td>
<td>.985</td>
<td>.876</td>
<td>.825</td>
<td>.980</td>
<td>.906</td>
<td>.906</td>
<td>.936</td>
<td>.999</td>
<td>.972</td>
<td>.951</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.053)</td>
<td>(.128)</td>
<td>(.136)</td>
<td>(.838)</td>
<td>(.133)</td>
<td>(.143)</td>
<td>(.147)</td>
<td>(.095)</td>
<td>(.059)</td>
<td>(.094)</td>
</tr>
<tr>
<td>Min. Total Sum of Squares</td>
<td></td>
<td>.935</td>
<td>.710</td>
<td>.673</td>
<td>.906</td>
<td>.835</td>
<td>.837</td>
<td>.780</td>
<td>.963</td>
<td>.911</td>
<td>.904</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.134)</td>
<td>(.156)</td>
<td>(.161)</td>
<td>(.148)</td>
<td>(.167)</td>
<td>(.164)</td>
<td>(.150)</td>
<td>(.087)</td>
<td>(.143)</td>
<td>(.198)</td>
</tr>
<tr>
<td>Min. Ave. Sum of Squares</td>
<td></td>
<td>.993</td>
<td>.828</td>
<td>.772</td>
<td>.987</td>
<td>.919</td>
<td>.900</td>
<td>.965</td>
<td>.979</td>
<td>.962</td>
<td>.962</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.023)</td>
<td>(.135)</td>
<td>(.126)</td>
<td>(.049)</td>
<td>(.124)</td>
<td>(.131)</td>
<td>(.097)</td>
<td>(.064)</td>
<td>(.094)</td>
<td>(.066)</td>
</tr>
<tr>
<td>Method</td>
<td>Error</td>
<td>20% Error</td>
<td>40% Error</td>
<td>Low Error</td>
<td>High Error</td>
<td>1-dim. Error</td>
<td>2-dim. Error</td>
<td>r</td>
<td>Std. Scores</td>
<td>r</td>
<td>Random Noise</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>------------</td>
<td>--------------</td>
<td>--------------</td>
<td>-----</td>
<td>-------------</td>
<td>-----</td>
<td>--------------</td>
</tr>
<tr>
<td>MacQueen's Method</td>
<td>.884</td>
<td>.895</td>
<td>.877</td>
<td>.884</td>
<td>.842</td>
<td>.793</td>
<td>.769</td>
<td>.894</td>
<td>.872</td>
<td>.872</td>
<td>.563</td>
</tr>
<tr>
<td></td>
<td>(.110)</td>
<td>(.119)</td>
<td>(.149)</td>
<td>(.114)</td>
<td>(.134)</td>
<td>(.139)</td>
<td>(.145)</td>
<td></td>
<td>(.114)</td>
<td>(.126)</td>
<td>(.121)</td>
</tr>
<tr>
<td>Forgy's Method</td>
<td>.932</td>
<td>.918</td>
<td>.911</td>
<td>.909</td>
<td>.872</td>
<td>.844</td>
<td>.794</td>
<td>.914</td>
<td>.915</td>
<td>.916</td>
<td>.572</td>
</tr>
<tr>
<td></td>
<td>(.102)</td>
<td>(.125)</td>
<td>(.128)</td>
<td>(.122)</td>
<td>(.123)</td>
<td>(.135)</td>
<td>(.146)</td>
<td></td>
<td>(.115)</td>
<td>(.110)</td>
<td>(.115)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jancer's Method</td>
<td>.927</td>
<td>.949</td>
<td>.952</td>
<td>.926</td>
<td>.909</td>
<td>.867</td>
<td>.823</td>
<td>.924</td>
<td>.946</td>
<td>.939</td>
<td>.569</td>
</tr>
<tr>
<td></td>
<td>(.119)</td>
<td>(.103)</td>
<td>(.105)</td>
<td>(.120)</td>
<td>(.121)</td>
<td>(.133)</td>
<td>(.149)</td>
<td></td>
<td>(.113)</td>
<td>(.102)</td>
<td>(.094)</td>
</tr>
<tr>
<td>Convergent K-means</td>
<td>.903</td>
<td>.920</td>
<td>.926</td>
<td>.901</td>
<td>.897</td>
<td>.849</td>
<td>.787</td>
<td>.914</td>
<td>.923</td>
<td>.915</td>
<td>.567</td>
</tr>
<tr>
<td></td>
<td>(.118)</td>
<td>(.109)</td>
<td>(.123)</td>
<td>(.125)</td>
<td>(.121)</td>
<td>(.133)</td>
<td>(.148)</td>
<td></td>
<td>(.119)</td>
<td>(.110)</td>
<td>(.124)</td>
</tr>
</tbody>
</table>

**NOTE:** First entry is the mean, parenthetical entry is the standard deviation.
<table>
<thead>
<tr>
<th>Method</th>
<th>Error Free</th>
<th>20% Outlier</th>
<th>40% Outlier</th>
<th>Low Error</th>
<th>High Error</th>
<th>1-dim. Error</th>
<th>2-dim. Error</th>
<th>r</th>
<th>Std. Scores</th>
<th>r_P</th>
<th>Equivalent Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Link</td>
<td>10*</td>
<td>2*</td>
<td>2*</td>
<td>13</td>
<td>15</td>
<td>5</td>
<td>6</td>
<td>11</td>
<td>11</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>Complete Link</td>
<td>3*</td>
<td>10</td>
<td>11</td>
<td>7*</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>8*</td>
<td>5*</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Group Average</td>
<td>1*</td>
<td>4*</td>
<td>6</td>
<td>1.5*</td>
<td>1*</td>
<td>1*</td>
<td>1</td>
<td>3*</td>
<td>2*</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>4*</td>
<td>6</td>
<td>8</td>
<td>1.5*</td>
<td>4*</td>
<td>2*</td>
<td>2*</td>
<td>5*</td>
<td>3*</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Centroid Method</td>
<td>8*</td>
<td>1*</td>
<td>1*</td>
<td>9</td>
<td>14</td>
<td>13.5</td>
<td>15</td>
<td>4*</td>
<td>14</td>
<td>15</td>
<td>4</td>
</tr>
<tr>
<td>Median Method</td>
<td>9*</td>
<td>3*</td>
<td>4*</td>
<td>8</td>
<td>13</td>
<td>13.5</td>
<td>14</td>
<td>10*</td>
<td>9</td>
<td>15</td>
<td>4</td>
</tr>
<tr>
<td>Ward’s Min. Variance</td>
<td>6*</td>
<td>14</td>
<td>14</td>
<td>4*</td>
<td>3*</td>
<td>6</td>
<td>5</td>
<td>2*</td>
<td>4*</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Beta-Flexible</td>
<td>2*</td>
<td>12</td>
<td>13</td>
<td>3*</td>
<td>2*</td>
<td>3*</td>
<td>4*</td>
<td>1*</td>
<td>1*</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Ave. Link in New Cluster</td>
<td>7*</td>
<td>11</td>
<td>10</td>
<td>6*</td>
<td>7*</td>
<td>7</td>
<td>7</td>
<td>9*</td>
<td>7*</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Min. Total Sum of Squares</td>
<td>11</td>
<td>15</td>
<td>15</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>7*</td>
<td>13</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>Min. Ave. Sum of Squares</td>
<td>5*</td>
<td>13</td>
<td>12</td>
<td>5*</td>
<td>5*</td>
<td>4*</td>
<td>3*</td>
<td>6*</td>
<td>6*</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>MacQueen’s Method</td>
<td>15</td>
<td>9</td>
<td>9</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>Forry’s Method</td>
<td>12</td>
<td>8</td>
<td>7</td>
<td>11</td>
<td>10</td>
<td>11</td>
<td>10</td>
<td>13.5</td>
<td>12</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Jancey’s Method</td>
<td>13</td>
<td>5</td>
<td>3*</td>
<td>10</td>
<td>6*</td>
<td>8</td>
<td>9</td>
<td>12</td>
<td>8</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>Convergent K-means</td>
<td>14</td>
<td>1</td>
<td>5</td>
<td>14</td>
<td>8</td>
<td>10</td>
<td>11</td>
<td>13.5</td>
<td>10</td>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>

NOTE: Tied rates were assigned average ranks.

*The column indicates the number of times each method placed in the statistically equivalent group.

*Asterisk notation indicates that the method is a member of the statistically equivalent superior group for the given error condition.
It should be noted that an additional column appears in Tables 5 and 6. The column gives the mean rates for the random noise data sets. These rates represent baseline or "null distribution" means for the recovery statistic for each method.

1. **Error free data.** In general, the same trends tend to hold as with exact recovery, with only a few exceptions. For the error free condition, all methods except MacQueen's method produced mean recovery rates above .90. All of the mean recovery rates are well above the baseline means in terms of the observed standard deviations. The group average procedure gave the highest mean recovery rate of any method. All four of the K-means methods gave lower recovery rates than any of the hierarchical methods.

2. **Outlier conditions.** The 20% and 40% outlier conditions produced differential effects on the methods. The complete linkage and Ward's method exhibited fairly noticeable decrements in recovery rates while the single linkage, group average, and centroid methods exhibited only a slight decrease in recovery when compared to the error free condition. It should be noted that the centroid method produced the highest mean recovery rate of all 15 methods, including the non-hierarchical methods. This result differs from the rank order performance obtained from exact recovery where the non-hierarchical methods produced higher exact rates than virtually any of the hierarchical procedures. Thus, it
seems that on the average the centroid method gave better overall recovery, although it did not obtain perfect recovery as often as some of the other methods. This result is also confirmed by noting that the centroid method produced smaller standard deviations than any of the K-means algorithms. It is also interesting to note that the four K-means procedures were virtually unaffected by the addition of outliers. In fact, three of these methods gave slightly higher mean recovery rates in the outlier conditions than in the error free condition.

3. **Distance perturbation conditions.** For the distance perturbation conditions, slight to moderate decrements in mean recovery were observed. The single linkage method was particularly affected by this error condition. The group average method gave the highest recovery rate in both high and low error levels, although the weighted average method did tie for the highest recovery rate at the low error level condition. Again, the K-means procedures were only slightly affected by the error conditions. The K-means methods did tend to give lower recovery rates than most of the hierarchical methods.

4. **Random noise dimension conditions.** The one and two-dimensional error conditions produced moderate to strong decrements in mean recovery. The K-means procedures even showed a marked decrease in performance for these
error conditions. The group average method again gave the highest recovery rates in both conditions while the weighted average method placed second.

5. Non-Euclidean indices and variable standardization conditions. The last three error conditions (r, standard scores, and $r_p$) produced fairly slight decrements in mean recovery rates. Most recovery rates remained above .90. Also, there is not a great deal of difference in mean recovery among the best four or five methods. The beta-flexible method gave the highest recovery rate in two conditions while the group average method gave better recovery in the third ($r_p$) condition.

When considering the mean rates obtained from the random noise data, it seems clear that the methods were recovering a great deal of cluster structure on the average in all of the error conditions. Overall, the group average method seems to give the best recovery performance. The method placed first in recovery six times, second once, third once, fourth once, and its lowest ranking was sixth. The method exhibited its lowest recovery performances in the two outlier error conditions.

Analysis of variance. Since the 108 basic error free parent data sets were subjected to all error conditions and were analyzed by all methods, the experimental design corresponded to a two factor within "subjects" analysis of variance. Both methods and conditions were assumed to be fixed
factors. Since the Rand statistic is bounded by 0.0 and 1.0, it seemed appropriate to transform the observed scores by the arc-sine transformation. However, in no case did the analysis of variance results differ between the raw and transformed scores in terms of the final inferences. In the tables below, the sum of squares which are reported are in terms of the raw scores rather than the transformed values.

The overall analysis of variance is presented in Table 8. Since homogeneity of treatment difference variances could not be assumed, the Geisser and Greenhouse correction procedure was adopted to control the Type I error rate for the repeated measures tests. Both main effects and the interaction effect are significant well beyond the .001 level for the conservative Geisser and Greenhouse test procedure (1 and 107 degrees of freedom for all tests.) Unfortunately, variance component estimates cannot be computed for the complete non-additive model. Components can be estimated if it is assumed that the subject factor (data sets) is strictly additive in the model. If such an assumption is made, estimates of strength of association are .024 for Methods, .448 for Conditions, and .075 for the Method by Condition interaction. Clearly, the Condition factor accounts for a very large proportion of variance in the data. However, if the additivity assumption is not justified, all estimates are negatively biased.
### TABLE 8

**Analysis of Variance - Rand Statistic**

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>D.F.</th>
<th>Mean Square</th>
<th>F-ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>238.741</td>
<td>10</td>
<td>23.874</td>
<td>511.188</td>
</tr>
<tr>
<td>Method</td>
<td>13.043</td>
<td>14</td>
<td>.932</td>
<td>33.138</td>
</tr>
<tr>
<td>Data Sets</td>
<td>40.061</td>
<td>107</td>
<td>.374</td>
<td></td>
</tr>
<tr>
<td>Condition x Method</td>
<td>41.410</td>
<td>140</td>
<td>.296</td>
<td>41.532</td>
</tr>
<tr>
<td>Data Sets x Method</td>
<td>42.114</td>
<td>1498</td>
<td>.028</td>
<td></td>
</tr>
<tr>
<td>Data Sets x Condition</td>
<td>49.972</td>
<td>1070</td>
<td>.047</td>
<td></td>
</tr>
<tr>
<td>Data Sets x Method x Condition</td>
<td>106.685</td>
<td>14980</td>
<td>.007</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** All computed F-ratios are significant beyond the .001 level for the conservative Geisser and Greenhouse level (1 and 107 degrees of freedom for all tests).
Because of the highly significant interaction effect, it was decided to conduct additional tests along the lines of simple main effects tests. Simple main effects were first computed for each condition across methods. The results of the simple main effects tests are presented in Table 9. All error conditions produced significant simple main effects at very conservative $\alpha$ levels. Using the Bonferroni inequality, each of the tests in the set of 11 error conditions are significant at the .01 level using the conservative degrees of freedom values. Strength of association estimates are also included in the table. The estimates assume a strictly additive model.

As a further data analysis, a Newman–Keuls post hoc test was performed for each error condition in order to determine significant pair-wise differences. The .05 $\alpha$ level of significance was used for each condition.

1. Error Free Condition. The best 10 methods were found to be non-significant in their pairwise mean differences. The complete set of equivalent methods is indicated by the asterisk notation in Table 7. The best three methods were the group-average, beta-flexible, and the complete linkage methods, respectively. The set of five methods which gave significantly lower recovery than the other 10 procedures included the four K-means algorithms and the minimum total sum of squares method.
TABLE 9  
Simple Main Effects Tests  
For Conditions Across Methods

<table>
<thead>
<tr>
<th>Condition</th>
<th>Mean Square</th>
<th>Mean Square Error</th>
<th>F-ratio</th>
<th>Omega-Squared$^a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Free</td>
<td>.15203</td>
<td>.00578</td>
<td>26.318</td>
<td>.157</td>
</tr>
<tr>
<td>20% Outlier</td>
<td>.65724</td>
<td>.00856</td>
<td>76.778</td>
<td>.299</td>
</tr>
<tr>
<td>40% Outlier</td>
<td>.94021</td>
<td>.00919</td>
<td>101.321</td>
<td>.357</td>
</tr>
<tr>
<td>Low Error</td>
<td>.18371</td>
<td>.00818</td>
<td>22.465</td>
<td>.128</td>
</tr>
<tr>
<td>High Error</td>
<td>.30842</td>
<td>.01449</td>
<td>26.854</td>
<td>.104</td>
</tr>
<tr>
<td>1-dim. Error</td>
<td>.18810</td>
<td>.01167</td>
<td>16.118</td>
<td>.079</td>
</tr>
<tr>
<td>2-dim. Error</td>
<td>.67265</td>
<td>.01420</td>
<td>47.369</td>
<td>.290</td>
</tr>
<tr>
<td>r</td>
<td>.07136</td>
<td>.00431</td>
<td>16.573</td>
<td>.053</td>
</tr>
<tr>
<td>Std. Scores</td>
<td>.15589</td>
<td>.00866</td>
<td>17.597</td>
<td>.103</td>
</tr>
<tr>
<td>Tp</td>
<td>.21789</td>
<td>.01136</td>
<td>19.187</td>
<td>.111</td>
</tr>
<tr>
<td>Random Noise</td>
<td>.31194</td>
<td>.00575</td>
<td>59.450</td>
<td>.297</td>
</tr>
</tbody>
</table>

NOTE: All mean squares are computed with 14 and 1496 degrees of freedom. All tests are significant beyond the .001 level for the conservative Geisser and Greenhouse level (1 and 107 degrees of freedom for all tests).

$^a$Omega-Squared values are exact estimates only when the assumption is made that a strictly additive design with respect to data sets holds. If the assumption is not justified, the proportion of variance estimates will be biased to an unknown degree.
2. **Low 20% outlier condition.** Only the best four methods were found to be statistically equivalent to each other. The best four methods were the centroid, single link, median, and group average procedures, respectively. The centroid method performed better than the lowest 10 methods. The other three methods performed better than the lowest nine methods. Jancey's method placed fifth and was significantly lower in mean recovery than the best four methods.

3. **High (40%) outlier condition.** Again, only the best four methods formed the equivalent group. The methods were the centroid, single link, Jancey, and median procedures. The centroid and single link procedures gave significantly better recovery than the methods ranking four through 15. Jancey's method and the median method performed better than the lowest 10 methods.

4. **Low error condition (error perturbation of the distances).** The best seven methods were found to be statistically equivalent. The best three methods were the group average, weighted average, and beta-flexible algorithms. The group average and weighted average performed better than the eight lowest methods while the beta-flexible performed better than the lowest seven methods. The remaining four methods in the set performed better than at least the lowest six methods.
5. **High error condition.** Again, the set of the best seven methods were found to be statistically equivalent. The best three procedures were the group average, beta-flexible, and Ward's minimum variance algorithms. All three of these methods performed better than the lowest eight methods. The remaining four methods in the set performed better than at least the lowest five methods.

6. **One-dimensional error condition.** The four highest ranking methods were found to be equivalent in terms of the pairwise mean differences. The four methods were the group average, weighted average, beta-flexible, and minimum average sum of squares procedures. These methods were found to be significantly better than the lowest 10, 9, 7, and 6 methods, respectively.

7. **Two-dimensional error condition.** Only the best four methods were found to be statistically equivalent. The four methods were the group average, weighted average, minimum average sum of squares, and beta-flexible algorithms. The group average method performed better than the lowest 11 methods, the weighted average procedure performed better than the nine lowest methods, while the minimum average sum of squares and the beta-flexible methods performed significantly better than the lowest six ranking algorithms.

8. **Correlation condition.** The best 10 methods were found to be non-significant in their pairwise differences. The best three were the beta-flexible, Ward's, and the group
average methods. All of the 10 best methods were found to be significantly better than the lowest five methods which included the K-means algorithms and the single linkage method.

9. **Standard scores condition.** The set of the best seven methods formed the statistically equivalent group. The best three methods were the beta-flexible, group average, and weighted average procedures. All seven methods were found to be significantly greater than the lowest eight methods.

10. **Cattell's \( r_p \) condition.** The best seven methods again formed the statistically equivalent group. The three ranking methods were the group average, weighted average, and complete link procedures. The top four methods performed better than the lowest eight methods. The next three methods performed significantly better than at least the four lowest ranking methods.

Overall, the simple main effects for conditions across methods indicated that in all conditions, the best four methods did not significantly differ from each other and in many conditions, the top group of equivalent methods was larger. Further, this set equivalent of methods as a whole tended to perform significantly better than the set of methods giving the lowest recovery rates. The last column in Table 7 indicates the number of times each method fell into the statistically equivalent superior group across the
10 error conditions. The best method was the group average procedure which fell into the group a total of nine times. The weighted average, beta-flexible, and minimum average sum of squares methods each placed in the equivalent group a total of eight times. On the other hand, three of the K-means procedures never fell into the equivalent group. However, as can be seen in Table 7, the actual set of best performing methods changed from condition to condition, thus confirming the presence of the method by condition interaction.

Simple main effects tests for each method across conditions were also conducted. The results are presented in Table 10. All methods produced significant simple main effects at very conservative α levels. Using the Bonferroni inequality, each of the tests in the set of 15 methods are significant at the .01 level using conservative degrees of freedom values.

As a further data analysis of the simple main effects for methods, it was deemed useful to determine whether the recovery performance in any error perturbation condition fell to chance levels. Thus, a post hoc test was carried out for each of the methods using a Dunnett's test for the control group. In the present study, the random noise condition served as the control group. The results indicated that in every case for each of the 15 methods, the recovery rates in the error perturbation conditions were significantly
<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Square</th>
<th>Mean Square Error</th>
<th>F-ratio</th>
<th>Omega-Squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Link</td>
<td>2.36224</td>
<td>.01295</td>
<td>137.898</td>
<td>.511</td>
</tr>
<tr>
<td>Complete Link</td>
<td>1.69935</td>
<td>.00961</td>
<td>184.432</td>
<td>.557</td>
</tr>
<tr>
<td>Group Average</td>
<td>1.86881</td>
<td>.00473</td>
<td>395.281</td>
<td>.743</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>1.77313</td>
<td>.00507</td>
<td>292.230</td>
<td>.675</td>
</tr>
<tr>
<td>Centroid Method</td>
<td>3.41693</td>
<td>.01664</td>
<td>205.404</td>
<td>.573</td>
</tr>
<tr>
<td>Median Method</td>
<td>3.02554</td>
<td>.01547</td>
<td>195.590</td>
<td>.571</td>
</tr>
<tr>
<td>Ward's Min. Variance</td>
<td>2.00389</td>
<td>.00738</td>
<td>264.462</td>
<td>.596</td>
</tr>
<tr>
<td>Beta-Flexible</td>
<td>1.87055</td>
<td>.00691</td>
<td>270.660</td>
<td>.634</td>
</tr>
<tr>
<td>Ave. Link in New Cluster</td>
<td>1.56663</td>
<td>.00747</td>
<td>249.791</td>
<td>.537</td>
</tr>
<tr>
<td>Min. Total Sum of Squares</td>
<td>1.65775</td>
<td>.00767</td>
<td>216.178</td>
<td>.415</td>
</tr>
<tr>
<td>Min. Ave. Sum of Squares</td>
<td>1.75830</td>
<td>.00748</td>
<td>235.070</td>
<td>.588</td>
</tr>
<tr>
<td>MacQueen's Method</td>
<td>1.04101</td>
<td>.01265</td>
<td>81.018</td>
<td>.362</td>
</tr>
<tr>
<td>Forgy's Method</td>
<td>1.19248</td>
<td>.01114</td>
<td>107.090</td>
<td>.414</td>
</tr>
<tr>
<td>Jancey's Method</td>
<td>1.34469</td>
<td>.00952</td>
<td>141.268</td>
<td>.467</td>
</tr>
<tr>
<td>Convergent K-means</td>
<td>1.23137</td>
<td>.01074</td>
<td>114.761</td>
<td>.418</td>
</tr>
</tbody>
</table>

**NOTE:** Mean squares are computed with 10 and 1070 degrees of freedom. All tests are significant beyond the .001 level for the conservative Geisser and Greenhouse level (1 and 107 degrees of freedom for all tests).
greater than the base line rates obtained from the random noise condition. The results held when the .01 α level was used for each method. Thus, even for the methods which gave the lowest recovery rate in each condition, the methods were still recovering a statistically significant amount of cluster structure from the data.

**Internal Criterion Measures**

**Single Partition Cophenetic Coefficient**

**Descriptive statistics.** The means and standard deviations for the single partition cophenetic correlations are presented in Table 11 for the hierarchical methods and Table 12 for the non-hierarchical methods. A visual inspection of the tables indicates that for any given method, the mean correlations in the error perturbation conditions are noticeably greater than the mean correlations obtained from the random noise condition. As expected, the random noise mean correlations are positive and are anywhere from two to five standard deviations above zero.

The correlation between the single partition cophenetic correlation and the Rand statistic overall data sets, conditions, and methods was .84. The average within-cell correlation was .66. Since the overall correlation was based on 17,820 observations, the correlation is significantly different from zero at any traditional α level. Thus, about 70% of the variance in the internal criterion can be accounted for by the variance in the external criterion.
<table>
<thead>
<tr>
<th>Method</th>
<th>Error Condition</th>
<th>20% Outlier</th>
<th>40% Outlier</th>
<th>Low Error</th>
<th>High Error</th>
<th>1-dim. Error</th>
<th>2-dim Error</th>
<th>r Std. Scores</th>
<th>r P Random Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Link</td>
<td>.834 (.169)</td>
<td>.824 (.145)</td>
<td>.801 (.145)</td>
<td>.703 (.278)</td>
<td>.524 (.313)</td>
<td>.679 (.186)</td>
<td>.589 (.197)</td>
<td>.789 (.192)</td>
<td>.771 (.223)</td>
</tr>
<tr>
<td>Complete Link</td>
<td>.854 (.079)</td>
<td>.715 (.131)</td>
<td>.587 (.117)</td>
<td>.767 (.172)</td>
<td>.592 (.231)</td>
<td>.700 (.102)</td>
<td>.618 (.120)</td>
<td>.809 (.111)</td>
<td>.825 (.103)</td>
</tr>
<tr>
<td>Group Average</td>
<td>.855 (.075)</td>
<td>.817 (.095)</td>
<td>.742 (.115)</td>
<td>.788 (.114)</td>
<td>.643 (.189)</td>
<td>.735 (.091)</td>
<td>.665 (.098)</td>
<td>.815 (.102)</td>
<td>.828 (.102)</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>.850 (.100)</td>
<td>.789 (.116)</td>
<td>.705 (.131)</td>
<td>.788 (.114)</td>
<td>.634 (.204)</td>
<td>.729 (.093)</td>
<td>.646 (.118)</td>
<td>.807 (.123)</td>
<td>.830 (.109)</td>
</tr>
<tr>
<td>Centroid Method</td>
<td>.838 (.145)</td>
<td>.834 (.131)</td>
<td>.813 (.134)</td>
<td>.758 (.206)</td>
<td>.563 (.293)</td>
<td>.639 (.212)</td>
<td>.410 (.230)</td>
<td>.809 (.122)</td>
<td>.756 (.246)</td>
</tr>
<tr>
<td>Median Method</td>
<td>.829 (.163)</td>
<td>.816 (.142)</td>
<td>.761 (.147)</td>
<td>.768 (.191)</td>
<td>.565 (.194)</td>
<td>.621 (.205)</td>
<td>.427 (.236)</td>
<td>.801 (.143)</td>
<td>.784 (.208)</td>
</tr>
<tr>
<td>Ward's Min. Variance</td>
<td>.846 (.104)</td>
<td>.571 (.108)</td>
<td>.483 (.071)</td>
<td>.785 (.123)</td>
<td>.628 (.159)</td>
<td>.699 (.116)</td>
<td>.633 (.098)</td>
<td>.804 (.111)</td>
<td>.821 (.104)</td>
</tr>
<tr>
<td>Beta-Flexible</td>
<td>.894 (.078)</td>
<td>.658 (.113)</td>
<td>.549 (.097)</td>
<td>.786 (.121)</td>
<td>.632 (.193)</td>
<td>.721 (.096)</td>
<td>.637 (.106)</td>
<td>.808 (.114)</td>
<td>.827 (.109)</td>
</tr>
<tr>
<td>Ave. Link in the</td>
<td>.840 (.104)</td>
<td>.719 (.127)</td>
<td>.641 (.127)</td>
<td>.773 (.130)</td>
<td>.618 (.205)</td>
<td>.700 (.124)</td>
<td>.625 (.110)</td>
<td>.799 (.116)</td>
<td>.808 (.111)</td>
</tr>
<tr>
<td>New Cluster</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min. Total Sum of</td>
<td>.772 (.199)</td>
<td>.440 (.068)</td>
<td>.354 (.046)</td>
<td>.676 (.223)</td>
<td>.521 (.233)</td>
<td>.644 (.149)</td>
<td>.528 (.163)</td>
<td>.794 (.123)</td>
<td>.723 (.211)</td>
</tr>
<tr>
<td>Squares</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min. Ave. Sum of</td>
<td>.851 (.078)</td>
<td>.648 (.116)</td>
<td>.548 (.088)</td>
<td>.781 (.130)</td>
<td>.620 (.200)</td>
<td>.715 (.103)</td>
<td>.643 (.097)</td>
<td>.805 (.114)</td>
<td>.821 (.107)</td>
</tr>
<tr>
<td>Squares</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** First entry is the mean, parenthetical entry is the standard deviation.
<table>
<thead>
<tr>
<th>Method</th>
<th>Error Free</th>
<th>20% Outlier</th>
<th>40% Outlier</th>
<th>Low Error</th>
<th>High Error</th>
<th>1-dim. Error</th>
<th>2-dim. Error</th>
<th>r</th>
<th>Std. Scores</th>
<th>p</th>
<th>Random Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacQueen's Method</td>
<td>.735 (.159)</td>
<td>.728 (.195)</td>
<td>.691 (.276)</td>
<td>.703 (.138)</td>
<td>.630 (.166)</td>
<td>.615 (.141)</td>
<td>.573 (.125)</td>
<td>.731 .694</td>
<td>.702 (.158)</td>
<td>.398 (.078)</td>
<td></td>
</tr>
<tr>
<td>Forgy's Method</td>
<td>.780 (.142)</td>
<td>.763 (.196)</td>
<td>.757 (.216)</td>
<td>.745 (.152)</td>
<td>.663 (.159)</td>
<td>.671 (.113)</td>
<td>.592 (.113)</td>
<td>.752 .744</td>
<td>.746 (.149)</td>
<td>.398 (.068)</td>
<td></td>
</tr>
<tr>
<td>Jancey's Method</td>
<td>.779 (.153)</td>
<td>.817 (.143)</td>
<td>.793 (.213)</td>
<td>.765 (.165)</td>
<td>.706 (.142)</td>
<td>.699 (.101)</td>
<td>.626 (.101)</td>
<td>.771 .784</td>
<td>.787 (.138)</td>
<td>.427 (.064)</td>
<td></td>
</tr>
<tr>
<td>Convergent K-means</td>
<td>.759 (.153)</td>
<td>.753 (.194)</td>
<td>.767 (.238)</td>
<td>.729 (.157)</td>
<td>.692 (.145)</td>
<td>.675 (.110)</td>
<td>.590 (.118)</td>
<td>.758 .754</td>
<td>.753 (.136)</td>
<td>.401 (.066)</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** First entry is the mean, parenthetical entry is the standard deviation.
Table 13 presents the correlation between the external and internal criterion for each of the methods. The correlations exhibit some limited variation which ranges from .743 to .886.

Hypothesis tests. A one-way repeated measures analysis of variance was performed for each method. The results of the tests are presented in Table 14. Since both the single partition coefficient and the hierarchical coefficient (reported below) are Pearson correlations, an arc-sine transformation seemed appropriate. However, as with the external criterion measure, no differences in terms of final inferences were found between the transformed and raw scores. All results reported below are given in terms of raw scores.

The hypothesis of equal mean correlations across conditions was rejected for all 15 methods using conservative degrees of freedom levels. Each test in the set of 15 is significant when the overall error rate was set to the .01 level by using the Bonferroni inequality. Strength of association estimates are also presented in the table. The estimates require the assumption of additivity in the model. However, the estimates seem to indicate that substantial proportions of variance are accounted for by the condition factor within each method. Post hoc tests were conducted for each method in order to determine whether the mean correlations in the error perturbation conditions were significantly greater than the base line rates obtained from
<table>
<thead>
<tr>
<th>Method</th>
<th>Single Partition Criterion</th>
<th>Hierarchical Criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Link</td>
<td>.866</td>
<td>.788</td>
</tr>
<tr>
<td>Complete Link</td>
<td>.839</td>
<td>.689</td>
</tr>
<tr>
<td>Group Average</td>
<td>.743</td>
<td>.681</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>.781</td>
<td>.704</td>
</tr>
<tr>
<td>Centroid Method</td>
<td>.883</td>
<td>.800</td>
</tr>
<tr>
<td>Median Method</td>
<td>.865</td>
<td>.792</td>
</tr>
<tr>
<td>Ward's Min. Variance</td>
<td>.816</td>
<td>.542</td>
</tr>
<tr>
<td>Beta-Flexible</td>
<td>.794</td>
<td>.581</td>
</tr>
<tr>
<td>Ave. Link in New Cluster</td>
<td>.803</td>
<td>.686</td>
</tr>
<tr>
<td>Min. Total Sum of Squares</td>
<td>.886</td>
<td>.666</td>
</tr>
<tr>
<td>Min. Ave. Sum of Squares</td>
<td>.801</td>
<td>.615</td>
</tr>
<tr>
<td>MacQueen's Method</td>
<td>.843</td>
<td>-</td>
</tr>
<tr>
<td>Forgy's Method</td>
<td>.835</td>
<td>-</td>
</tr>
<tr>
<td>Jancey's Method</td>
<td>.838</td>
<td>-</td>
</tr>
<tr>
<td>Convergent K-means</td>
<td>.834</td>
<td>-</td>
</tr>
<tr>
<td>Method</td>
<td>Mean Square</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Single Link</td>
<td>3.95932</td>
<td>.02131</td>
</tr>
<tr>
<td>Complete Link</td>
<td>2.39498</td>
<td>.01100</td>
</tr>
<tr>
<td>Group Average</td>
<td>1.95699</td>
<td>.00722</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>2.14591</td>
<td>.00870</td>
</tr>
<tr>
<td>Centroid Method</td>
<td>4.26628</td>
<td>.02569</td>
</tr>
<tr>
<td>Median Method</td>
<td>4.32097</td>
<td>.02647</td>
</tr>
<tr>
<td>Ward's Min. Variance</td>
<td>2.56302</td>
<td>.00848</td>
</tr>
<tr>
<td>Beta-Flexible</td>
<td>2.24816</td>
<td>.00781</td>
</tr>
<tr>
<td>Ave. Link in New Cluster</td>
<td>1.77741</td>
<td>.00881</td>
</tr>
<tr>
<td>Med. Total Sum of Squares</td>
<td>3.25455</td>
<td>.01639</td>
</tr>
<tr>
<td>Min. Ave. Sum of Squares</td>
<td>2.17198</td>
<td>.00896</td>
</tr>
<tr>
<td>MacQueen's Method</td>
<td>1.13666</td>
<td>.02244</td>
</tr>
<tr>
<td>Forgy's Method</td>
<td>1.37081</td>
<td>.01624</td>
</tr>
<tr>
<td>Lance's Method</td>
<td>1.36395</td>
<td>.01241</td>
</tr>
<tr>
<td>Convergent K-means</td>
<td>1.32283</td>
<td>.01637</td>
</tr>
</tbody>
</table>

**NOTE:** Mean squares are computed with 10 and 1070 degrees of freedom. All tests are significant beyond the .001 level for the conservative Geisser and Greenhouse level (1 and 107 degrees of freedom for all tests).
the random noise condition. Thus, a Dunnett's test for the control group was conducted for each method. In every case, the perturbation condition means were significantly greater than the random noise condition means when the α level was set at .05 for each method. All tests remained significant at the .01 level except for a single comparison involving the 40% outlier condition for the minimum total sum of squares method. Overall, the results are in good accord with the control group comparisons conducted with the external criterion measure.

Hierarchical Cophenetic Coefficient

Descriptive statistics. The means and standard deviations for the hierarchical cophenetic correlations are presented in Table 15. As noted earlier, the index could only be computed for the 11 hierarchical methods. A visual inspection of the table indicates that for any given method, the mean error perturbation correlations are greater than the mean correlations obtained from the random noise condition. Again, the random noise correlation means are positive and noticeably greater than zero.

The correlation between the hierarchical cophenetic index and the Rand statistic overall data sets, conditions, and methods was .70. The average within cell correlation was .47. Both correlations are less than the values obtained for the single partition measure. The result seems reasonable in that the hierarchical coefficient is a measure
### TABLE 15
Mean Hierarchical Cophenetic Correlations

<table>
<thead>
<tr>
<th>Method</th>
<th>Error Free</th>
<th>25% Outlier</th>
<th>40% Outlier</th>
<th>Low Error</th>
<th>High Error</th>
<th>1-dim. Error</th>
<th>2-dim. Error</th>
<th>r</th>
<th>Std. Scores</th>
<th>P</th>
<th>Random Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Link</td>
<td>.908 (.066)</td>
<td>.901 (.067)</td>
<td>.899 (.067)</td>
<td>.818 (.139)</td>
<td>.647 (.231)</td>
<td>.802 (.084)</td>
<td>.714 (.096)</td>
<td>.872 (.128)</td>
<td>.878 (.093)</td>
<td>.895 (.097)</td>
<td>.447 (.062)</td>
</tr>
<tr>
<td>Complete Link</td>
<td>.909 (.061)</td>
<td>.889 (.074)</td>
<td>.882 (.088)</td>
<td>.825 (.148)</td>
<td>.684 (.201)</td>
<td>.803 (.072)</td>
<td>.731 (.086)</td>
<td>.896 (.075)</td>
<td>.878 (.083)</td>
<td>.888 (.093)</td>
<td>.517 (.049)</td>
</tr>
<tr>
<td>Group Average</td>
<td>.925 (.047)</td>
<td>.923 (.049)</td>
<td>.922 (.048)</td>
<td>.863 (.104)</td>
<td>.745 (.161)</td>
<td>.851 (.058)</td>
<td>.796 (.060)</td>
<td>.919 (.057)</td>
<td>.905 (.063)</td>
<td>.933 (.059)</td>
<td>.597 (.037)</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>.919 (.054)</td>
<td>.903 (.028)</td>
<td>.888 (.092)</td>
<td>.858 (.105)</td>
<td>.739 (.165)</td>
<td>.836 (.064)</td>
<td>.779 (.071)</td>
<td>.910 (.073)</td>
<td>.896 (.068)</td>
<td>.923 (.069)</td>
<td>.559 (.042)</td>
</tr>
<tr>
<td>Centroid Method</td>
<td>.908 (.068)</td>
<td>.908 (.069)</td>
<td>.907 (.151)</td>
<td>.834 (.215)</td>
<td>.677 (.110)</td>
<td>.799 (.127)</td>
<td>.665 (.099)</td>
<td>.915 (.106)</td>
<td>.873 (.121)</td>
<td>.875 (.576)</td>
<td>.428 (.104)</td>
</tr>
<tr>
<td>Median Method</td>
<td>.891 (.097)</td>
<td>.881 (.098)</td>
<td>.866 (.135)</td>
<td>.822 (.226)</td>
<td>.657 (.112)</td>
<td>.750 (.165)</td>
<td>.624 (.073)</td>
<td>.902 (.109)</td>
<td>.862 (.140)</td>
<td>.355 (.084)</td>
<td>.496 (.046)</td>
</tr>
<tr>
<td>Ward’s Min. Variance</td>
<td>.885 (.074)</td>
<td>.886 (.067)</td>
<td>.886 (.118)</td>
<td>.824 (.183)</td>
<td>.691 (.086)</td>
<td>.780 (.081)</td>
<td>.718 (.079)</td>
<td>.886 (.082)</td>
<td>.853 (.104)</td>
<td>.855 (.092)</td>
<td>.469 (.046)</td>
</tr>
<tr>
<td>Beta-Flexible</td>
<td>.904 (.054)</td>
<td>.895 (.062)</td>
<td>.898 (.057)</td>
<td>.841 (.112)</td>
<td>.713 (.172)</td>
<td>.813 (.067)</td>
<td>.750 (.072)</td>
<td>.903 (.067)</td>
<td>.875 (.076)</td>
<td>.892 (.046)</td>
<td>.528 (.046)</td>
</tr>
<tr>
<td>Ave. Link in the</td>
<td>.880 (.089)</td>
<td>.899 (.101)</td>
<td>.898 (.109)</td>
<td>.841 (.129)</td>
<td>.715 (.173)</td>
<td>.804 (.094)</td>
<td>.748 (.084)</td>
<td>.888 (.096)</td>
<td>.855 (.097)</td>
<td>.873 (.094)</td>
<td>.552 (.041)</td>
</tr>
<tr>
<td>New Cluster</td>
<td>.795 (.158)</td>
<td>.785 (.159)</td>
<td>.758 (.184)</td>
<td>.723 (.193)</td>
<td>.580 (.225)</td>
<td>.660 (.151)</td>
<td>.557 (.158)</td>
<td>.852 (.111)</td>
<td>.728 (.167)</td>
<td>.738 (.177)</td>
<td>.320 (.083)</td>
</tr>
<tr>
<td>Min. Total Sum of</td>
<td>.795 (.075)</td>
<td>.891 (.097)</td>
<td>.882 (.132)</td>
<td>.828 (.176)</td>
<td>.707 (.081)</td>
<td>.811 (.011)</td>
<td>.756 (.078)</td>
<td>.895 (.071)</td>
<td>.860 (.088)</td>
<td>.879 (.093)</td>
<td>.542 (.045)</td>
</tr>
<tr>
<td>Squares</td>
<td>.795 (.075)</td>
<td>.891 (.097)</td>
<td>.882 (.132)</td>
<td>.828 (.176)</td>
<td>.707 (.081)</td>
<td>.811 (.011)</td>
<td>.756 (.078)</td>
<td>.895 (.071)</td>
<td>.860 (.088)</td>
<td>.879 (.093)</td>
<td>.542 (.045)</td>
</tr>
</tbody>
</table>

**NOTE:** First entry in the mean, parenthetical entry is the standard deviation.
of the fit of the entire hierarchy structure rather than the recovery at a specific level as with the single partition coefficient. Thus, the obtained correlation of .70 indicates that only about half of the variance in the hierarchical measure can be accounted for by the variance in the Rand statistic. Hence, the hierarchical coefficient seems to be a less useful index of correct cluster recovery when the primary emphasis is in terms of a single partition. The correlations between the hierarchical coefficient and the Rand statistic for each method are also presented in Table 11. Somewhat more variation is seen in these correlations than for the single partition index. The correlations range from .542 to .792.

**Hypothesis tests.** A one-way repeated measures analysis of variance was performed for each method. The results of the test are presented in Table 16. Again, the hypothesis of equal mean correlations across conditions was rejected for all 15 methods using conservative degrees of freedom levels. Each test in the set of 15 is significant when the overall error rate is set to the .01 level with the Bonferroni inequality. The proportion of variance estimates are also quite substantial.

As with the single partition cophenetic correlation, a Dunnett's test for the control group was conducted for each method. Again, in every case the perturbation correlation means were significantly greater than the random error
<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Square</th>
<th>Mean Square Error</th>
<th>F-ratio</th>
<th>Omega-Squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Link</td>
<td>2.21140</td>
<td>.00731</td>
<td>302.454</td>
<td>.583</td>
</tr>
<tr>
<td>Complete Link</td>
<td>1.59093</td>
<td>.00566</td>
<td>281.305</td>
<td>.560</td>
</tr>
<tr>
<td>Group Average</td>
<td>1.17790</td>
<td>.00273</td>
<td>430.679</td>
<td>.634</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>1.30185</td>
<td>.00348</td>
<td>374.233</td>
<td>.605</td>
</tr>
<tr>
<td>Centroid Method</td>
<td>2.50967</td>
<td>.00648</td>
<td>387.312</td>
<td>.615</td>
</tr>
<tr>
<td>Median Method</td>
<td>3.00127</td>
<td>.00878</td>
<td>341.931</td>
<td>.617</td>
</tr>
<tr>
<td>Ward’s Min. Variance</td>
<td>1.58988</td>
<td>.00439</td>
<td>362.086</td>
<td>.592</td>
</tr>
<tr>
<td>Beta-Flexible</td>
<td>1.47064</td>
<td>.00368</td>
<td>399.924</td>
<td>.632</td>
</tr>
<tr>
<td>Ave. Link in New Cluster</td>
<td>1.13306</td>
<td>.00998</td>
<td>300.069</td>
<td>.478</td>
</tr>
<tr>
<td>Min. Total Sum of Squares</td>
<td>2.39911</td>
<td>.01217</td>
<td>197.215</td>
<td>.416</td>
</tr>
<tr>
<td>Min. Ave. Sum of Squares</td>
<td>1.29220</td>
<td>.00397</td>
<td>325.651</td>
<td>.528</td>
</tr>
</tbody>
</table>

NOTE: Mean squares are computed with 10 and 1070 degrees of freedom. All tests are significant beyond the .001 level for the conservative Geisser and Greenhouse level (1 and 107 degrees of freedom for all tests).
condition means when the α level was set to .01 for each method. The results indicate that the methods were recovering a significant amount of hierarchy structure in all 10 error perturbation conditions.

**Correlation Measure of the Error Perturbation Process**

The mean error perturbation correlations along with standard deviations are presented in Table 17. These values represent the mean correlations between the error free condition data sets and the same data sets after the error perturbation process. The correlations could not be meaningfully defined for the two outlier conditions.

The low error, r, standard scores, and \( r_p \) condition gave fairly high mean correlations, thus indicating that these types or levels of error perturbation had only a slight to moderate effect on the data. The high error, 1-dimensional, and 2-dimensional error conditions produced lower mean correlations. The result seems consistent with the nature of the error conditions. Also presented in Table 17 are the mean recovery rates and exact recovery rates overall methods within error condition. A very high correlation was found between the mean error perturbation correlations and the overall mean recovery (.979) and also with the exact rates (.856). Thus, the error perturbation process was apparently directly related to cluster recovery to a great extent.
### TABLE 17

Mean Error Perturbation Correlations

<table>
<thead>
<tr>
<th>Error Condition</th>
<th>Correlation</th>
<th>Standard Deviation</th>
<th>Mean Overall Recovery</th>
<th>Exact Overall Recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Free</td>
<td>-</td>
<td>-</td>
<td>.964</td>
<td>1288</td>
</tr>
<tr>
<td>20% Outliers</td>
<td>-</td>
<td>-</td>
<td>.894</td>
<td>437</td>
</tr>
<tr>
<td>40% Outliers</td>
<td>-</td>
<td>-</td>
<td>.861</td>
<td>270</td>
</tr>
<tr>
<td>Low Error</td>
<td>.926</td>
<td>.086</td>
<td>.950</td>
<td>1152</td>
</tr>
<tr>
<td>High Error</td>
<td>.779</td>
<td>.185</td>
<td>.883</td>
<td>649</td>
</tr>
<tr>
<td>1-dim. Error</td>
<td>.857</td>
<td>.066</td>
<td>.864</td>
<td>558</td>
</tr>
<tr>
<td>2-dim. Error</td>
<td>.767</td>
<td>.091</td>
<td>.807</td>
<td>394</td>
</tr>
<tr>
<td>r</td>
<td>.918</td>
<td>.044</td>
<td>.948</td>
<td>1099</td>
</tr>
<tr>
<td>Std. Scores</td>
<td>.899</td>
<td>.081</td>
<td>.946</td>
<td>1109</td>
</tr>
<tr>
<td>r_p</td>
<td>.944</td>
<td>.040</td>
<td>.935</td>
<td>1071</td>
</tr>
<tr>
<td>Random Noise</td>
<td>.001</td>
<td>.040</td>
<td>.537</td>
<td>0</td>
</tr>
</tbody>
</table>
IV. DISCUSSION

Evaluation of Recovery Performance

Perhaps the most disturbing aspect of the results of the present study with regard to correct cluster recovery is the strong method by error condition interaction for the external criterion. The interaction appears to be somewhat disordinal and thus implies that the set of best methods changes from one type of error perturbation condition to the next. Of course, the ideal state would be a complete lack of any method by condition interaction. Fortunately, it seems that at least one method, the group average method, did exhibit some robustness to all types of error perturbation except the outlier conditions. As indicated in Table 7, the group average method placed in the equivalent superior group in nine out of 10 possible cases. The weighted average, beta-flexible, and minimum average sum of squares methods each placed in the equivalent group a total of 8 times. However, the beta-flexible and minimum average sum of squares methods performed extremely poorly in the outlier conditions. The two methods did not even rank among the best 10 algorithms in these conditions. Thus, general usage of these two methods seems unwarranted. It is also interesting to note that the methods which have been more traditionally
used in Psychology (single linkage, complete linkage, and Ward's method) are not among the best four methods in terms of the number of times the procedures placed in the statistically equivalent superior group.

This aspect of nonrobustness has been appearing on a regular basis in the other Monte Carlo studies of clustering algorithms. Edelbrock (1977) in a study of five hierarchical methods also obtained a higher order interaction between methods and other design variables. As noted earlier in the paper, Milligan and Isaac (1978) found a rank ordering for four hierarchical methods tested with simulated ultrametric data which differed from the ordering obtained from the multivariate normal mixture studies. In fact, Edelbrock (1977) used a subset of the same data which Blashfield (1976) used in his experiment. However, Edelbrock obtained a performance rank ordering much more similar to the rank ordering found by Milligan and Isaac than by Blashfield.

Although the present study used truncated multivariate normal mixtures to generate the point locations, the rank order performance for the same four hierarchical methods much more closely matches the rank ordering of the ultrametric study than the multivariate mixture studies. Apparently, the major confounding factor is that the multivariate normal mixture studies did not insure that the
clusters exhibited at least some limited level of external isolation whereas the ultrametric study and the present report did insure a degree of between cluster separation.

Several observations can be made with regard to the results obtained from the present study. It seems clear that the hierarchical methods tend to perform better as a group than the non-hierarchical procedures in virtually all error conditions. Within the non-hierarchical group, Jancey's method appears to be the best. The K-means solutions are not unique since randomly selected seed points were used as starting centroids. Thus, different seeds would produce different solutions. However, on an a priori basis better guesses for seed points are not usually available in applied research. It seems that for the K-means methods to be useful, good seed point estimates must be used. Of course, robustness to the selection of seed points could be used as an additional criterion for the selection of the best K-means method. The present study did not systematically examine this feature of the algorithms. However, both approaches, hierarchical and non-hierarchical, require the specification or selection of the correct partition level in terms of the number of clusters. Of course, the results for the K-means algorithms may or may not extend to other non-hierarchical procedures. Blashfield (1977a) performed a small Monte Carlo experiment with 20 multivariate normal mixtures and four non-hierarchical algorithms with two variants on
each algorithm. The results indicated that the methods which can use $|W|$ as a minimization criterion (Friedman and Rubin, 1967; MacRae, 1971) may perform better than the K-means procedures. However, the highest reported median Kappa value did not exceed the highest Kappa value reported by Blashfield (1976) in his study of four hierarchical methods.

When considering the error conditions, it seems apparent that the conditions involving data reduction (standard scores, $r$, $r_p$ conditions) represent fairly mild forms of error perturbation. Rohlf and Sokal (1965) did suggest that results obtained for $r_p$ and regular Euclidean distance should be essentially the same since the two measures are "practically" linearly related. However, the $r_p$ condition did in general produce a mild decrease in recovery performance for most methods. Although many authors have stressed the fact that great care must be taken when selecting a proximity measure (Cronbach and Gleser, 1953; Fleiss and Zubin, 1969; and others), it appears that the algorithms are somewhat robust to this feature of the clustering process. This result also parallels the findings by Edelbrock (1977) where it was found that the correlation coefficient works just as well, if not better than the Euclidean distance measure for multivariate normal mixtures. The Monte Carlo
results seem to be clearly indicating that the choice of the clustering algorithm is much more important than the choice of the similarity coefficient.

The other types of error perturbation apparently have substantially more serious effects. All 15 methods exhibited marked decrements in recovery performance in the condition which involved the addition of just one dimension (variable) of random noise to the basic set of defining dimensions. The results indicate that a researcher should be particularly cautious when selecting the variables to be used in the clustering process.

When the distances themselves are error perturbed, the decrement in recovery performance is directly related to the degree of error added. Further, there seems to be a differential reaction to this condition by the algorithms. When compared to their respective error free rates, some methods, particularly the K-means procedures, seem to be virtually unaffected by the error perturbation of the distances. The hierarchical methods exhibit slight to moderate decrements in mean recovery. At least one hierarchical method, the single linkage algorithm, is strongly affected by this type of error perturbation. With regard to applied research, the intrinsic error level of the data is usually not under control of the researcher. The only comment which can be made is that the less error in the data, the more likely that the correct structure will be recovered. The clustering methods
which exhibited some robustness to this type of error perturbation should probably be used for most data analysis purposes.

Perhaps the most interesting error condition in terms of the simulation results was the outlier condition. Here again, the non-hierarchical methods were virtually unaffected by the addition of outliers or intermediates to the data whether exact recovery or mean recovery is considered. In fact, the K-means procedures gave better recovery in general with respect to exact recovery than the hierarchical methods. However, when mean rates were considered, several hierarchical methods produced significantly higher mean rates than the non-hierarchical methods. It should be remembered that the addition of outliers was accompanied by a decrease in mean recovery for the hierarchical methods when compared to the error-free rates. This decrease in some instances was quite large as for the complete link, Ward's method, minimum total and minimum average sum of squares methods.

The most fascinating result which occurred in the outlier conditions was that the single linkage method, long thought to be extremely sensitive to outliers in the data, showed only fairly slight decreases in mean recovery. This result is in direct conflict with most of the findings and literature on clustering (Cormack 1971; Lance and Williams, 1967a; Wishart, 1969; among other).
At least two potential explanations can be offered for the observed result. First, it is possible that the actual number of outliers or intermediates present in the constructed data was not sufficiently great for the single linkage method to exhibit the chaining effect. A much greater concentration of outliers would very likely reduce the distinctiveness of the cluster to such a degree that the chaining effect would appear. Many real life data sets may possess very high concentrations of outliers and intermediates. Since most researchers commenting on the single linkage algorithm are reporting their experience with real life data sets, then the inconsistency may be due to this feature of the data generation process. However, it should be noted that the outlier condition did adversely affect some of the other algorithms.

An alternative explanation of the observed result is that the observations made in the clustering literature about the single linkage method may not necessarily be accurate. To be sure, the method does have a chaining characteristic. However, the ascribed cause may not be correct. The single linkage method may in fact be fairly robust to the addition of outliers or intermediates. Rather, the observed chaining effect may be due to another type of error perturbation. Rather, the present research showed that the method was particularly sensitive to the error perturbation of the interpoint distances. This result was previously
confirmed in the Milligan and Isaac (1978) study where it was found that the single linkage algorithm showed a marked decrease in recovery performance in very mild error conditions long before any other method was affected at all. The single linkage method also performed poorly in the multivariate normal mixture studies. In the Blashfield (1976) and Edelbrock (1977) studies, all distances were error perturbed to some degree, thus unknowingly biasing the data against the single linkage method. It is true that in the other multivariate normal mixture studies, such as Kuiper and Fisher (1975), the distances were not error perturbed. However, a considerable amount of cluster overlap was present in the data sets used in these studies which would definitely facilitate any chaining tendency and hence reduce recovery.

In any event, since real life data and particularly behavioral measurements are likely to possess at least some mild level of perturbation of the true "Psychological" distances, then the single linkage method would still be undesirable for general use.

In applied research settings, since outliers or intermediates form discrete units, the researcher has the potential for removing them from a data set. The present simulation results indicate that the decrement in recovery is directly related to the percentage of outliers present. Thus, the removal of outliers should provide an improved chance of correct cluster recovery. The major problem in
this approach is the identification of the outliers. True outliers, points which fall outside the major portion of the data mass are probably easier to identify than intermediates between clusters. The generalized distance measure is one potential index which could be used to identify outliers and assign probability values to them. This obviously requires the assumption of multivariate normality. However, it is probably the intermediates between cluster which cause the greatest decrement in correct cluster recovery. Identification of these elements is also more difficult. Perhaps some type of iterative process might prove useful. An initial clustering obtained from one of the better algorithms may be used to provide centroids which can in turn be used with some type of distance measure in order to identify outliers to any given cluster. Those elements which are identified as outliers to all clusters could then be eliminated from the data set or at least put in a residual pool. The data could then be reclustered, possibly with a different algorithm.

An alternative procedure would involve using a hierarchical method and eliminating all points which remain as single point clusters below a certain partition level in the hierarchy. Other strategies could also be proposed. Clearly, a variety of procedures should be examined in Monte Carlo experiments and some method should be adopted for applied research.
Although the Dunnett's control group tests indicated that in every case the difference between the perturbation condition means and the random noise means was statistically significant for each method, the question of practical significance is of interest. In absolute terms, mean Rand statistic values above .90 would indicate fairly good recovery. It seems likely that in the conditions where the methods produced high mean recovery values the researcher would have little difficulty in interpreting the structure of the data, if the structure can be interpreted at all. Since the mild error conditions (error free, standard scores, r, r_p) uniformly produced recovery means above .90, then most of the clustering algorithms would be useful in detecting the structure in this type of data. When considering the remaining perturbation conditions, only the best algorithms should be used. These methods usually do possess mean statistic values above .90. Although mean recovery values below .80 were found to be highly statistically significant in the present study, the obtained solutions probably diffuse the true data structure to the extent that an applied researcher would have some difficulty in interpreting or detecting the underlying structure. It also seems more likely that an incorrect interpretation could occur. For example, a Rand statistic value of .75 for the data sets used in the present study indicates that about 306 pairwise misclassifications occurred out of a total possible 1,225
pairs. Thus, a researcher needs to take considerable care in selecting the algorithm to be used when it is suspected that the data may possess a significant percentage of outliers or high levels of interpoint distance perturbation. In addition, the inclusion of just one random noise dimension can have a fairly serious effect on recovery. The inclusion of two random noise dimensions caused six methods to give recovery means below .80. In fact, only one method, the group average algorithm, produced a mean recovery value above .90 (.903). Overall, the group average procedure seems to give about the best recovery performance. The method is somewhat weak in the outlier conditions.

**Evaluation of Internal Criteria**

The results for the single partition cophenetic correlation coefficient are encouraging. The substantial correlation of .84 between the index and the external criterion indicates that the coefficient can be used to indicate when correct cluster recovery has occurred. The success of the statistic in the present study seems to justify the use of the index as a test statistic to test the hypothesis of no cluster structure in the data. A simple procedure would involve conducting a small Monte Carlo study to generate the null distribution values for any given application.

It should be noted that a large number of internal criteria exist which have been proposed in the clustering literature. Rohlf (1974) lists almost twenty possible indices
and his listing is not complete. Another review can be found in Hubert and Levin (1976). An obvious research opportunity would involve a detailed Monte Carlo experiment of many of these indices in order to determine if better recovery measures are available. The development and validation of a measure which can be formulated into a hypothesis test framework would be a substantial contribution to the field of clustering.

Since the hierarchical methods appear to be the most promising, perhaps indices can be developed or adopted which uses the information based on successive hierarchy partitions as recovery measures. Unusually large decrements in a test statistic's value across two partitions would indicate that an incorrect cluster merger has occurred in the hierarchy, or that the clustering process is attempting to produce a solution with too few clusters for the true structure of the data. Other approaches or modifications are possible. For example, an index could be developed which is analogous to the constraint measure presented by Isaac and Poor (1974) for the use in detecting true dimensionality in multidimensional scaling solutions. Essentially, the constraint index involves computing the difference between the expected value of the statistic when the data is random (lacking structure for the dimensionality) and the observed value of the statistic for the data set at hand. In the clustering application, an index using this
approach would achieve its maximum value for the partition level which would correspond to the optimal cluster solution. Again, research on these and other approaches would be worthwhile.

Results for the hierarchical cophenetic coefficient are somewhat less encouraging. Even if the index was as highly correlated with the external criterion as the single partition measure, the problem of selecting the appropriate partition would remain. Since most researchers in the social sciences are searching for the best single partition rather than the fit of the entire hierarchy, then this type of index would logically be less satisfactory and also less often used. If the entire hierarchy structure is meaningful then the present report deals only tangentially with the problem. The Rand statistic as used in the present study is only a single partition measure. A more direct experiment should be conducted which constructs a complete hierarchy (such as in the Milligan and Isaac (1978) study) and then uses an external criterion which compares this structure with the obtained clustering. Rohlf and Fisher (1968) did provide null distribution values for the hierarchical cophenetic index for selected algorithms. However, apparently no study has been conducted in order to determine the measure's usefulness when structure is present. Again, as with the single partition index, several alternative hierarchical measures have been developed. Baker (1974) and
Hubert (1974) used the Goodman and Kruskal (1954) gamma statistic as a rank order variant of the hierarchical cophenetic coefficient. The alternative measures should also be examined for desirable operating characteristics.
V. SUMMARY

The study was designed to evaluate the recovery behavior of a set of clustering algorithms over a series of constructed data sets which were subjected to six types of error perturbation. Eleven agglomerative hierarchical methods were examined and included the complete linkage, single linkage, group average, weighted average, centroid, median, minimum variance, Beta-flexible, average linkage within the new group, minimum total and minimum average within group sum of squares. Four non-hierarchical centroid sorting methods were also examined and included MacQueen's K-means procedure, Forgy's method, Jancey's method, and the convergent K-means algorithm.

Two to five clusters were present in each of the 108 basic error free parent data sets. The data sets were generated in a Euclidean space where point distributions within clusters were generated by a truncated multivariate normal random number generator. The sets were constructed in such a manner that cluster overlap was not possible. Thus, the clusters exhibited the properties of internal cohesion and external isolation.
Error Condition 1 represented the error free parent data sets. The condition provided a test of the clustering algorithms in an ideal error free situation. Condition 2 involved adding a number of outlier points to the basic data sets. Condition 3 involved the error perturbation of the true distances in the parent data sets. Condition 4 represented the addition of one or two random error dimensions to the basic sets. Condition 5 involved the use of non-Euclidean distance measures. Finally, Condition 6 involved standardizing the variables in the basic data sets before computation of the Euclidean distances.

The primary measure of cluster recovery was the Rand statistic. The index is an external criterion measure where the known cluster membership was used as the criterion. A single partition cophenetic correlation coefficient and a hierarchical cophenetic correlation were also computed as internal criterion measures.

Analysis of variance results for the external criterion statistic indicated highly significant effects for methods, conditions, and the method by condition interaction. Error Conditions 5 and 6 were found to have relatively mild effects on cluster recovery whereas the higher error levels in Conditions 2, 3, and 4 caused fairly marked decrements in correct cluster recovery. In general, the non-hierarchical methods were found to give lower recovery rates than the hierarchical methods. Of the four K-means procedures,
Jancey's method appeared to be the best. The method by condition interaction was interpreted as revealing that the rank order performance of the algorithms changed from one error condition to the next. Neuman-Keuls post hoc tests within the error condition levels indicated that membership in the statistically equivalent superior group did change across conditions to some extent. However, at least one method, the group average method, appears to be about the best procedure in all error perturbation conditions except those involving a high concentration of outliers or intermediates.

Finally, a substantial correlation (.84) over all data sets, conditions, and methods was found for the single partition cophenetic correlation coefficient (an internal goodness-of-fit criterion) and the external criterion. The results indicate that the coefficient can be used with some degree of confidence as an indicator of correct cluster recovery. The correlation between the hierarchical cophenetic correlation coefficient and the Rand statistic was .70, thus indicating that this measure is somewhat less satisfactory as an index of correct cluster recovery.
APPENDIX A

Computer Program Listing for Data Generation Process
DATA GENERATOR PROGRAM - METHODS 1-11 (HIERARCHICAL)
PROGRAMMED BY GLENN W. MELLIGAN
DOCTORAL DISSERTATION - JANUARY 1976
OUTPUT OF DATA SETS ARE ON FILE 1 (FD1DDO1)
THE FIRST LINE CONSISTS OF PRELIMINARY INFORMATION AND IS FORMATTED
[10F8.2]
THE REMAINING LINES CONSIST OF A LOWER HALF MATRIX WITH FORMAT
[10F8.2]
ICB SETS ARE CREATED CONSISTING OF 11 MATRICES PER SET

DIMENSION COORD(10), SOUNDS(2,6,8), STDDS(5,5), UNIF(150),
GDIST(N,75,70,8), NORMS(1), NUCP(1), NUCV(1), T(5)
INTEGER SEED, SET, CLUSN, TLDIM, TDIMR, DENST, EPS, POINT,
DENSTY, OUTLJR(2), TEMP2, CLUST(15), TLDIM, TLDMSM, SAVEM(10),
CTYPES(10), T2, DENSITY(25,22), I2, J2, L1, R1, S1, T1, U1, V1, X1, Y1, Z1,
L2, R2, S2, T2, U2, V2, X2, Y2, Z2, E1, F1, G1, H1, I1, J1, K1, L1,
M1, N1, O1, P1, Q1, R1, S1, T1, U1, V1, W1, X1, Y1, Z1, E1, F1, G1,
H1, I1, J1, K1, L1, M1, N1, O1, P1, Q1, R1, S1, T1, U1, V1, W1, X1, Y1, Z1,
EPS(1000), L1, R1, S1, T1, U1, V1, X1, Y1, Z1, A1, B1, C1, D1, E1, F1,
G1, H1, I1, J1, K1, L1, M1, N1, O1, P1, Q1, R1, S1, T1, U1, V1, W1, X1, Y1,
Z1, A1, B1, C1, D1, E1, F1, G1, H1, I1, J1, K1, L1, M1, N1, O1, P1, Q1,
R1, S1, T1, U1, V1, W1, X1, Y1, Z1, A1, B1, C1, D1, E1, F1, G1, H1
CALL LLXAND
SEED=9419577
SET=0
PRINT 92, SEED
92 FORMAT(1X, 10I9)
C REPLICATION AND CONFIGURATION DD=LOOPS
DO 121 II=1,6
121 SX(II)=1.6
DO 1 RPP=1.3
D0 2 CLUSN=1,4
TDMSM=TDIMR/2+2
TDIMR=TDIMR-1
TP1=TP1+1
DO 4 DENSIT=1,3
SET=SET+1
2 CONTINUE
C GENERATE CLUSTER BOUNDARIES
NBOND=TDIMR-1
CALL RANDOM(SEED, UNIF, RECOND)
ICNT=0
DO 5 IDIM=1,TDIMR
5 CONTINUE
C COMPUTE CLUSTER LENGTHS
DO 6 ICNT=1,TLDIM
ICNT=ICNT+1
LENGTH(ICLUS,IDIM)=10.6*36.0*UNIF(ICNT)
CONTINUE
C COMPUTE ORDER OF CLUSTERS, FIRST SORT THE RANDOM NUMBERS
DO 7 J=1,TDIMR
D0 8 I=1,TLDIM
ICNT=ICNT+1
UNIF(I)=UNIF(J)
DO 9 K=1,CLUSI
K1=1
DO 10 K2=1,CLUSI
JF(UNIF(J1,J))=JF(UNIF(K)) GO TO 10
10 TEMP=UNIF(J1)
JF(J1,J)=JF(J)
JF(J1,K)=TEMP
CONTINUE
8 CONTINUE

CONTINUE
CONTINUE
CONTINUE
&BOUND=(TCLUSN-1)*TDIMNR*1
CALL RANDOM(SETN,UNI,BOUND)
ICTN=0

COMPUTE CLUSTER BOUNDARIES FOR THE DIMENSION WHERE OVERLAP IS NOT
PERMITTED, AND ALSO COMPUTE CENTROID AND STD. DEV.
LOWER=0.
DO 12 I=1,TCLUSN
LOWER=LOWER+LENGTH(I,1)
ICTN=ICTN+1
I=CLUSTER(I,1)
LOWER=LOWER/MAX2(1,LOWER)
DO 12 J=1,ICTN
LOWER=LOWER+LENGTH(I,1)
UPPER=LOWER+LENGTH(I,1)
MEAN(I,1)=(UPPER-LOWER)/2.+LOWER
STDEV(I,1)=LENGTH(I,1)/2.
IF(STDEV(I,1).EQ.TCLUSN) GO TO 15
ADU=(STDEV(I,1)*LENGTH(I,1,1)/3.)*(25.*UNI(1,ICTN))
LOWER=UPPER+ADU
ICTN=ICTN+1
12 CONTINUE
SAVE=UPPER*.27
SAVE=UPPER

COMPUTE CLUSTER BOUNDARIES FOR THE DIMENSIONS WHERE OVERLAP IS
PERMITTED, AND COMPUTE CENTROID AND STD. DEV.
DO 11 J=2,TDIMNR
DO 14 I=1,TCLUSN
LOWER=UNI(1,ICTN)*SAVEED
ICTN=ICTN+1
I=CLUSTER(I,1)
LOWER=LOWER+LENGTH(I,1)
UPPER=LOWER+LENGTH(I,1)
MEAN(I,1)=(UPPER-LOWER)/2.+LOWER
STDEV(I,1)=LENGTH(I,1,1)/3.
11 CONTINUE

SPECIFY DENSITY WITHIN CLUSTER AND OUTLIERS
POINT=1
IF(DENSIT.EQ.2) POINT=15
IF(DENSIT.EQ.3) POINT=29
IF(DENSIT.EQ.4) POINT=POINT+2
IF(DENSIT.EQ.5) POINT=POINT+9
DO 15 I=1,TCLUSN
DENSIT(I)=GRIDPO(1,ICTN)
POINT(I)=POINT+1
15 CONTINUE

GENERATE WITHIN CLUSTER DATA POINTS
NUM=0
DO 16 J=1,TCLUSN
ICTN=1
OLDNUM(1,NUM)+1
NUM=NUM(J,1)+NUM
NUMB=1+NUM(J,1)*TDIMNR
CALL NORMAL(SETN,NUMB)
J=J+2
ICTN=ICTN+1
SAVE(N,J)=OLDNUM
SAVE(N,J)=NUM
DO 39 K=0,OLDNUM
39 CONTINUE
DO 17 J=1,TDIMNR
DO 20 I=1,J
DENSIT(I,J)=GRIDPO(I,J)*NORMAL(1,ICTN)*STDEV(I,J,1)
GO TO 17
17 CONTINUE
16 CONTINUE
15 CONTINUE
14 CONTINUE
13 CONTINUE
12 CONTINUE
11 CONTINUE
10 CONTINUE
9 CONTINUE
8 CONTINUE
7 CONTINUE
6 CONTINUE
5 CONTINUE
4 CONTINUE
3 CONTINUE
2 CONTINUE
1 CONTINUE
JGM = 1

GENERATE OUTLIERS

NUM = 50

DO 21 J = 1, TCLUSN
    OLDNUM = NUM + 1
    NUM = NUM(J) + 1
    NUM = NUM(J) + DIMNUM(J) * NUM
    CALL NORMAL(Seed, NORM, NUM)
    DO 22 K = OLDNUM, NUM
        DO 23 J = 1, DIMNUM
            ICONT = ICONT + 1
            COORD([J]) = MEAN([J]) + NORM(ICONT) * STDDEV([J]) * 3.0
            DO 25 L = 1, TCLUSN
                IF (COORD([J,L]) .LT. BOUNDS(1) .OR. (COORD([K,L]) .GT. BOUNDS(2)))
                    CONTINUE
                    ICONT = ICONT - 1
                    CALL NORMAL(Seed, NORM, ICONT, DIMNUM)
                    GO TO 22
            CONTINUE
        CONTINUE
    CONTINUE
    GO TO 95
    JGM = 2
    GO TO 94

COMPUTE DISTANCES FOR CONDITION 1

DO 26 J = 2, 50
    K = 1, K1
    SUM = 0.0
    DO 28 L = 1, DIMNUM
        SUM = SUM + (COORD([J,L]) - COORD(K,L)) * (COORD([J,L]) - COORD(K,L))
    CONTINUE
    DIST([J]) = SQRT(SUM)
    CONTINUE

WRITE CONDITION 1 ON TAPE

NPOINT = 50
TYPE = 1
WRITE(1, 30) SET, TYPE, NPOINT, DIMNUM, TCLUSN, DENSIT, C(SAVEPOS(J), I = 1, IIT)
DO 30 I = 2, 50
    K = 1, K1
    WRITE(1, 31) DIST([I,K], K = 1, K1)

COMPUTE DISTANCES FOR 50% OUTLIER CONDITION

DO 32 J = 51, 60
    K = 1, K1
    SUM = 0.0
    DO 34 L = 1, DIMNUM
        SUM = SUM + (COORD([J,L]) - COORD(K,L)) * (COORD([J,L]) - COORD(K,L))
    CONTINUE
    DIST([J]) = SQRT(SUM)
    CONTINUE

WRITE CONDITION 2A ON TAPE

NPOINT = 60
TYPE = 2
WRITE(1, 30) SET, TYPE, NPOINT, DIMNUM, TCLUSN, DENSIT, C(SAVEPOS(J), I = 1, IIT)
DO 30 I = 2, 60
    K = 1, K1
    WRITE(1, 31) DIST([I,K], K = 1, K1)

COMPUTE DISTANCES FOR 40% OUTLIERS

DO 35 J = 61, 70
    K = 1, K1
    SUM = 0.0
    I = 1, 10

GO TO 96
 KP=K IF(K.GT.50) KP=K+10 DO 37 L=1,TDIMMR 37 SUM=SUM+((COORD(I,P,L)-COORD(KP,L))*(COORD(I,P,L)-COORD(KP,L))) DIST(I,K)=SQRT(SUM) CONTINUE 38 CONTINUE WRITE(29) ON TAPE 39 NPQNT=70 TYPE=3 WRITE(1,29) SET,TYPE,NPQNT,TDIMMR,TDIMNR,CLUSRK,TCLSUN,DENSIT, CGAVENKE(1,1:1,1:1) DO 38 I=2,70 K=1-1 WRITE(1,31) (DIST(I,K),K=1,K1) 38 COMPUTE DISTANCE MATRIX FOR LOW/HIGH ERROR SETS 39 IREP=1 GAMMA=1.0 TYPE=4 NPQNT=50 ICONT=0 40 NUMBER=1225*TOIMNR CALL NORMAL(SEED,NORML,NUMBER) DO 41 I=1,50 K=1-1 DO 42 K=1,K1 SUM=0.0 DO 42 L=1,TDIMMR ICONT=ICONT+1 SUM=SUM+(COORD(I,L)+COORD(K,L)+GAMMA*NORMLI(1,1))**2 DIST(I,K)=SQRT(SUM) 42 CONTINUE CONTINUE WRITE(29) ON TAPE 43 WRITE(1,29) SET,TYPE,NPQNT,TDIMMR,TDIMNR,CLUSRK,TCLSUN,DENSIT, CGAVENKE(1,1:1,1:1) DO 44 I=2,50 K=1-1 WRITE(1,31) (DIST(I,K),K=1,K1) 44 COMPUTE CORRELATION OF CONDITION 3A/R TO 1 45 SUM1=0.0 SUM2=0.0 SUM3=0.0 SUM4=0.0 SUM5=0.0 SUM6=0.0 DO 45 I=2,50 K=1-1 DO 46 K=1,K1 SUM1=SUM1+DIST(I,K) SUM2=SUM2+DIST(I,K)*DIST(I,K) SUM3=SUM3+DIST(I,K)**2 SUM4=SUM4+DIST(I,K) SUM5=SUM5+DIST(I,K)**2 SUM6=SUM6+DIST(I,K) 46 CONTINUE TOP=SUM2-SUM1/1225. BOTTOM=SUM2-SUM1/1225. B=SUM2-SUM1/1225. COR=IREP*TOP/BOTTOM LOOP BACK TO COMPUTE HIGH ERROR SET 47 IF(IREP.GT.2) GO TO 47 IREP=2 GAMMA=2. TYPE=5 ICONT=0 GO TO 40 COMPUTE DISTANCE MATRIX FOR 1 DIMENSION, RANDOM ERROR 41 FIRST GENERATE RANDOM NOISE DIMENSIONS 47 CALL RANDOM(SEED,UNI,150)
DO 48 I=1,50
   COORD(I,TP)=UNI(I)*SAVE
DO 49 J=2,50
   K=1,K1
   DO 50 K1=1,K1
      SUM=0.
      DO 51 K1=1,K1
         SUM=SUM+(COORD(I,L)-COORD(K,L))**2
      END
   END
   DIST(I,J)=SQRT(SUM)
CONTINUE
49 CONTINUE
WRITE (44,AA) ON TAPE
   TYPE=4,
   WRITE(1,29) SET,TYPE,NPOINT,DIMNR,TDIMNR,CLUSNR,TCLUSN,DENSIT,
   C(SAVENV(I),I=1,14T)
DO 52 J=2,50
   K=1
   WRITE(1,31) (DIST(I,K),K=1,K1)
50 COMPUTE CORRELATION OF CONDITION AA TO 1
   SUM1=0.0
   SUM2=0.0
   SUM3=0.0
   DO 53 I=2,50
      K=1,K1
      SUM1=SUM1+DIST(I,K)
   END
   SUM2=SUM2+DIST(I,K)*DIST(I,K)
   SUM3=SUM3+DIST(K,K)*DIST(I,K)
   CONTINUE
   TOP=SUM2-SUM1**2/SUM3/1225.
   BOTTOM=SQRT((SUM2-SUM1**2/SUM3/1225.))**2
   CORR(I,J)=TOP/BOTTOM
55 COMPUTE DISTANCE MATRIX FOR TWO DIMENSIONAL RANDOM ERROR
   DO 55 I=1,51,100
      IP50=IP50+1
      COORD(IP50,TP)=UNI(I)*SAVE
   END
   DO 56 J=2,50
      K=1,K1
      SUM0=0.0
      DO 58 K=1,K1
         SUM0=SUM0+COORD(I,K)-COORD(K,L)
      END
      DIST(I,K)=SQRT(SUM0)
      CONTINUE
56 CONTINUE
WRITE (44,AA) ON TAPE
   TYPE=7,
   WRITE(1,29) SET,TYPE,NPOINT,DIMNR,TDIMNR,CLUSNR,TCLUSN,DENSIT,
   C(SAVENV(I),I=1,14T)
DO 59 J=2,50
   K=1
   WRITE(1,31) (DIST(I,K),K=1,K1)
58 COMPUTE CORRELATION OF CONDITION 48 TO 1
   SUM1=0.0
   SUM2=0.0
   SUM3=0.0
   DO 60 I=2,50
      K=1,K1
      SUM1=SUM1+DIST(I,K)
   END
   SUM2=SUM2+DIST(I,K)*DIST(I,K)
   SUM3=SUM3+DIST(K,K)*DIST(I,K)
   CONTINUE
   TOP=SUM3-SUM1**2/SUM3/1225.
   BOTTOM=SQRT((SUM3-SUM1**2/SUM3/1225.))**2
   CORR(I,J)=TOP/BOTTOM
60 COMPUTE CONDITION 56 DISTANCE MATRIX
   DO 76 J=2,50
K=I-1
DO 77 K=1,K1
J=1
S1=0.0
S2=0.0
S3=0.0
S4=0.0
DO 78 J=1,TDIMNR
S1=S1+COORD(I,J)
S2=S2+COORD(I,J)*COORD(I,J)
S3=S3+COORD(I,J)*COORD(K,J)
S4=S4+COORD(I,J)*COORD(K,J)
TOP=S5=S3/FLOAT(TDIMNR)
BOTTOM=SRT((S2-S1)*S1/ELOAT(TDIMNR))*(S4-S3*S3/FLOAT(TDIMNR))
77 CONTINUE
WRITE CONDITION 5B ON TAPE
WRITE(1,29) SET,TYPE,NPOINT,TDIMNR,CLUSNR,TCLUDN,DENSIT,
C(SAVEND(I,1,1,1)),I=1,ITIT
DO 79 I=2,50
K=I-1
WRITE(1,31) (DIST(I,K),K=1,K1)
Compute correlation of condition 1 TO 1
SUM1=0.0
SUM2=0.0
SUM5=0.0
DO 80 I=2,50
K=I-1
DO 81 K=1,K1
SUM1=SUM1+DIST(I,K)
SUM2=SUM2+DIST(I,K)*DIST(I,K)
SUM5=SUM5+DIST(I,K)*DIST(K,K)
80 CONTINUE
TOP=SUM5-SUM1*SUM1/SUM2
BOTTOM=SRT((SUM2-SUM1)*SUM1/125.)
CORR(6)=TOP/BOTTOM
STANDARDIZE THE COORDINATE VALUES
IDIM=1
SUM1=0.0
SUM2=0.0
DO 62 I=1,50
SUM1=SUM1+COORD(I,IDIM)
SUM2=SUM2+COORD(I,IDIM)*COORD(I,IDIM)
CMEAN=SUM1/50.
SUM1=(SUM1*SUM1)/50.
STD=SRT((SUM2-SUM1)/50.)
63 DO 65 I=1,50
COORD(I,IDIM)=(COORD(I,IDIM)-CMEAN)/STD
IF (IDIM.LE.TDIMNR) GO TO 64
COMPUTE DISTANCE MATRIX FOR CONDITION 6
ICNT=0
DO 67 I=2,50
K=I-1
DO 68 K=1,K1
SUM=0.0
DO 67 L=1,TDIMNR
SUM=SUM+(COORD(I,L)-COORD(K,L))*(COORD(I,L)-COORD(K,L))
DIST(I,K)=SRT(SUM)
ICNT=ICNT+1
65 CONTINUE
WRITE CONDITION 6 ON TAPE
WRITE(1,29) SET,TYPE,NPOINT,TDIMNR,CLUSNR,TCLUDN,DENSIT,
C(SAVEND(I,1,1,1)),I=1,ITIT
DO 68 I=2,50
K=I-1
WRITE(1,31) (DIST(I,K),K=1,K1)
COMPUTE CORRELATION OF CONDITION 6 TO 1

SUM1=0.0
SUM2=0.0
SUM5=0.0
DO 69 I=2,50
   K1=1
   DO 70 K=1,K1
   SUM1=SUM1+DIST(I,K)
   SUM2=SUM2+DIST(I,K)*DIST(I,K)
   SUM5=SUM5+DIST(I,K)*DIST(K,1)
   CONTINUE
   TOP=(SUM1-SUM2/1225.)*SUM5/1225.
   BOTTOM=SUM5*(SUM2-SUM1*SUM1/1225.)*K1
   CORR(TOP/BOTTOM)

COMPUTE CATTELL INDEX DISTANCE MATRIX

CHI2MED=9.357
CHI2DIN=9.348
CHI2DIN=7.344
DO 71 I=2,50
   K1=1
   DO 72 K=1,K1
   ICNT=ICNT+1
   DIST(ICNT)=(TOP/(ICNT-1.0))*(1-(CHI2MED-CHI2DIN)/(CHI2MED+CHI2DIN)-1.0)
   CONTINUE

WRITE CONDITION 5A ON TAPE

TYPE=10
WRITE(1,29) SET, TYPE, NPOINT, DIMNR, TOIMNR, CLUSNR, TCLUSN, DENSIT, C1, C2
   ICNT=ICNT+1
   K1=1
   WRITE(1,31) (DIST(I,K), K=1,K1)

COMPUTE CORRELATION OF CONDITION 5A TO 1

SUM1=0.0
SUM2=0.0
SUM5=0.0
DO 73 J=2,50
   K1=1
   DO 75 K=1,K1
   SUM1=SUM1+DIST(I,K)
   SUM2=SUM2+DIST(I,K)*DIST(I,K)
   SUM5=SUM5+DIST(I,K)*DIST(K,1)
   CONTINUE
   TOP=(SUM1-SUM2/1225.)*SUM5/1225.
   BOTTOM=SUM5*(SUM2-SUM1*SUM1/1225)*K1
   CORR(TOP/BOTTOM)

COMPUTE RANDOM NOISE DATA AND DISTANCES

NUMBR=50*TOIMNR
CALL RANDOM (SEED, NORM, NUMBR)
   ICNT=ICNT+1
   K1=1
   WRITE(1,31) (DIST(I,K), K=1,K1)

COMPUTE RANDOM NOISE DATA ON TAPE

WRITE(1,29) SET, TYPE, NPOINT, DIMNR, TOIMNR, CLUSNR, TCLUSN, DENSIT, C1, C2
   ICNT=ICNT+1
   K1=1
   WRITE(1,31) (DIST(I,K), K=1,K1)
COMPUTE CORRELATIONS OF RANDOM NOISE DATA TO CONDITION

DO 68, I=2, 50
K1=1
DO 89 K=1, K1
SUM1=SUM1+DIST(I, K)
SUM2=SUM2+DIST(I, K)*DIST(I, K)
SUM5=SUM5+DIST(K, I)*DIST(K, I)
CONTINUE
89 SUM=SUM1+SUM2+SUM5
88 CONTINUE
TOP=SQRT((SUM2-SUM)*SUM3/1225.)
END

PRINT AND PUNCH CORRELATION INFORMATION

DO 122 I=1, 8
SXI(I)=SX(I)+C(600(I))
PRINT 90, SET, (CORRI(I), I=1, 8)
CONTINUE
122 PRINT 90, SET, (CORRI(I), I=1, 8)
CONTINUE
4 CONTINUE
CONTINUE
3 CONTINUE

PRINT MEAN CORRELATION VALUES

DO 120 I=1, 8
MEAN(I)=SX(I)/N
PRINT 191, MEAN(I), I=1, 8
120 FORMAT(I8)
31 FORMAT(4F10.3)
90 FORMAT(1X, 12, 8F10.3)
191 FORMAT(4F10.3)
STOP
END
DATA GENERATOR PROGRAM - METHODS 12-15 (NON-HIERARCHICAL)

This version of the generator reads in the coordinate values for each condition rather than a linear half triangular matrix.

100 sets are created consisting of 11 matrices per set

DIMENSION UNCOORD(90,13), SOURCES(5,5,5), SIDESV(5,5), UNI(100, 1),
       CERGTE(90,5), LIST(90,10), NDIM1(5)
       INTEGER SEED=1, TCLUS=1, CLUSYN#1, DIMARK, R=SIGN(1,1),
       CY=1, TP1=2, TP2=3, NDIM
       REAL MEANS(90,5), LENGTH(5,10), LOWR, NORM(90,5)
       DATA DENITY/52.4714,17.87,12.41,11.79,10.98,10.59,5.97,7.22,
             9.35,9.19,1.87,1.61,1.06,1.02,0.86,0.69,0.53,0.39,
             0.16,0.16,1.08,3.53,3.98,3.53,3.37,
             CALL LKLAND
             SEED=149277
             SET=0

REPLICATION AND CONFIGURATION DC-LOCPS

DO 1 REPS=1,3
   TLUSN=1, CLUSYN#1
   ITTL=2, TLUSN
   1 DO 4 DIMARK=1,3
      TDIMK=1, TDIMK
      4 TP1=TDIMK
      TP2=TP1+1
      5 DO 6 DENIT=1,5
      SET=SET+1

GENERATE CLUSTER BOUNDARIES

NDIMK=TDIMK+1
CALL RANDOM(SEED, UNI, NDIMK)
ICN=0
DO 6 DIMK=1, TDIMK

COMPUTE CLUSTER LENGTHS

DO 6 TLUSN=1, TLUSN
   ICN=ICN+1
   LENGTH(TLUSN, DIMK)=DIMK+3.0
      END(TLUS)
   CONTINUE

COMPUTE ORDER OF CLUSTERS, FIRST SORT THE RANDOM NUMBERS

DO 10 ICN=1, TDIMK
   ICN=ICN+1
   K=1, TCLUS
   10 DO 15 K=1, TCLUS
      UNIT=UNIT(K)
      UNIT=UNIT(K)
      TEMP=UNIT(K)
      UNIT=UNIT(K)
      K=K+1
      15 CONTINUE
   CONTINUE
   TDIMK=1
   UNIT=UNIT(K)
   UNIT=UNIT(K)
   CONTINUE

CALL RANDOM(SEED, UNI, NDIMK)
ICNT=1

COMPUTE CLUSTER BOUNDARIES FOR THE DIMENSION WHERE OVERLAP IS NOT PERMITTED, AND ALSO COMPUTE CENTROID AND STD. DEV.

LOWER=0.
DO 12 I=1,TCLUST
UPPER=LOWER+LENGTH(I,1)
IF(LOWER+LENGTH(I,1))=UPPER
BOUND(1:1,1:1)=LOWER
STDEV(1:1)=LENGTH(I,1)/3.
IF(LOWER+LENGTH(I,1))>UPPER
CONTINUE
12
SAVE=UPPER
CONTINUE
SAVE=UPPER

COMPUTE CLUSTER BOUNDARIES FOR THE DIMENSIONS WHERE OVERLAP IS PERMITTED, AND COMPUTE CENTROID AND STD. DEV.

DO 11 J=1,TCLUST
DO 10 K=1,ICNT
LOWER=UPPER=LENGTH(K,1)
IF(LOWER+LENGTH(K,1))=UPPER
BOUND(1:1,1:1)=LOWER
STDEV(1:1)=LENGTH(K,1)/3.
CONTINUE
11
SAVE=UPPER
CONTINUE
SAVE=UPPER

SPECIFY DENSITY WITHIN CLUSTER AND OUTLIER:

POINT=1
IF(DENSIT(EQ,2)) POINT=0
IF(DENSIT(EQ,3)) POINT=2
IF(TCLUST(EQ,2)) POINT=0
IF(TCLUST(EQ,3)) POINT=0
IF(TCLUST(EQ,4)) POINT=0
DO 15 J=1,TCLUST
IF(DENSIT(EQ,POINT))
NUM(J)=DENSIT(J,POINT)
CONTINUE
15
NUM(J)=DENSIT(J,POINT)
CONTINUE

GENERATE WITHIN CLUSTER DATA POINTS

NUM=0
DO 16 J=1,TCLUST
ICNT=1
OLDNUM=NUM
NUM=J+1
NUMR=J+1
CALL NORMAL(SEED,NOR4L+NUMR)
JJK=2
JJK=0
SAVE=NOR4L
SAVE=NOR4L+NUM
DO 39 K=OLDNUM,NUM
LIST(K)=J
DO 17 J=1,TDIM40
DO 18 K=OLDNUM,NUM
NORMAL(4,J,1)=NORMAL(4,ICNT)*STDEV(J,1)
IF(NORMAL(4,J,1)<=BOUND(1,J,1)) GO TO 19
IF(NORMAL(4,J,1)>=BOUND(2,J,1)) GO TO 19
ICNT=1
GO TO 16
19
CALL NORMAL(SEED,NOR4L+ICNT+1)
CONTINUE
18
CONTINUE
39
CONTINUE
16
CONTINUE

GENERATE OUTLIERS

NUM=50
ICNT=1
DO 21 J=1,TCLUS
NUMOUT=NUM
NUMOUT(J)=JG/NUM
NUMOUT=NUMOUT(J)*DIMR pains)
CALL NORMALISED(NRML,NUM
) DO 22 K=1,NUM
DO 21 22 CONTINUE
ICNT=ICNT+1
23 DO 24 J=1,TCLUS
DO 24 25 CONTINUE
THETA(J)=1
ICNT=ICNT+1
25 CONTINUE
ICNT=ICNT+1
CALL NORMALISED(NRML,INST
) ICNT=ICNT+1
CONTINUE
CONTINUE
CONTINUE
CONTINUE
IF(IOM. EQ.2) GO TO 95
INUM=2
GO TO 94
C WRITE CONDITION 1 ON TAPE
55 NPPOINT=50
WRITE(1,14) SI;I;TYPENPQN,POINT,TDO,CLUSN,TCLUSN,DENSIT
C(ISAVENS(1:1)),J
DO 30 I=1,50
WRITE(1,31) (CLUHR(I,K),K=1, TDIMR)
30 CONTINUE
C WRITE CONDITION 2 ON TAPE
C NPPOINT=10
WRITE(1,29) TYPENPQN,POINT,TDO,CLUSN,TCLUSN,DENSIT
C(ISAVENS(1:1)),J
DO 39 I=1,50
WRITE(1,31) (CLUHR(I,K),K=1, TDIMR)
39 CONTINUE
C WRITE CONDITION 3 ON TAPE
NPPOINT=70
WRITE(1,14) SI;I;TYPENPQN,POINT,TDO,CLUSN,TCLUSN,DENSIT
C(ISAVENS(1:1)),J
DO 38 I=1,50
WRITE(1,31) (CLUHR(I,K),K=1, TDIMR)
38 CONTINUE
C COMPUTE COORDINATE MATRIX FOR LUMINACE ERROR
HRP=1
GAMMA=1.0
NPPOINT=50
ICNT=0
40 NUMN=125*TDIMR
CALL NORMALISED(SF, NORML,NUMN)
DO 41 I=1,50
NUMN=K*TDIMR
ICNT=ICNT+1
41 NUMN=125*TDIMR
CALL NORMALISED(SF, NORML,NUMN)
DO 44 I=1,50
WRITE(1,31) (ERROR(I,K),K=1, TDIMR)
44 CONTINUE
C WRITE CONDITION 3A/E ON TAPE
WRITE(1,14) SI;I;TYPENPQN,POINT,TDO,CLUSN,TCLUSN,DENSIT
C(ISAVENS(1:1)),J
DO 45 I=1,50
WRITE(1,31) (ERROR(I,K),K=1, TDIMR)
45 CONTINUE
C LOOP BACK TO COMPUTE YNIE SET
IF(IOM. EQ.2) GO TO 47
REP=2
GAMMA=2.
TYPE=6
INRT=5
GO TO 40
C GENERATE DATA FOR 1 DIMENSION RANDOM ERROR
40 CALL RANM(D600,UM1150)
DO 40 I=1,50
CWRITE CONDITION 4A ON TAPE
41 TYPE=6
WRITE(1,129) SET,TYPE,NPOINT,ODIMN,TDIMN,CLUSNR,TCLUSN,DENSIT,
C(SAVENB1),I=1,ITIT
DO 52 I=1,50
CWRITE (1,131) (CORD1(I,K),K=1,TP1)
52 C GENERATE DATA FOR 2 DIMENSION RANDOM ERROR
C DO 56 I=1,50
CWRITE CONDITION 4B ON TAPE
53 TYPE=6
WRITE(1,129) SET,TYPE,NPOINT,ODIMN,TDIMN,CLUSNR,TCLUSN,DENSIT,
C(SAVENB1),I=1,ITIT
DO 56 I=1,50
CWRITE (1,131) (CORD1(I,K),K=1,TP1)
56 CWRITE CONDITION 5B ON TAPE
57 TYPE=6
WRITE(1,129) SET,TYPE,NPOINT,ODIMN,TDIMN,CLUSNR,TCLUSN,DENSIT,
C(SAVENB1),I=1,ITIT
DO 70 I=1,50
CWRITE (1,131) (CORD1(I,K),K=1,TDIMN)
70 CSTANDARDIZE THE COORDINATE VALUES
C IDIMN=1
SUM=0.0
SUM2=0.0
SUM1=0.0
SUM=CORD1(I,1)+SUM
SUM2=CORD1(I,1)+SUM2
SUM1=SUM1+SUM2
MEAN=(SUM2-SUM1)/50.
STD=(50*Y(SUM2-SUM1)/50.)
DU 53 I=1,50
CORD1(I,1)=CORD1(I,1)-MEAN/STD
IDIMN=IDIMN+1
IF IDIMN.LE.TDIMN GO TO 64
CWRITE CONDITION 6 ON TAPE
C TYPE=6
WRITE(1,129) SET,TYPE,NPOINT,ODIMN,TDIMN,CLUSNR,TCLUSN,DENSIT,
C(SAVENB1),I=1,ITIT
DO 60 I=1,50
CWRITE (1,131) (CORD1(I,K),K=1,TDIMN)
60 CWRITE CONDITION 5A ON TAPE
C TYPE=10
WRITE(1,129) SET,TYPE,NPOINT,ODIMN,TDIMN,CLUSNR,TCLUSN,DENSIT,
C(SAVENB1),I=1,ITIT
DO 73 I=1,50
CWRITE (1,131) (CORD1(I,K),K=1,TDIMN)
73 CCOMPUTE RANDOM NOISE DATA
C NUMR=50*TDIMN
CALL RANM(S=0,NDM1,NUMR)
CNUMT=1
DO 82 I=1,50
CNUMT=NUMT+1
82
COORD(I,J)=NORMAL(I,CT)*SAVE
CONTINUE
WRITE RANDOM NOISE DATA ON TAPE
    TYPE=11
    WRITE(1,29) SET, TYPE, GOPINT, DIMAR, TOI, ANH, CLUSNR, TCLUSNR, DEN, CT
    DO 67 J=1,35
67    WRITE(1,31) (COORD(I,K), K=1, TOI)
CONTINUE
CONTINUE
CONTINUE
CONTINUE
CONTINUE
CONTINUE
CONTINUE
CONTINUE
CONTINUE
STOP
END
NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR PACKAGE LLRANDOM

PURPOSE

GENERATES RANDOM NUMBERS VIA EIGHT ROUTINES:

\texttt{INTG} GENERATES INTEGER RANDOM NUMBERS
\texttt{RANDOM} GENERATES SINGLE PRECISION FLOATING POINT RANDOM NUMBERS ON \(\{0,1\}\)
\texttt{NORMAL} GENERATES SINGLE PRECISION FLOATING POINT NORMAL DEVIATES (MEAN 0, VARIANCE 1)
\texttt{EXPON} GENERATES SINGLE PRECISION FLOATING POINT EXPONENTIAL DEVIATES (MEAN 1)
\texttt{SINT}, \texttt{GRAND}, \texttt{SNORM}, \texttt{SEXPN}... AS ABOVE BUT WITH SHUFFLING OR THE RANDOM NUMBERS OR DEVIATES FOR (HOPEFULLY) BETTER RANDOMNESS PROPERTIES

USAGE

CALL LLRAND ONCE AND ONLY ONCE, AT THE BEGINNING OF YOUR FORTRAN MAIN PROGRAM

THEN CALL \texttt{INTG}(IX,N)

OR CALL \texttt{GRAND}(IX,A,N)

OR CALL \texttt{NORMAL}(IX,A,N)

OR CALL \texttt{EXPON}(IX,A,N)

OR CALL \texttt{SINT}(IX,A,N)

OR CALL \texttt{GRAND}(IX,A,N)

OR CALL \texttt{SNORM}(IX,A,N)

OR CALL \texttt{SEXPN}(IX,A,N)

DESCRIPTION OF PARAMETERS

\texttt{IX} IS THE STARTING VALUE OF THE SEQUENCE AND MAY CONTAIN ANY INTEGER NUMBER BETWEEN \(1\) AND \(2^{32} - 1\). (THIS VARIABLE SHOULD NOT BE ALTERED BY THE USER DURING THE EXECUTION OF HIS PROGRAM, UNLESS HE DESIRES TO REPEAT A SEQUENCE OF RANDOM NUMBERS OR DEVIATES.)

\texttt{N} \textbf{IS} EITHER A SCALAR OR VECTOR VARIABLE WITH A SPECIFIED DIMENSION INTO WHICH THE RANDOM NUMBER(S) OR DEVIATE(S) ARE TO BE STORED. FOR \texttt{INTG} AND \texttt{SINT} THIS ARGUMENT SHOULD BE OF \texttt{INTEGER} TYPE.

\texttt{A} IS AN INTEGER VARIABLE (OR CONSTANT) DESIGNATING HOW MANY RANDOM NUMBERS OR DEVIATES ARE TO BE GENERATED DURING THIS CALL. IF \texttt{A} IS A SCALAR, IT MUST BE A VECTOR OR DIMENSION \(=\texttt{N}\). IF \texttt{N} \# 1, THEN \texttt{A} MAY BE A SCALAR.

REMARKS

THIS PACKAGE IS SPECIFIC TO IBM SYSTEM 360/370 COMPUTERS.

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

THIS IS A COMPLETE PACKAGE.

EXAMPLE OUTPUT FROM RANDOM

THE FIRST 50 NUMBERS GENERATED FROM SEED IX=269571 ARE:

\[
\begin{array}{cccccccc}
0.265203 & 0.5288504 & 0.3367534 & 0.7775312 & 0.2417047 \\
0.216043 & 0.2706619 & 0.6349393 & 0.5395365 & 0.956704 \\
0.0000808 & 0.9187261 & 0.4462852 & 0.860176 & 0.9301214 \\
0.8144897 & 0.2950247 & 0.7171293 & 0.0366447 & 0.7356305 \\
0.2511211 & 0.556487 & 0.272149 & 0.3928515 \\
0.2930548 & 0.3868318 & 0.9768311 & 0.3497837 & 0.4641740 \\
0.3938826 & 0.876858 & 0.7758283 & 0.3464888 & 0.9674013 \\
0.113967 & 0.782747 & 0.5408276 & 0.867664 & 0.6462207 \\
0.4971402 & 0.0089293 & 0.6494936 & 0.0024971 & 0.8452160 \\
0.7238113 & 0.0969594 & 0.5279287 & 0.8977858 & 0.0871457 \\
\end{array}
\]

METHOD

SEE THE FOLLOWING REFERENCES:


REXP TOOTH FUNCTION
FUNCTION REXP(IX,IX)
DIMENSION (L65)
DATA C /Z46F000, Z46E1000, Z46D4000, Z46C7000, Z46B9000,
      REXP C /Z46F000, Z46E1000, Z46D4000, Z46C7000, Z46B9000,
      C /Z46F000, Z46E1000, Z46D4000, Z46C7000, Z46B9000,
      C /Z46F000, Z46E1000, Z46D4000, Z46C7000, Z46B9000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
      C /Z46D4000, Z46D4000, Z46D4000, Z46D4000, Z46D4000,
DC  AL1(6)
DC  CL4(128)
STM  R14,R12-12(R13)  SAVE REGISTERS IN HIGH SAVE AREA
EM  R12>R15  ESTABLISH BASE ADDRESS
ST  R13,Sappointed  SAVE CALLER'S R 13
LR  R2,R13  NEW SAVE AREA
LA  R13,(R1,R2)  STORE WITH CALLING ROUTINE

** ISSUE SPEI TO GET FIXED POINT OVERFLOWS AS WELL AS FORTRAN **

** INTERRUPTS **

SPIE  FIXIT((#9,12,13,15)
ST  R1,R12-12(R13)  SAVE FORTRAN'S PICA ADDRESS
L  R13,Sappointed  RESTORE CALLER'S R13
EM  R14,R12-12(R13)  RESTORE THE REGISTERS
BEC  10,R16  RETURN
MISSING CARD

** SPIE BRINGS US HERE ON INTERRUPTS **

FORT R1 = PICA
LD  R13,R12-12(R13)  ALL FIXED, CONTINUE
R15,R15,R16  EXTENDED ERROR HANDLING ROUTINE
R15  TAKE CARE OF IT
MISSING CARD

** ENTRY POINT : INT **

CNDP  0,F  BASE REGISTER
USING  INT,R15  BRANCH AROUND ID
DC  AL1(3)
DC  CL3,INT
STM  R14,R12-12(R13)  SAVE REGISTERS IN HIGH SAVE AREA
ST  R13,Sappointed  ADDRESS OF HIGH SAVE AREA IN LOW SAVE AR.
LR  R2,R13  COPY TO R2
LA  R13,Sappointed  ADDRESS OF LOW SAVE AREA
ST  R13,(R1,R2)  ADDRESS OF LOW SAVE AREA IN HIGH SAVE AREA
R6,475  LOAD MULTIPLIER
LA  R2,R2  CONSTANT FOR X
LM  R5,(0,R1)  ADDRESSES OF THREE ARGUMENTS
L  R5,R0,R5  LOAD STARTING VALUE INTO R6
L  R0,(0,R7)  NUMBER OF CONSECUTIVE WORDS TO FILL
SRA  R5,2  CONVERT TO LONG
LR  RT,R2  INITIAL VALUE FOR INDEX REGISTER
CNDP  0,A  ALIGN RIXE LOOP FOR SPEED
MN  R4,99  CONI PRODUCT OF A AND XIN+1
ALDA  R4,1  R4 REMAINDER 1 RS = QUOTIENT
SRL  R4  ADD QUOTIENT TO REMAINDER THEPEVX
AR  R4,R8  SIMULATING DIVISION BY 2**31-1
LR  RT,XIN+1  PUT XIN INTO R4 FOR NEXT GO AROUND
ST  R5,(7,R6)  STORE IN CALLER'S ARRAY
N1  XLE  LOOP AROUND AGAIN
LR  R4,(R1)  GET STARTING VALUE ADDRESS AGAIN
ST  R9,(R4)  STORE AS STARTING VALUE FOR NEXT CALL
LM  R13,Sappointed  SAVE AREA POINTER
R15,R14-12(R13)  RESTORE THE REGISTERS
BEC  15,R16  RETURN
MISSING CARD

** ENTRY POINT : SINT **

CNDP  0,6  BASE REGISTER
USING  SINT,R15  BRANCH AROUND ID
DC  CL4,SINT
STM  R14,R12-12(R13)  SAVE REGISTERS IN HIGH SAVE AREA
ST  R13,Sappointed  ADDRESS OF HIGH SAVE AREA IN LOW SAVE AR.
LR  R2,R13  COPY TO R2
LA  R13,Sappointed  ADDRESS OF LOW SAVE AREA
DC AL 1113
DC CLS, 54D
STM R14, R12, R12(R13) SAVE REGISTERS IN HIGH SAVE AREA
ST R14, R12, R12(R13) ADDRESS OF HIGH SAVE AREA IN LOW SAVE AREA
LR R2, R2 COPY TO R2
LA R13, (5A) ADDRESS OF LOW SAVE AREA
ST R13, R12, R12(R2) ADDRESS OF LOW SAVE AREA IN HIGH SAVE AREA
LM R9, R11, 475 LOAD MULTIPLIER AND NORMALIZATION CONSTANT
LC R2, = CONSTANT FOR IXLE
L R9, R9, (R51) LOAD STARTING VALUE INTO R5
L R3, D, (R7) NUMBER OF CONSECUTIVE WORDS TO FILL
SLA R3, = CONVERT TO BYTES
SR R6, R2 BACKUP ONE WORD IN CALLER'S ARRAY
LR R7, R2 INITIAL VALUE FOR INDEX REGISTER
SUR PRO, R40 CLEAR FLOATING POINT REGISTER 0
LA R12, N0 ADDRESS OF IXLE INSTRUCTION
LA R13, M0 ADDRESS OF NORMALIZATION ROUTINE
LA R9, TABLE ADDRESS OF SHuffling TABLE
L R1, MASK INDEX MASK FOR SHuffling
CNOP 0, 8 ALIGN IXLE LOOP FOR SPEED
NR R4, 1 R4 = REMAINDER ; R5 = QUOTIENT
SLA R4, = ADD QUOTIENT TO REMAINDER THEN FILL
AR R4, 0 SIMULATING DIVISION BY 2**81-1
LR R5, R4 PUT R11 INTO R5 FOR NEXT GO AROUND
NR R4, 51 EXTRACT RIGHT MOST 5 BITS
SLA R4, = CONVERT TO BYTE OFFSET IN TABLE
ST R5, R14(R4, R4) SELECT RANDOM TABLE VALUE
ST R5, R14(R4, R4) REPLACE TABLE VALUE WITH (N)
SR R2, 0 MAKE ROOM FOR THE EXPONENT
CR R0, 10 ON THE EXPONENT
ST R0, G(R7, R6) STORE IN CALLER'S ARRAY
CR R0, 10 DID IT NEED NORMALIZATION?
L R1, 2, 2 YES, GO NORMALIZE IT
C R1, 0, 0 EXIT
BLX R7, R4, 4 LOOP AROUND AGAIN
L R1, 2, 13 GET ARGUMENT LIST POINTER AGAIN
AR R4, D(R1) GET STARTING VALUE ADDRESS AGAIN
ST R0, 0, D(R1) STORE AS STARTING VALUE FOR NEXT CALL
LM R1, R12, R12(R13) RESTORE THE REGISTERS
L R1, 4, 2 RETURN
LE FRX, G(R7, R6) LOAD INTO FLOATING POINT REGISTER 2
AP R3, Z, 0 ADD ZERO AND NORMALIZE
STEP R2, 0, (R7, R6) STORE BACK NORMALIZED
BR R12 CONTINUE THE IXLE LOOP

ENTRY POINT : NORMAL

CNOP 0, 8 BASE REGISTER
NORMAL 12(R15) BRANCH AROUND ID
DC AL 1113
DC CLS, 54D
STM R14, R12, R12(R13) SAVE REGISTERS IN HIGH SAVE AREA
ST R14, R12, R12(R13) ADDRESS OF HIGH SAVE AREA IN LOW SAVE AREA
LR R2, R2 COPY TO R2
LA R13, 5A ADDRESS OF LOW SAVE AREA
ST R13, R12, R12(R2) ADDRESS OF LOW SAVE AREA IN HIGH SAVE AREA
LM R9, R11, 475 LOAD MULTIPLIER AND EXPONENT, AND TEST MASK
LA R12, N0 ADDRESS FOR IXLE
LM R5, R7, (R1) ADDRESSES OF THREE ARGUMENTS
L R9, 0, D(R1) LOAD STARTING VALUE INTO R9
LM R3, R9, (R51) NUMBER OF CONSECUTIVE WORDS TO FILL
L R9, 0, R9 CONVERT TO BYTES
SR R6, R2 BACKUP ONE WORD IN CALLER'S ARRAY
LR R7, R2 INITIAL VALUE FOR INDEX REGISTER
LA R13, TABLE ADDRESS OF TABLE OF CONSTANTS
LA R12, N5 ADDRESS OF IXLE
CNOP 0, 8 ALIGN IXLE LOOP FOR SPEED
MR R4, R9 R4 = REMAINDER ; R5 = QUOTIENT
SRL R5, 1 ADD QUOTIENT TO REMAINDER THEN FILL
AR R4, R5 SIMULATING DIVISION BY 2**31-1
LR R0, R3 PUT R4 INTO R0 FOR NOW
NR R0, 6 SHOULD WE MAKE IT NEGATIVE?
BC R8, F, 0 POSITIVE, KEEP GOING
ER R5, 5 MAKE R5 TRUE NEGATIVE
SLA R4, = CLEAR R4 TO ZERO
FL R5, C1 R5 LESS THAN X**600000000? 0

ENTRY POINT : NORMAL

CNOP 0, 8 BASE REGISTER
BC 11+62  NO  000002520
SLDL R4+8  SHIFT FIRST 8 BITS OF RS INTO R4 AS INDEX & 000002570
JE R4,(R4,R13)  OBTAIN CONSTANT FROM TABLE & 000002570
SRL R4+0,PWD+1  STORE IN SECOND BYTE OF PWD & 000005850
ALR R5,R10  SHIFT REMAINING 24 BITS RIGHT THEN OR ON & 000005850
ST R5,(R7,R6)  EXPONENT TO MAKE ±24 BITS/16 & 000005970
LE E000002590
AE 000002600
FE E0(C7,R6)  STORE IN CALLER'S ARRAY & 000002600
AE E000002610
ST R5,(R7,R6)  STORE NORMAL DEVIATE IN CALLER'S ARRAY & 000002610
LR R6+R0  COPY BACK TO R6 FOR NEXT GO AROUND & 000002610
BR R12  GO TO BLE AND CONTINUE & 000002610
CL R5,C2  NO LESS THAN X'00000000'? & 000002640
SLDL R4+6  SHIFT FIRST 6 BITS OF RS INTO R4 AS INDEX & 000002650
SL R4,(R4,R13)  OBTAIN CONSTANT FROM TABLE & 000002650
SRL R4+0,PWD+1  STORE IN SECOND BYTE OF PWD & 000002650
ALR R5,R10  SHIFT REMAINING 24 BITS RIGHT THEN OR ON & 000002700
ST R5,(R7,R6)  EXPONENT TO MAKE ±24 BITS/16 & 000002710
LE E000002730
AE E000002740
ST R5,(R7,R6)  STORE NORMAL DEVIATE IN CALLER'S ARRAY & 000002740
LR R5,R0  COPY BACK TO Rs FOR NEXT GO AROUND & 000002740
BR R12  GO TO BLE AND CONTINUE & 000002740
CL R5,C9  NO LESS THAN X'E4000000' & 000002780
SLDL R4+12  SHIFT FIRST 12 BITS OF RS INTO R4 & 000002800
SL R4,(R4,R13)  OBTAIN CONSTANT FROM TABLE & 000002800
SRL R4+0,PWD+1  STORE IN SECOND BYTE OF PWD & 000002800
ALR R5,R10  SHIFT REMAINING 24 BITS RIGHT THEN OR ON & 000002850
ST R5,(R7,R6)  EXPONENT TO MAKE ±24 BITS/16 & 000002850
LE E000002860
AE E000002870
ST R5,(R7,R6)  STORE NORMAL DEVIATE IN CALLER'S ARRAY & 000002870
LR R5,R0  COPY BACK TO RS FOR NEXT GO AROUND & 000002870
BR R12  GO TO BLE AND CONTINUE & 000002870
CL R5,C4  NO LESS THAN X'40000000'? & 000002900
SLDL R4+18  SHIFT FIRST 12 BITS OF RS INTO R4 & 000002930
SL R4,(R4,R13)  OBTAIN CONSTANT FROM TABLE & 000002930
SRL R4+0,PWD+1  STORE IN SECOND BYTE OF PWD & 000002930
ALR R5,R10  SHIFT REMAINING 24 BITS RIGHT THEN OR ON & 000002980
ST R5,(R7,R6)  EXPONENT TO MAKE ±24 BITS/16 & 000002980
LE E000002990
AE E000002990
ST R5,(R7,R6)  STORE NORMAL DEVIATE IN CALLER'S ARRAY & 000002990
LR R5,R0  COPY BACK TO RS FOR NEXT GO AROUND & 000002990
BR R12  GO TO BLE AND CONTINUE & 000002990
CL R5,C5  NO LESS THAN X'4F800000'? & 000002990
SHA 000003000
IC 000003000
AL 000003000
ST R5,(R7,R6)  OBTAIN CONSTANT FROM TABLE & 000003000
SRL R5+R6  STORE RS AS SECOND BYTE OF NWRD & 000003000
ALR R5,R10  SHIFT REMAINING 24 BITS RIGHT THEN OR ON & 000003000
ST R5,(R7,R6)  EXPONENT TO MAKE ±24 BITS/16 & 000003000
LE E000003000
AE E000003000
ST R5,(R7,R6)  STORE NORMAL DEVIATE IN CALLER'S ARRAY & 000003000
LR R5,R0  COPY BACK TO RS FOR NEXT GO AROUND & 000003000
BR R12  GO TO BLE AND CONTINUE & 000003000
CL R5,XRD  PASS STARTING VALUE & 000003000
LA R1,(XRD)  LOAD LOW SAVE AREA POINTER & 000003000
LA R1,A10  ARGUMENT LIST FOR CALL TO NRTH & 000003000
LA R1,C15  COPY CASE REGISTER FOR NRTH LINKAGE & 000003000
LR R15,ARMOR  ADDRESSES OF FUNCTION SUBROUTINE NRTH & 000003000
BL R14,R15  BRANCH TO NRTH & 000003000
LE R6,XRD  LOAD SAVE AREA REGISTER & 000003000
ST R5,(R7,R6)  STORE NORMAL DEVIATE IN CALLER'S ARRAY & 000003000
LR R5+R6  STORE RS AS STARTING VALUE & 000003000
LA R13,APL  LOAD NRTH AREA REGISTER & 000003000
LA R13,SA24  RESTORE NRTH AREA POINTER & 000003000
LA R13,XRD  RESTORE NRTH AREA REGISTER & 000003000
LA R4,(X1)  GET STARTING VALUE ADDRESS AGAIN & 000003000
ST R5,R0  STORE AS STARTING VALUE FOR NEXT CALL & 000003000
BR 15+R14  RETURN & 000003000
* 000003000
** ENTRY POINT : SNORM 000003000
** CNDP C.R 000003000
** USING SNORM,R15 000003000
** BRANCH ARROUND ID 000003000
** DC \A15) 000003000
** DC CL\SNORM 000003000
** ST R4+R12,(R13) SAVE REGISTERS IN HIGH SAVE AREA 000003000
** 000003000
ST R13, 5A2+4  ADDRESS OF HIGH SAVE AREA IN LOW SAVE 1R.
LA R14, 5A2  ADDRESS OF LOW SAVE AREA
LA R15, R14 (R2)  ADDRESS OF LOW SAVE AREA IN HIGH SAVE AR.
LM R0, R4, R14 (R2)  LOAD MULTIPLIER, EXPONENT, AND TEST MASK
LA R2, 4  CONSTANT FOR BXLE
LM R3, 8 (R1)  ADDRESSES OF THREE ARGUMENTS
L R5, 0 (R5)  LOAD STARTING VALUE INTO R5
L R6, 1 (R7)  NUMBER OF CONSECUTIVE WORDS TO FILL
SLA R3  CONVERT TO BYTES
ER R5, R2  BACKUP ONE WORD IN CALLER'S ARRAY
LA R13, ATABLE  ADDRESS OF TABLE OF CONSTANTS
LA R14, R8 (R1)  ADDRESS OF SHUFFLING TABLE
LA R12, R8  ADDRESS OF BYBLE
R1 R3  INDEX MASK FOR SHuffling
END 0, 8  ALIGN BXLE LOOP FOR SPEED
SLDA R4, 1  FORM PRODUCT OF A AND X(N-1)
SRL R5, R4  ADD QUOTIENT TO REMAINDER THEREBY
LR R5, R5, R6  SIMULATING DIVISION BY 2**31-1
LR R5, R4  PUT X(N) INTO R5
SLA R4, 7  EXTRACT RIGHT-MOST 7 BITS
SLA R5, R5  CONVERT TO BYTE OFFSETS IN TABLE
R0 R5  R4, R14 (R8)  SELECT 1 RANDOM TABLE VALUE
ST R5, R14 (R8)  REPLACE TABLE VALUE WITH X(N)
XR R5, R0  EXCHANGE R0 AND R5
XR R0, R5  BY EXCLUSIVE ORING
XR R0, R5  SHUFFLE THEM WITH EACH OTHER
RC R4, R5  SHOULD WE MAKE IT NEGATIVE?
RC R4, R5  POSITIVE, KEEP GOING
LMA R5, R5  MAKE R5 TRUE NEGATIVE
SLR R4, R4  CLEAR R4 TO ZERO
CL R5, C1  R5 LESS THAN 1*100000000?
BC 11, F2S  NO
SLOL R4, R14 (R3)  SHIFT FIRST 8 BITS OF R5 INTO R4 AS INDEX
SLA R4, R4  STORE IN SECOND BYTE OF NWRD
SRL R5, R8  SHIFT REMAINING 24 BITS RIGHT THEN ON
SLA R5, R5, R6  STORE REMAINING 24 BITS RIGHT THEN ON
ALR R5, R10  EXPONENT TO MAKE 128 BITS/16
AE R0, R5, R6  EXTRACT CHARACTERISTIC TO FLOATING POINT
AE R0, R17, R6  REGISTER Q AND ADD FRACTION
ST R0  STORE NORMAL DEVIATE IN CALLER'S ARRAY
LR R5, R0  COPY BACK TO R5 FOR NEXT GO AROUND
BR R12  GO TO BXLE AND CONTINUE
F2S  RC R1, R3  R5 LESS THAN 1*1000000000?
BF RC R1, R3  NO
SLOL R4, R14 (R3)  SHIFT FIRST 8 BITS OF R5 INTO R4 AS INDEX
SLA R4, R4  STORE IN SECOND BYTE OF NWRD
SRL R5, R8  SHIFT REMAINING 24 BITS RIGHT THEN ON
SLA R5, R5, R6  STORE REMAINING 24 BITS RIGHT THEN ON
ALR R5, R10  EXPONENT TO MAKE 128 BITS/16
AE R0, R5, R6  EXTRACT CHARACTERISTIC TO FLOATING POINT
AE R0, R17, R6  REGISTER Q AND ADD FRACTION
ST R0  STORE NORMAL DEVIATE IN CALLER'S ARRAY
LR R5, R0  COPY BACK TO R5 FOR NEXT GO AROUND
BR R12  GO TO BXLE AND CONTINUE
F3S  CL R1, R3  R5 LESS THAN 1*10000000000?
BF CL R1, R3  NO
SLOL R4, R14 (R3)  SHIFT FIRST 12 BITS OF R5 INTO R4
SLA R4, R4  STORE IN SECOND BYTE OF NWRD
SRL R5, R10  SHIFT REMAINING 20 BITS RIGHT THEN ON
SLA R5, R5, R6  STORE REMAINING 24 BITS RIGHT THEN ON
ALR R5, R10  EXPONENT TO MAKE 120 BITS/16
AE R0, R5, R6  EXTRACT CHARACTERISTIC TO FLOATING POINT
AE R0, R17, R6  REGISTER Q AND ADD FRACTION
ST R0  STORE NORMAL DEVIATE IN CALLER'S ARRAY
LR R5, R0  COPY BACK TO R5 FOR NEXT GO AROUND
BR R12  GO TO BXLE AND CONTINUE
F4S  CL R1, R4  R5 LESS THAN 1*100000000000?
BF CL R1, R4  NO
SLOL R4, R14 (R3)  SHIFT FIRST 12 BITS OF R5 INTO R4
SLA R4, R4  STORE IN SECOND BYTE OF NWRD
SRL R5, R10  SHIFT REMAINING 20 BITS RIGHT THEN ON
SLA R5, R5, R6  STORE REMAINING 24 BITS RIGHT THEN ON
ALR R5, R10  EXPONENT TO MAKE 120 BITS/16
ST R5,(R7,R6)  STORE IN CALLER'S ARRAY
LE R6,R0,R0  LOAD CHARACTERISTIC TO FLOATING POINT
ST R3,(R7,R6)  STORE IN CALLER'S ARRAY
ST R3,(R7,R6)  STORE NORMAL DEVIATE IN CALLER'S ARRAY
LR R5,R5  COPY BACK TO R5 FOR NEXT GO AROUND
BR R12  GO TO BXLE AND CONTINUE
FSR R5,R5,RD  STORE RS IN ARGUMENT LIST
ST R5,RWD  PASS STARTING VALUE
LA R13,R5,A  LOAD LOW SAVE AREA POINTER
LA R13,R5,T  ARGUMENT LIST FOR CALL TO RND
LR R8,R15  COPY BASE FOR BALR LINKAGE
LR R0,R7,DR  ADDRESS OF FUNCTION SUBROUTINE RND
BALR R4,R15  BRANCH TO RND
LR R15,RR  RESTORE BASE REGISTER
ST R0,(R7,R6)  STORE NORMAL DEVIATE IN CALLER'S ARRAY
LR R5,RWD  NEW STARTING VALUE
LA R13,R8,T  RESTORE R8 TO TABLE OF CONSTANTS
LA R8,TABLE  RESTORE R8 TO ADDRESS OF SHUFFLING TABLE
LR R3,MASK  RESTORE R3 TO INDEX MASK
BXLE R7,R5,16  LOOP AROUND AGAIN
LR R12,R12,R13  RESTORE HIGH SAVE AREA POINTER
LR R4,R0,R13  GET ARGUMENT LIST POINTER AGAIN
ST R0,R7,0(R1)  STORE AS STARTING VALUE FOR NEXT CALL
LM R14,R12,12(R13)  RESTORE THE REGISTERS

* ENTRY POINT : EXPOND
* CNOP 0,8  BASE REGISTER
USING EXPOND,R15  BRANCH AROUND ID
EXPOND R8,R15  BASE REGISTER
DC A1(15)  BRANCH AROUND ID
DC 13H,13H
STM R12,12,12(R13)  SAVE REGISTERS IN HIGH SAVE AREA
ST R8,(R2,R4)  ADDRESS OF HIGH SAVE AREA IN LOW SAVE AREA
LR R2,R13  COPY TO R2
LA R3,A2  ADDRESS OF LOW SAVE AREA
LA R13,R8,(R2)  ADDRESS OF LOW SAVE AREA IN HIGH SAVE AREA
LM R9,R11,A7N  LOAD MULTIPLIER,EXPONENT AND TEST MASK
LA BXLE  CONV_2_BYTES
LM R5,R7,0(R1)  ADDRESSES OF THREE ARGUMENTS
LR R5,R7,0(R1)  LOAD STARTING VALUE INTO R5
LM R3,C1(R7)  NUMBER OF CONSEQUENTIAL WORDS TO FILL
SLA R7  CONV_2_BYTES
SR R0,R2  BACKUP ONE WORD IN CALLER'S ARRAY
LM R7,0(R2)  INITIAL VALUE FOR INDEX REGISTER
LA R13,R2  ADDRESS OF TABLE OF CONSTANTS
LDR 12,N7  ADDRESS OF PAIL
LDDP 4  ALIGN BXLE LOOP FOR SPEED
MAR R4,R9  FORM PRODUCT OF A AND X(N-1)
SLA R4,R5  R4 = REMAINDER ; R5 = REMAINDER
SRL R5,1  ADD QUOTIENT TO REMAINDER THEREBY
LA R4  SIMULATING DIVISION BY 2**31-1
LR R5,R4  PUT X(N) INTO R5
LR R5,R5  COPY R5 INTO R6 FOR NOW
NC R5,R11  SHOULD WE MAKE IT NEGATIVE?
HC R5,R5  POSITIVE KEEP GOING
LNR R4,R5  MAKE R5 TRUE NEGATIVE
SLL R4,R4  CLEAR R4 TO ZERO
FL R5,0,0  RS LESS THAN X*EPS00000000?
FL 11,E2  NO
SLL R4,0  SHIFT FIRST 8 BITS OF RS INTO R4 AS INDEX
IC R4,(R4,R13)  OBTAIN CONSTANT FROM TABLE
ST R4,PWDR  STORE IN SECOND BYTE OF PWDR
SRL R5,8  SHIFT REMAINING 24 BITS RIGHT THEN ON
ALR R5,10  EXPONENT TO MAKE .120 BITS16
ST R5,(R7,R6)  STORE IN CALLER'S ARRAY
LR R6,PWDR  LOAD CHARACTERISTIC TO FLOATING POINT
ST R0,(R7,R6)  STORE EXPONENTIAL DEVIATE IN ARRAY
LR R3,R5,R6  COPY BACK TO RS FOR NEXT GO AROUND
BR R12  GO TO BXLE AND CONTINUE
E2 FL R5,0  RS LESS THAN X*EPS00000000?
FL 11,E2  NO
SLL R4,0  SHIFT FIRST 12 BITS OF RS INTO R4
IC R4,(R4,R13)  OBTAIN CONSTANT FROM TABLE
ST R4,PWDR  STORE IN SECOND BYTE OF PWDR
SRL R5,8  SHIFT REMAINING 24 BITS RIGHT THEN ON
ALR R5,10  EXPONENT TO MAKE .120 BITS16

ST R5,0(R7,R6) STORE IN CALLER'S ARRAY
LE R6,9(R6) LOAD CHARACTERISTIC TO FLOATING POINT
AE R6,9(R7,R6) REGISTER G AND ADD FRACTION
STE R6,9(R7,R6) STORE EXPONENTIAL DEViATE IN ARRAY
LR R6,R6 COPY BACK TO RS FOR NEXT GO AROUND
BR R12 GO TO BXLE AND CONTINUE
E3 ST R5,E9D STORE R5 IN ARGUMENT LIST
ST R5,9(R9) PASS ARGUMENT
LA R1,9(R1) LOAD LOW SAVE AREA POINTER
LR R1,R15 COPY BASE REGISTER FOR RALE LINKAGE
LR R15,9(REP) ADDRESS OF FUNCTION SUBROUTINE REENTRY
BALR BRANCH TO REENTRY
LR R15,RR RESTORE BASE REGISTER
STE R6,0(R7,R6) STORE EXPONENTIAL DEViATE IN ARRAY
L R5,9(R9) NEW STARTING VALUE
LA R1,9(R9) RESTORE R13 TO TABLE OF CONTENTS
N7 SBLE R13,R17 LOOP AGAIN AROUND
L R13,ALJ2 RESTORE HIGH SAVE AREA POINTER
L R13,AX(R13) GET ARGUMENT LIST POINTER AGAIN
LT R4,0(R1) GET STARTING VALUE ADDRESS AGAIN
ST R4,0(R1) STORE AS STARTING VALUE FOR NEXT CALL
LM R4,R12,1(R13) RESTORE THE REGISTERS
BRR 13,A14 RETURN
* MISSING CARD
** ENTRY POINT : SEXPON
CNOP 0,8 USING SEXPON,R15 BASE REGISTER
SEXPN B 121,R15 BRANCH AROUND TO
OC 641,(8) 00005290
GC CL5,SEXPON* 00005290
STM R4,R12,12(R13) SAVE REGISTERS IN HIGH SAVE AREA
ST R4,R15 ADDRESS OF HIGH SAVE AREA IN LOW SAVE AR.
LR R4,R2 COPY TO R2
LA R4,ALJ2 ADDRESS OF LOW SAVE AREA
LM R4,0(R4) ADDRESSED OF LOW SAVE AREA
LA R4,ALJ7 ADDRESS OF LOW SAVE AREA IN HIGH SAVE AREA
LM R4,R11,A75 ADDRESS OF MULTIPLIER EXPONENT AND TEST MASK
LA R2 ALJ2 constant for буле
LM R4,R7,0(R1) ADDRESSES OF THREE ARGUMENTS
L R4,0(R5) LOAD STARTING VALUE IN R5
LM R4,R3,0(R7) NUMBER OF CONSECUTIVE WORDS TO FILL
SLA R4 R3 CONVERT TO BYTES
LR R4,0(R2) BACKUP ONE WORD IN CALLER'S ARRAY
LR R7,9(R9) INITIAL VALUE FOR INDEX REGISTER
LA R6,9(R9) ADDRESS OF TABLE OF CONSTANTS
LA R6,9(R6) ADDRESS OF SHUFFLING TABLE
LA R6,9(R5) ADDRESSES OF XLATE
LR R6,9(R1) INDEX MASK FOR SHUFFLING
cnop 0,4 ALIGNED LOOP FOR SPEED
mrd R4,R0 FORM PRODUCT OF A AND XIN-1)
SKL R5,R1 R5 = QUOTIENT ; R4 = REMAINDER
LA R4 ALJ2 AND QUOTIENT TO REMAINDER THEREBY
AP R4,R5 SIMPLIFICATION BY 2W*1-1
LR R5,XIN INTO RS
NP R4,11 EXTRACT RIGHTMOST 7 BITS
SLA R4 CONVERT TO BYTES DEPENT ON RS
LM R5,0(R4,R8) SELECT RANDOM TABLE VALUE
LT R5,R5,R8 REPLACE TABLE VALUE WITH XIN1
ZA R4,R5 EXCHANGE NO AND RS
KR R5,R4 BY EXCLUSIVE ORING
XR R6,R6 THEM WITH EACH OTHER
NR R4,R11 SHOULD WE MAKE IT NEGATIVE?
SG R5,R16 POSITIVE, KEEP GOING
LNR R5,R5 MAKE R5 TRUE NEGATIVE
SLR R6,15 CLEAR R4 TO ZERO
CL R5,01 RS LESS THAN X*000000001?
BE R5,25 NO
SGL R4,8 SHIFT FIRST 8 BITS OF RS INTO R4 AS INDEX
SGT R5,0(R4,R13) OBTAIN CONSTANT FROM TABLE
SGT R4,E9D+1 STORE IN SECOND BYTE OF PWRD
SAL R5,8 SHiFT REMAINING 24 BITS RIGHT THEN OR ON
ALR R5,17 EXONENT TO MAKE 124 BITS/16
ST R5,9(RE7,R6) STORE IN CALLER'S ARMY
LR R6,R6 LOAD CHARACTERISTIC TO FLOATING POINT
AR R6,0(R7,R6) REGISTER G AND ADD FRACTION
STE R6,9(RE7,R6) STORE EXPONENTIAL DEViATE IN ARRAY
LR R6,R6 COPY BACK TO RS FOR NEXT GO AROUND
BR R12 GO TO BXLE AND CONTINUE
E2S CL R5,02 RS LESS THAN X*1700000001?
BE R5,740 9C
7C 11,E3S NO
SLDL R4,12       SHIF Y FIRST 12 BITS OF R5 INTO R4   000005800
SL R4,D1M      SUBTRACT OFF   000005710
STC R6,DP4,R13   OBTAIN CONSTANT FROM TABLE   000005B80
STC R6,PW0+1   STORE THE SECOND BYTE OF PW0   000005C70
SL R5,B       SHIF T REMAINING 20 BITS RIGHT TH N OR ON   000005C40
ALR R5,R10     EXPONENT TO MAKE .(2C B171)/16   000005C80
ST R5,D1R7,R6   STORE IN CALLER'S ARRAY   000005680
LE FRC,PW0     LOAD CHARACTERISTIC TO FLOATING POINT   000008780
JE FRC,D1(R7,R6)   REGISTER D AND ADD FRACTION   000005820
STE FRC,D1(R7,R6)   STORE EXPONENTIAL DEVIATE IN ARRAY   000008900
LR R4,R6      COPY BACK TO R5 FOR NEXT GO AROUND   000004640
BR R5       GO TO R6 AND CONTINUE   000005910

E5S ST R5,EWRD   STORE RS IN ARGUMENT LIST   000004920
ST R6,WP0      PASS STARTING VALUE   000004930
LA R13,S2    LOAD LOW SAVE AREA POINTER   000005460
LE R1,PLIST   ARGUMENT LIST FOR CALL TO REXP    000005400
LR R6,R15    COPY BASE FOR BRK LINKAGE   000005460
LR R4,R15    ADDRESS OF FUNCTION SUBROUTINE REXP    000005390
BALR R4,R15    BRANCH TO REXP    000005490
ST R5,REXR   STORE REXR REGISTER   000005440
STE FRC,D1(R7,R6)   STORE EXPONENTIAL DEVIATE IN ARRAY   000006090
LR R5,RN0    NEW STARTING VALUE   000006110
LA R6,R ISLE   RESTORE R13 TO TABLE OF CONTENTS   000006070
LA R8,TA BLE   RESTORE RA TO ADDRESS OF SHUFFLING TABLE   000006030
LA R6,INDEX   RESTORE R1 TO INDEX MASK   000004640
AXLE R7,R7+10   LOOP AROUND AGAIN   00000D90
LA R8,R12(R13)   RESTORE HIGH SAVE AREA POINTER   000004660
LR R4,R12(R13)   GET ARGUMENT LIST POINTER AGAIN   000006570
ST R5,GID(R4)   GET STARTING VALUE ADDRESS AGAIN   000006500
WM R13,12(R13)   RESTORE THE REGISTERS   000006590
BCR 15,R14   RETURN   000005110

* * *  CONSTANTS AND STORAGE * * *

SA DS 1BP   000001A80
GAP DS 1BP   000001A80
PICA DS F   000001800
PM2 DC F1247438365*   000001900
A75 DC F168077   000006269
A75N DC F168077   000006269
MASK DC X'0CC00000'   000003240
C1 DC X'04AA000'   000006270
PROC DC X'00000000'   000006270
C2 DC X'00000000'   000006270
C4 DC X'F5EC0000'   000006270
C6 DC X'00000000'   000006270
C8 DC X'00000000'   000006270
C9 DC X'00000000'   000006270
DI DC X'00000000'   000006270
DIM DC X'00000000'   000006270
FWRD DS F   000006440
FRWD DS F   000006440
R1ST DC AL4(F1RD)   000006410
R2ST DC AL4(F1RD)   000006410
ALIST DC AL4(F1RD)   000006410
E1ST DC AL4(F1RD)   000006410
E2ST DC AL4(F1RD)   000006410
E3ST DC AL4(F1RD)   000006410
E4ST DC AL4(F1RD)   000006410
E5ST DC AL4(F1RD)   000006410
E6ST DC AL4(F1RD)   000006410
E7ST DC AL4(F1RD)   000006410
E8ST DC AL4(F1RD)   000006410
E11ST DC AL4(F1RD)   000006410
AINT DC VIPINT   000004460
AS1MP DC VEXPOND   000005400
AFRDC DC AF1RTH   000005400
AREXP DC AF1RTH   000005400
TABLE DC X'1347A505*'.X'326C0AFT*'.X'0DF68288*'.X'769277F1'   000005460
DC X'15482CE1*'.X'28AB0A4*'.X'11942E23*'.X'99CC04EF'   000005460
DC X'25F9F0D4*'.X'164A945F*'.X'07FF8FAB*'.X'174C36F15'   000005460
DC X'164D04*'.X'778B737D*'.X'3C6FBF87*'.X'40F6F8B7'   000005640
DC X'12979CE8*'.X'6356C333*'.X'644999EC'X'764A9B5F'   000005640
DC X'111344C8*'.X'4D023661*'.X'2DF4890'X'7626FD00'   000005640
DC X'125559BE*'.X'76649496'X'62C0C4DB*'.X'13F4F05C'   000005640
DC X'129F0954*'.X'57279332*'.X'249777AD*'.X'065FC0E3'   000005640
DC X'6C93537A*'.X'10C8BF0F*'.X'380B04BF'X'1430A95CE4'   000006610
<table>
<thead>
<tr>
<th>Label</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD</td>
<td>5</td>
</tr>
<tr>
<td>R4</td>
<td>6</td>
</tr>
<tr>
<td>R9</td>
<td>7</td>
</tr>
<tr>
<td>R0</td>
<td>8</td>
</tr>
<tr>
<td>R1C</td>
<td>9</td>
</tr>
<tr>
<td>R11</td>
<td>10</td>
</tr>
<tr>
<td>R12</td>
<td>11</td>
</tr>
<tr>
<td>R13</td>
<td>12</td>
</tr>
<tr>
<td>R14</td>
<td>13</td>
</tr>
<tr>
<td>R15</td>
<td>14</td>
</tr>
<tr>
<td>FR0</td>
<td>15</td>
</tr>
<tr>
<td>FR2</td>
<td>16</td>
</tr>
<tr>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

00007440
00007450
00007460
00007470
00007480
00007490
00007500
00007510
00007520
00007530
00007540
00007550
00007560
00007570
APPENDIX B

Computer Program Listings for Clustering Algorithms
**GENERALIZED AGGLOMERATIVE HIERARCHICAL CLUSTERING ALGORITHMS**

**DOCTORAL DISSERTATION - JANUARY 1978**

**INPUT IS FROM FILE 1 (FTOFDD1) - PROGRAM READS DATA EXACTLY AS WRITTEN BY THE GENERATOR PROGRAM**

**ONE CARD OF PUNCH INPUT DATA SELECTION METHOD = FORMAT (11)**

**PUNCH OUTPUT CONSISTS OF EXTERNAL AND INTERNAL CRITERION STATISTICS**

**METHOD CODES ARE:**

1. SINGLE LINKAGE METHOD
2. COMPLETE LINKAGE METHOD
3. GROUP AVERAGE
4. WEIGHTED AVERAGE
5. CENTROID
6. MEDIAN METHOD
7. WARD'S MINIMUM VARIANCE METHOD
8. LANCE & WILLIAM'S BETA-FLEXIBLE METHOD (BETA=+.25)

**DIMENSION DI(70,70),LKI(70),LKLJ(70),HKORR(111),PKORR(111),**

**CPH=2,EVEC(2x15),XAND(111)**

**REAL RI(110)**

**INTEGER SET,TYPE,DIJM,CLUJNR,TLCSUJN,DENSI**

**LOGICAL NI(10)**

**READ(51,1) METHOD - PRINT HEADINGS - READ DATA**

**READ(51,1) METHOD**

**PRINT 5**

**GO TO (51,52,53,54,55,56,57,58),METHOD**

**PRINT 51**

**GO TO 3**

**PRINT 52**

**GO TO 3**

**PRINT 53**

**GO TO 3**

**PRINT 54**

**GO TO 3**

**PRINT 55**

**GO TO 3**

**PRINT 56**

**GO TO 3**

**PRINT 57**

**GO TO 3**

**PRINT 58**

**GO TO 4**

**DO 5 JSET=1,110**

**READ(I1,N,SET,N,DIJMN,CLUJNR,TLCSN,DENSI,(LKLJ(I),I=1,10)**

**N1=N-1**

**N2=N-1**

**W1W=FALSE**

**INC=0**

**D1=0.0**

**PI1=0.0**

**DO 9 J=2,N**

**W1W=FALSE**

**D1=0.0**

**PI1=0.0**

**9 I=I-1**

**READ(1,6) (D1,I,J=1,N)**

**DO 10 J=2,N**

**W1W=FALSE**

**EVEC=EVEC+1**

**EVEC(I)=B(J1)**

**G1=DI(J1)**

**SET UP ARRAY FOR CRITERION STATISTICS**

**IF N>100 GO TO 15**

**ICLSN=TLCSN**

**LEVEL=N-TLCSN**

**DO 11 J=2,50**

**J1=J1+1**

**DO 11 JJ=1,J1**

**IF N=1 GO TO 11**

**WRAND(JJ,JJ)=FALSE**

**DO 12 J=1,ICLSN2,2**

**M1=LK1(J)**

**M2=LIK1(J+1)**

**END**
M1=M1+1
GO TO 12,13,M=M11,M2
II=CM-1
GO TO 13,J=N1,1
WRAN(1),(G4)<.TRUE.
CONTINUE
12 CONTINUE
CC BEGIN CLUSTERING
15 DO 18 JJ=1,J1
100 IF (WN(JJ)) GO TO 102
N1=N1-1
DO 101 J=1,J1
IF (.NOT.WN(J)) GO TO 104
101 CONTINUE
102 CONTINUE
STOP
104 X=DI(N1,N1)
GO TO 150,J=2,N
IF (WN(J)) GO TO 150
II=I-1
DO 149 J=1,II
IF (WN(J)) GO TO 149
IF (DI(J,J),G1,1) GO TO 149
N1=II
GO TO 149
CONTINUE
149 CONTINUE
150 CONTINUE
IF(JJ=1,X)
LKI(JJ)=N1
LX(JJ)=-N1
WN(JJ)=.TRUE.
DO 159 L=1,N
IF (WN(L)) GO TO 179
DO 170 D=L,N1+1,M(L,N1)
D1(D,L)=D(L,N1)
GO TO 170
170 CONTINUE
179 CONTINUE
DO 180 D=L,N1+1,M(L,N1)
D1(D,L)=0.0
GO TO 180
180 CONTINUE
41 D1(L,N1)=MIN1(D1(L,N1),D1(L,N1))
DIM(L,L)=D1(L,N1)
GO TO 170
42 D1(L,N1)=MAX1(D1(L,N1),D1(L,N1))
DIM(L,L)=D1(L,N1)
GO TO 170
43 D1(L,N1)=P1(N1)+P1(N2)*D1(L,N1)/{P1(N1)+P1(N2)}
DIM(L,L)=D1(L,N1)
GO TO 170
44 D1(L,N1)=5.0*D1(L,N1)+D1(L,N1)
DIM(L,L)=D1(L,N1)
GO TO 170
45 D1(L,N1)=P1(N1)+P1(N2)*D1(L,N1)+P1(N1)*D1(L,N1)+P1(N1)*P1(N2)*D1(L,N1)/{P1(N1)+P1(N2)}
DIM(L,L)=D1(L,N1)
GO TO 170
46 D1(L,N1)=5.0*D1(L,N1)+D1(L,N1)-25*D(N1,N1)
DIM(L,L)=D1(L,N1)
GO TO 170
47 D1(L,N1)=P1(N1)+P1(N2)*D1(L,N1)+P1(N1)+P1(N2)*D1(L,N1)-P1(L1)*D1(L,N1)+P1(L1)*D1(L,N1)/{P1(N1)+P1(N2)}
DIM(L,L)=D1(L,N1)
GO TO 170
48 D1(L,N1)=625.0*D1(L,N1)+D1(L,N1)-25*D(N1,N1)
DIM(L,L)=D1(L,N1)
CONTINUE
170 CONTINUE
179 CONTINUE
181 CONTINUE
LKI(JJ)=J(JJ)
LX(JJ)=-LX(JJ)
II=N1-1
GO TO 181
180 CONTINUE
CONTINUE
STOP
515 CONTINUE
515 CONTINUE
520 L(K1)=K(K1)
L(K1)=L(K1)
GO TO 515
530 CONTINUE
STOP
530 CONTINUE
555 J1=1+J1,N
IF (L(K1,J1)) GO TO 570
CONTINUE
STOP
DO 570 JJ=1,N
IF(LK(JJ).EQ.LK(J)) GO TO 580
CONTINUE
STOP
580 LK(J)=J
CONTINUE
DO 370 I=2,N2
11 I=J,J
DO 371 J=1,1
JJ=J
IF(IX(JJ).NE.LK(JJ)) GO TO 371
CONTINUE
GO TO 370
CONTINUE
INITIALIZE FOR CRITERION STATISTICS
DO 481 IXP=2,N
II=IXP-1
DO 481 IXP=1,II
DII=IxPIXP=0.0
DO 476 IXPIXP=1
JII=JII+I
JJI=IIJX
IXII=IXII+1
DO 486 III=1,JII
JII=JII+1
JII=JII+X
JII=MAXO(LK(III),LX(JIII))
JII=MAXO(LK(III),LX(JIII))
IF(IX(JII)+JII).NE.0.0) GO TO 488
IXII=IXII+1
DII=IXII+IJPI
DII=IXII+I
DII=IXII+JII
CONTINUE
GO TO 477
COMPUTE RAND AND SINGLE PARTITION CRITERION IF CORRECT LEVEL
SUM=0.
SUM=0.
SUM=0.
SUM=0.
CONTINUE
DO 1069 I=2,50
II=I
DO 1069 III=1,II
DII=I
SUM=SUM+EVEG(I
SUM=SUM+EVEG(I
SUM=SUM+EVEG(I
SUM=SUM+EVEG(I
IF(WD(IJ).JII)) GO TO 700
SUM=SUM+1
SUM=SUM+1
SUM=SUM+1
SUM=SUM+1
CONTINUE
GO TO 1069
700 IF(WRAN(JJ,II)) SUM=SUM+1
CONTINUE
RETURN TO PREPARATION FOR HIERARCHICAL CRITERION
IF(LEVEL.EQ.N2) GO TO 802
LEVEL=LEVEL+1
DO 860 IXP=1,LEVEL1,N2
JII=JII(I
JII=JII+1
JII=JII+1
JII=JII+1
IF(JII) GO TO 802
LEVEL=LEVEL+1
DO 870 JXP=1,LEVEL1,N
JII=JII(I
JII=JII+1
JII=JII+1
JII=JII+1
IF(JII) GO TO 802
LEVEL=LEVEL+1
DO 880 I=1,LEVEL1
JII=JII(I
JII=JII+1
JII=JII+1
JII=JII+1
IF(JII) GO TO 802
CONTINUE
DO 830 1=2,50
II=I-1
DO 820 J=1,11
IF(D(I,J).LE.0.0) D(I,J)=1R1(N1)
CONTINUE
C
COMPUTE HIERARCHICAL INTERNAL CRITERION MEASURE
C
IVEC=0
SUM1=0.0
SUM2=0.0
SUM3=0.0
SUM4=0.0
DO 1591 JAA=2,50
JAAA=JAA-1
DO 1592 JAA=1,JAAA
IVEC=IVEC+1
SUM1=SUM1+D(JAA,JAA)
SUM2=SUM2+D(JAA,JAA)*D(JAA,JAA)
SUM3=SUM3+EVECT(IVEC)*EVECT(IVEC)
SUM4=SUM4+EVECT(IVEC)*EVECT(IVEC)*D(JAA,JAA)
1592 CONTINUE
1591 CONTINUE
BOTTOM=SUM3/SUM1*SUM3/1225.1*(SUM4-SUM3*SUM3/1225.1)
MCORR(TYPE)=TOP/BOTTOM
C
PRINT AND PUNCH RESULTS AS APPROPRIATE
C
PRINT 1061,SET,,TYPE,,RAND(TYPE),,PCORR(TYPE),,MCORR(TYPE)
1061 IYPE=NE.11) GO TO 5
1062 IYPE=11.11
LKI(I)=INT(RAND(I)*1000.+5)
LKI(I)=INT(PCORR(I)*1000.+5)
PUNCH 1663,SET,,METHOD,,DIMN,,CLUSNR,,DENSIT,,(LK(I),LK(I),LK(I)).
C
CONTINUE
5 FORMAT(I)
1 FORMAT(2) GENERALIZED AGGLOMERATIVE HIERARCHICAL CLUSTERING ALGORITHM
4 FORMAT(*SET TYPE , RAND INTERNAL.,1 INTERNAL,.,2\,+/)
6 FORMAT(13,,5X,13I3)
8 FORMAT(15H,2)
61 FORMAT(2H SINGLE LINKAGE METHOD)
62 FORMAT(2H COMPLETE LINKAGE METHOD)
63 FORMAT(2H GROUP AVERAGE METHOD)
64 FORMAT(2H WEIGHTED AVERAGE METHOD)
65 FORMAT(4H CENTROID METHOD)
66 FORMAT(5H MEDIAN METHOD)
67 FORMAT(3H MARDI S MINIMUM VARIANCE METHOD)
68 FORMAT(3H LANCE & WILLIAM S BETA-FLEXIBLE METHOD)
1061 FORMAT(5I3,,4X,5I3,,2X,5I3,,6X,5I3,,5I3,,16X,5I3,,6X,5I3)
1063 FORMAT(13,,5X,13I3,,6(3I3),/,,13,,5(3I3))
STOP
END
AGGLOMERATIVE HIERARCHICAL CLUSTERING ALGORITHM

INPUT: READ DATA EXACTLY AS GENERATED

METHOD CODE IS:

PRINT HEADING - READ DATA

SET UP ARRAY FOR CRITERION STATISTICS

BEGIN CLUSTERING

CONTINUE

STOP
DO 150 I=1,N
   IF (N(I)) GO TO 150
   I1 = I-1
   DO 149 J=I+1,N
   IF (W(J)) GO TO 149
   TEST = SUMLNK(J)+SUMLNK(I)+D(J,I)/((P(I)+P(J))*(P(I)+P(J)-1.0)/2.0)
   IF (TEST.GT.X) GO TO 149
   NJ = I
   NJ(J) = .TRUE.
   DO 140 L=I+1,N
   IF (W(L)) GO TO 140
   DI(L,NJ)+D(L,NJ)
   DI(L,NJ) = DI(L,N)
   CONTINUE
   IF (P(N(J))+P(N(J)+1)) SUMLNK(N(J)+SUMLNK(N(J)+1)+D(N(J),N(J)+1))
   CONTINUE
   LKI(I)+LKI(N(J))
   LKI(N(J)) = LKI(I)
   DO 520 I=I2,N
   J = N-I
   IF (LKI(J), .EQ., LK(J(J))) GO TO 515
   CONTINUE
   STOP
   515 DO 530 JJ=KK,J
   IF (LKI(JJ), .EQ., LK(J(J))) GO TO 530
   CONTINUE
   STOP
   530 LKI(JJ) = J
   LK(J(J)) = J
   CONTINUE
   DO 585 I=2,N2
   J = I-1
   DO 545 J=I1,J
   IF (LKI(J), .EQ., LKI(J(J))) GO TO 570
   CONTINUE
   STOP
   570 DO 575 JJ=1,N
   IF (LKI(JJ), .EQ., LK(J(J))) GO TO 575
   CONTINUE
   STOP
   575 LKI(J(J)) = J
   LK(J(J)) = J
   CONTINUE
   DO 595 I=2,N2
   I = 1
   DO 570 J=I1,J
   J = J+1
   IF (LKI(J), .EQ., LKI(J(J))) GO TO 371
   CONTINUE
   STOP
   372 LKI(J(J)) = LK(J(J))
   GO TO 370
   CONTINUE
   CONTINUE
   C
   C INITIALIZE FOR CRITERION STATISTICS
   C
   DO 481 I=1,N
   IF = J(J)=I
   DO 481 I=1,N
   JJ = LK(I)
   JX = J(I)
   JX1 = JX1
   DO 477 JX = JJ + JX1
   JJX1 = JX1
   IF (JX = JJ) GO TO 481
   MDJPL = JPL
   MDJPL = MDJPL
   MDJPL = MDJPL
   C
   C COMPUTE RAND AND SINGLE PARTITION CRITERION IF CORRECT LEVEL
   C
   C
SUM=0.
SUM1=0.
SUM2=0.
SUM3=0.
SUM4=0.
IVEC=0.
DO 1069 I=2,50
  IIF(I.EQ.1) SUM=SUM+1
  IVEC=IVEC+1
SUM3=SUM3+IVEC(IVEC)
SUM4=SUM4+IVEC(IVEC)*IVEC(IVEC)
IF(RAND(JJ,JJ).LT.700) GO TO 1069
SUM1=SUM1+1
SUM2=SUM2+IVEC(IVEC)
IF((J.JJ).LT.1) SUM=SUM+1
GO TO 1669
700 CONTINUE
1069 CONTINUE
RAND(TYPE)=SUM/1225.
TOP=SUM2-SUM1*SUM3/1225.
BOTTOM=SUM4-SUM3*SUM3/1225.
PCOR(TYPE)=TOP/BOTTOM
C RETURN TO PREPARATION FOR HIERARCHICAL CRITERION
C IF(RAND(JJ,JJ).LT.8) GO TO 802
  LEVEL=LEVEL+1
  DO 876 I1PX=LEVEL1,N2
    JX=LK(I1PX)
    JX=JX-1
    JX=JX*JX
    IF(JX.GT.0.9) GO TO 877
    I1PX=I1PX1
    DO 877 I1PX=JX,JX
      I1PX=I1PX1
      DO 888 I1=JX,JX
        JPX1=MINT(LK(I1),LK(I1)*LXP2)
        JPX2=MAX(LK(I1),LK(I1)*LXP2)
        IF(I1PX1(JPX1).LT.0.9) GO TO 888
        I1PX1=I1PX1
        I1PX2=I1PX2
      888 CONTINUE
567 CONTINUE
876 CONTINUE
802 DO 830 I=2,50
  I1=I
  DO 830 J=1,N1
    IF(I1.EQ.0.0) DI(J,J)=IRI(N1)
  830 CONTINUE
C COMPUTE HIERARCHICAL INTERNAL CRITERION MEASURE
C IVEC=0
SUM1=0.
SUM2=0.
SUM3=0.
SUM4=0.
DO 1591 IAA=2,50
  JAA=JAA+1
  IVEC=IVEC+1
SUM1=SUM1+D(IAA,JAA)
SUM2=SUM2+D(IAA,JAA)*D(IAA,JAA)
SUM3=SUM3+IVEC(IVEC)
SUM4=SUM4+IVEC(IVEC)*IVEC(IVEC)
SUM5=SUM5+IVEC(IVEC)*D(IAA,JAA)
1592 CONTINUE
1591 CONTINUE
TOP=SUM3-SUM2*SUM1/1225.
BOTTOM=SUM4-SUM3*SUM3/1225.
MCOR(TYPE)=TOP/BOTTOM
C PRINT AND PUNCH RESULTS AS APPROPRIATE
C PRINT 1061,SET,TYPE,RAND(TYPE),PCOR(TYPE),MCOR(TYPE)
IF(TYPE.NE.11) GO TO 1061
  LK(I1)=INT(RAND(I1)*1000.+5)
  LK(I1)=INT(MCOR(I1)*1000.+5)
1062 CONTINUE
  PRINT 1062,SET,TYPE,LK(I1),LXJ1,LXJ2,LK(J1),LXK(J1),LXK(J2),LXK(J2),J=7,11
CONTINUE
2 FORMAT('AGGLOMERATIVE HIERARCHICAL CLUSTERING ALGORITHM', //)
4 FORMAT('OSET TYPE INTERNAL.1 INTERNAL.2', //)
6 FORMAT('4I3,2X,13I3')
0 FORMAT('10F8.2')
69 FORMAT('AVERAGE LINKAGE WITHIN THE NEW GROUP')
1061 FORMAT('4I3,4X,I2,2X,F5.1,6X,F8.3,6X,F8.3')
1063 FORMAT('13,12,311,6(3I4,1,7),13,12,5(3I4)')
STOP
END
**AGGLOMERATIVE HIERARCHICAL CLUSTERING ALGORITHMS**

PROGRAMMBy GLENN W. MILLIGAN

DOCTORAL DISSERTATION - FEBRUARY 1979

INPUT IS FROM FILE 1 (PT01F001) - PROGRAM READS DATA EXACTLY AS
WRITE BY THE GENERATOR PROGRAM

PUNCH OUTPUT CONSISTS OF EXTERNAL AND INTERNAL CRITERION STATISTICS

THE METHOD CODE IS:
10-MINIMUM TOTAL WITHIN GROUP SUM OF SQUARES IN THE NEW CLUSTER

-----------------------------------------------

DIMENSION D(70,70),LK(70),LJK(70),LKI(70),MCORR(11),PCORR(11),
CP(170),VECT(415),RAND(11),SUMSQ(70)

REAL R(70)

INTEGER SET,TYPE,DIMNR,CLUSNR,TCLUSN,DENSIT

LOGICAL W(70),WD(70,70),WRAND(50,50)

PRINT HEADINGS - READ DATA

METHOD=10
PRINT 2
PRINT 69
PRINT 4
DO 5 JSET=1,1180
READ(1,60) JSET,TYPE,N,DIMNR,CLUSNR,TCLUSN,DENSIT,LKI(I),I=1,10
N=I+1
J=I+1
K=I+1
I=I+1
DO 1 J=1,1180
SEND J=J+1
5 CONTINUE

SET UP ARRAY FOR CRITERION STATISTICS

IF(TYPE.GT.1) GO TO 15
ICLUSN=TCLUSN+2
LEVEF=I-1
DO 11 J=2,50
J=J+1
11 CONTINUE

IF TYPE.GT.1) GO TO 15
ICLUSN=TCLUSN+2
LEVEF=I-1
DO 11 J=2,50
J=J+1
11 CONTINUE

BEGIN CLUSTERING

DO 100 I=1,N
DO 102 NI=2,N
IF (WINI) GO TO 102
102 CONTINUE
104 CONTINUE
100 CONTINUE
102 CONTINUE
104 CONTINUE

CONTINUE

BEGIN CLUSTERING

DO 100 I=1,N
DO 102 NI=2,N
IF (WINI) GO TO 102
102 CONTINUE
104 CONTINUE

CONTINUE

BEGIN CLUSTERING

DO 100 I=1,N
DO 102 NI=2,N
IF (WINI) GO TO 102
102 CONTINUE
104 CONTINUE

CONTINUE
IF (W(IJ)) GO TO 150
DO 149 J=1,11
IF (W(IJ)) GO TO 140
TEST=SUMSQ(I)+SUMSQ(J)+D(I,J)
IF (TEST.GT.X) GO TO 149
NJ=1
NJ=J
2 TEST
149 CONTINUE
RI(IJ)=X
LKI(IJ)=NJ
WNIJ= .TRUE.
DO 150 L=1,N
IF (W(L)) GO TO 179
DLNIJ=(P(L)+P(NJ))*D(L,NJ)+P(L)*D(NI,NJ))/
C(P(NJ))+P(NJ)*P(L))
D(NI,L)=D(L,NI)
179 CONTINUE
P(NJ)=P(NJ)+P(NJ)
SUMSQ(NI)=SUMSQ(NI)+SUMSQ(NJ)+D(NI,NJ)
181 CONTINUE
LKI(L)=LKI(NI)
LKI(L)=LKI(NI)
DO 550 I=2,NI
JJ=J
DO 510 KK=1,JJ
IF(LK(JJ),EQ,LK(KK)) GO TO 515
CONTINUE
STOP
510 DO 530 K=KK,JJ
(I=I+K)
530 LK(K)=LK(I)
DO 550 I=1,N2
DO 565 JJ=1,N
IF(LK(JJ),EQ,LKI(I)) GO TO 570
CONTINUE
STOP
570 DO 575 JJ=1,N
IF(LK(JJ),EQ,LKI(JJ)) GO TO 580
CONTINUE
STOP
580 LKI(JJ)=J
LKI(JJ)=J
585 CONTINUE
DO 370 I=2,N2
I=I-1
DO 370 J=1,JJ
J=J-1
IF(LK(JJ),NE,LKI(JJ)) GO TO 371
372 LK(JJ)=LKI(JJ)
GO TO 370
371 CONTINUE
370 CONTINUE
C FOR CRITERION STATISTICS
C DO 481 IIXP=2,N
I=IIXP-1
DO 481 IIXP=1,11
HI1P=0.0
DO 476 J=1,LEVEL
JKL=LKI(IIXP)
JX=LKI(IIXP)
XXI=JX-1
DO 477 IX=JKL
HI1P=HI1P+JX
477 CONTINUE
476 CONTINUE
DO 486 I=1,LEVEL
JXP=LKI(I)
JPL=MAX(JK,JX)
486 CONTINUE
CONTINUE
477 CONTINUE
476 CONTINUE
C COMPUTE RAN AND SINGLE PARTITION CRITERION IF CORRECT LEVEL
SUM=0
SUM1=0
SUM2=0
SUM3=0
SUM4=0
IVEC=0
DO 1069 I=2,50
   1069 IVEG=IVEG+1
   SUM=SUM+IVEC(IVEC)
   SUM1=SUM+IVEC(IJ1)*IVEC(IVEG)
   IF(WIJ(JJ,II))GO TO 700
   SUM2=SUM+IVEC(IVEC)
   IF(WIJ(JJ,II))SUM=SUM+1
   GO TO 1049
   700 CONTINUE
   SUM=SUM+1
   IF(WRAND(JJ,II))SUM=SUM+1
   CONTINUE
   REALTYPE=SUM1/1225
   SUM2=SUM1*SUM3/1225
   SUM4=SUM2*SUM3/1225
   CONTINUE
   IF(LEVEL(NE,2))GO TO 802
   JLEVEL1=LEVEL1
   JJX=LKJ1(IIP1X)
   LKJ1=JX
   CONTINUE
   GO TO 802
   JJX=LKJ1(IIP1X)
   LKJ1=JX
   JJX=JX
   DO 877 JX=1,JX1
   JX=JX+1
   CONTINUE
   DO 888 JX=1,JX1
   JJX=LKJ1(IIP1X)
   LKJ1=JX
   JJX=JX
   CONTINUE
   CONTINUE
   DO 830 I=2,50
   I=I-1
   CONTINUE
   IF((J1,J1).EQ.0.0) D(J1,J1)=IR1(N1)
   CONTINUE
   CONTINUE
   SUM1=0.0
   SUM2=0.0
   SUM3=0.0
   SUM4=0.0
   DO 1591 JAA=2,50
   JJAA=JAA-1
   DO 1592 JAA=1,JJAA
   IVEG=IVEC+1
   SUM=SUM+DIAA(JAA)
   SUM2=SUM+DIAA(JAA)*DIAA(JAA)
   SUM3=SUM+IVEC(IVEC)
   SUM4=SUM+IVEC(IJ1)*IVEC(IVEG)
   SUM5=SUM+IVEC(IVEG)*DIAA(JAA)
   CONTINUE
   CONTINUE
   GO TO 1591
   SUM=SUM*SUM3/1225
   CONTINUE
   PRINT 1061,TYPE,RAND(TYPE),PCORR(TYPE),MCORR(TYPE)
   IF(TYPE=NE)GO TO 5
   CONTINUE
   LKJ1=INT(RAND(1)*1000.+.5)
   LKJ1=INT(PCORR(1)*1000.+.5)
   CONTINUE
   PUNCH 1065,SET,METHOD,DMNR,SLNSR,DHFT,N(LK1,JJ1),N(LK1,JJ1),
      CONTINUE
2 FORMAT(' AGglomerative Hierarchical Clustering Algorithm',//)
4 FORMAT('* OSEI, TYPE RAND INTERNAL,1 INTERNAL,2',//)
6 FORMAT('13,3X,13,3)
8 FORMAT('1053,1)')
89 FORMAT(' MINIMUM TOTAL WITHIN GROUP SUM OF SQUARES IN THE NEW CLUS
CTE5')
1061 FORMAT('* 13,4X,12,2X,F5.3,6X,F6.3,6X,F6.3)
1063 FORMAT('13,12311,61311,113,1251311))
STOP
END
AGGLOMERATIVE HIERARCHICAL CLUSTERING ALGORITHMS
PROGRAMMED BY GLENN W. MILLIGAN
DOCTORAL DISSERTATION - FEBRUARY 1978
INPUT IS FROM FILE I (FTOI.FDO) - PROGRAM READS DATA EXACTLY AS
WRITTEN BY THE GENERATOR PROGRAM
PUNCH OUTPUT CONSISTS OF EXTERNAL AND INTERNAL CRITERION STATISTICS
THE METHOD CODE IS:
1 - MAXIMUM AVERAGE WITHIN GROUP SUM OF SQUARES IN THE NEW
   CLUSTER

**********************************************************************
DIMENSION D(70,70), L(70), LK(70), LK(70), LKI(70), HC(11), PC(11),
CP(70), IVEC(245), RAND(11), SUMSQR(70)
REAL IR(70)
INTEGER SET, TYPE, DIMNR, CLUSNR, TCLUSN, DENSIT
LOGICAL WTI(70), WD(70,70), WRAND(50,50)

PRINT HEADINGS - READ DATA
METHOD=11
PRINT 1
PRINT 69
PRINT 4
DO 2 JST=1, 1188
READ(1, 6) SET, TYPE, N, DIMNR, CLUSNR, TCLUSN, DENSIT, LKI(I), I=1, 10
N(I)=N(I)+1
W(I)=.FALSE.
IVEC=0
SUMSQR(1)=0.0
P(I)=2.0
DO 9 I=1, N
SUMSQR(I)=.0
W(I)=.FALSE.
IVEC=1
PRINT 1
IVEC=IVEC+1
DO 10 J=1, JST
READ(1, 8) D(J, I), J=1, 11
DO 9 J=1, 11
W(J)=.FALSE.
IVEC=IVEC+1
IVEC=IVEC+1
D(I, J)=D(J, I)

SET UP ARRAY FOR CRITERION STATISTICS
IF(TYPE.GT.1) GO TO 15
TCLUSN=CLUSNR+1
LEVELS-N-1
GO TO 11
TCLUSN=CLUSNR+2
LEVELS=N-1
GO TO 11
11 JI=JI+1
WRAND(JI, JJ)=.FALSE.
DO 12 JM=1, TCLUSN
M=LI(JJ)
M=M*M
II=II+1
GO TO 13
12 JG=JG+1
GO TO 13
13 JI=JI+1
WRAND(I, JG)=.TRUE.
CONTINUE
CONTINUE

BEGIN CLUSTERING
DO 14 J=1, N
DO 10 NJ=2, N
IF (M(J).GT.WI(J)) GO TO 102
NJ=NJ+1
GO TO 10
10 CONTINUE
CONTINUE
STOP
104 X=SUMSQR(NJ)+SUMSQR(NJ)+D(NI, NJ)/(P(NI)+P(NJ))
DO 150 I=2,N
IF (W(I)) GO TO 149
I=I-1
DO 140 J=1,I
IF (W(J)) GO TO 149
TEST=(SUMSQ(I)*SUMSQ(J)+D(I,J))/(P(I)*P(J))
IF (TEST.GT.X) GO TO 149
I=I+1
J=J+1
CONTINUE
149
STOP

150 CONTINUE
LK(I)=X
LK(I,J)=NJ
NJ=TRUE.
DO 179 L=1,N
IF (W(L)) GO TO 179
(P(NI)*P(NJ)+P(L))
D(NJ)=D(L,NJ)
179 CONTINUE
P(NJ)=P(NI)+P(NJ)
SUMSQ(NJ)=SUMSQ(NJ)+SUMSQ(NJ)*D(NI,NJ)
181 CONTINUE
LK(I,J)=LK(I,J)
LK(J,J)=LK(I,J)
D(I,J)=D(J,J)
DO 550 I=2,N
J=I+1
DO 555 J=I+1,N
IF (LK(J,J).EQ.LK(I,I)) GO TO 515
510 CONTINUE
STOP
515 DO 530 K=1,KK
K(I)=K(I)+K
530 CONTINUE
LK(K)=LK(K)
DO 560 K=1,N
DO 565 J=1,N
IF (LK(J,J).EQ.LK(K)) GO TO 570
565 CONTINUE
STOP
570 DO 575 J=1,N
IF (K(I,J).EQ.LK(J)) GO TO 580
575 CONTINUE
STOP
580 LK(I,J)=J
LK(J,I)=J
585 CONTINUE
DO 370 I=2,N
J=I-1
DO 375 J=1,11
J=J-1
IF (LK(J,J).NE.LK(I,I)) GO TO 371
372 LK(I,J)=LK(J,I)
GO TO 370
375 CONTINUE
370 CONTINUE
C INITIALIZE FOR CRITERION STATISTICS
C
DO 481 IXP=2,N
II=II+1
481 DO 481 IJP=1,II
DIIP=II+1-P
DO 476 IJP=1,LEVEL
JI=K(IJP)
JX=K(JX)
JJP=II+1
II=II+1
476 DO 477 IXP=JX,JX+1
JXX=IJP+1
DO 480 IJP=IJP+1
JJP=JJP+1
IF (JJP.EQ.IJP+1) GO TO 488
480 DO 488 IJP=IJP+1
JJP=JJP+1
JJP=JJP+1
IF (JJP.EQ.IJP+1) GO TO 488
488 CONTINUE
477 CONTINUE
476 CONTINUE
C COMPUTE RAND AND SINGLE PARTITION CRITERION IF CORRECT LEVEL
C
SUM=0.0
SUM1=0.0
SUM2=0.0
SUM3=0.0
SUM4=0.0
IVEC=0.0
DO 1069 I=2,50
II=I-1
DO 1069 J=J+I
IVEC=IVEC+1
SUM3=SUM3+EVEC(IVEC)
SUM4=SUM4+EVEC(IVEC)*EVEC(IVEC)
IF(WD(J,J)) GO TO 700
SUM=SUM+1
SUM2=SUM2+EVEC(IVEC)
IF(NI(NI)=RAND(J,J)) SUM=SUM+1
GO TO 1069
C
1069 CONTINUE
SUM=SUM/1225.0
CONTINUE
C
RETURN TO PREPARATION FOR HIERARCHICAL CRITERION
C
IF(LEVEL.EQ.N2) GO TO 802
LEVEL=LEVEL+1
DO 876 I=1,LEVEL+1,N2
JX=LK(J)(I,PA)
JX=LK(J)(I)
JX=JX+1
DO 877 I=1,LEVEL+1,N2
JX=LK(J)(I,PA)
JX=LK(J)(I)
JX=JX+1
DO 878 I=1,LEVEL+1,N2
JX=LK(J)(I,PA)
JX=LK(J)(I)
JX=JX+1
GO TO 878
C
888 CONTINUE
C
877 CONTINUE
C
876 CONTINUE
C
DO 880 I=2,50
II=I-1
DO 880 J=1,II
IF(D(I,J).EQ.0.0) D(J,J)=TR(N1)
CONTINUE
C
C
COMPUTE HIERARCHICAL INTERNAL CRITERION MEASURE
C
IVEC=0.0
SUM=0.0
SUM2=0.0
SUM3=0.0
SUM4=0.0
SUM5=0.0
DO 1591 J AA=2,50
JAA=J AA-1
DO 1592 I AA=1,J AA
IVEC=IVEC+1
SUM1=SUM1+D(IAA, JAA)
SUM2=SUM2+D(I AA, J AA)*D(IAA, J AA)
SUM3=SUM3+EVEC(IVEC)
SUM4=SUM4+EVEC(IVEC)*EVEC(IVEC)
SUM5=SUM5+EVEC(IVEC)*EVEC(IVEC)*D(IAA, J AA)
CONTINUE
C
1591 CONTINUE
C
SUM=SUM1+SUM2+SUM3+SUM4+SUM5
CONTINUE
C
C
PRINT AND PUNCH RESULTS AS APPROPRIATE
C
PRINT 1061, SET, TYPE, RAND(TYPE), PCHRRTYPE, MCCHRTYPE
IF(TYPE.WE.1) GO TO 5
DO 1062 I=1,11
LK(I)=INT(RAND(I)*10000.0)
LK(I)=INT(RAND(I)*10000.0)
PUNCH 1063, SET, METHOD, DIMMR, CLUSNR, DENSI, LK(I), LK(I), LK(I), LK(I), J=1,11
C
1062 CONTINUE
C
END
CONTINUE
AGGLOMERATIVE HIERARCHICAL CLUSTERING ALGORITHM

FORMAT('SET TYPE RAND INTERNAL.1 INTERNAL.2')
FORMAT(4I3,3X,I3I3)
FORMAT(10F8.2)
FORMAT('MINIMUM AVERAGE WITHIN GROUP SUM OF SQUARES IN THE NEW
CLUSTER')
1061 FORMAT(13,4X,12.2F,5.3,6X,6.3,6X,6.3)
1063 FORMAT(13,12.3I1,6F314),/,13,12.5(3I4))
STOP
END
NON-HIERARCHICAL CLUSTERING ALGORITHMS - K-MEANS PROCEDURES
PROGRAMMED BY GLENN W. MILLIGAN
DOLPHIN DISSERTATION - FEBRUARY 1978
INPUT IS FROM FILE I (FT01EQ1) - PROGRAM READS DATA EXACTLY AS
PREWRITTEN BY THE GENERATOR PROGRAM
PUNCH OUTPUT CONSISTS OF EXTERNAL AND INTERNAL CRITERION STATISTICS
THE METHOD CODE IS:
12-MACHQUEEN'S ORIGINAL K-MEANS PROCEDURE

COMMON NUMBR(5),MENBR(70),DATA(700)
DIMENSION PCORR(71),RAND(71),IEVEC(71),CHIMED(71),LIST(70)
DATA CHIMED /20.0,0.0,0.0,357.9,0.0,3.348,0.0,7.348/
LOGICAL M1(70),M2(70),M3(70)
INTEGER SET,TYPE,IMIN,CLMNRS,NC,NSNINT,DENSIT

CALL LLRAND
ISEQ=55436385
PRINT 1
PRINT 62
PRINT 4
M1(QU)=12
UU=J+KJ=1
IEVEC
READ (1,10) SET,TYPE,NE,DMIN,CLMNRS,NC,DENSIT,MENBR(1),I=1,10
IF (TYPE.EQ.6) NV=NV+1
IF (TYPE.EQ.7) NV=NV+2

SET UP ARRAYS FOR CRITERION STATISTICS
IF (TYPE.GT.11) GO TO 15
ILLUS2=NC+2
DO 11 J=2,50
J1=J-1
DO 11 J=1,NC+2
WRAND(J,J)=.FALSE.
M1=MEMBR(J)
M2=MEMBR(J+1)
M1=M1+1
DO 12 I=1,NC+1
II=1+I
DO 12 I=I+1,50
WII(J,J)=.TRUE.
CONTINUE
12 CONTINUE
DO 15 J=1,NC+2
DO 15 J=1,50
WII(J,J)=.FALSE.
CONTINUE
15 CONTINUE
CALL THE KMEAN SUBROUTINE
CALL KMEAN (NE,NV,NC,MEMBR,CREASED,TYPE)
UPDATE THE LOGICAL ARRAY
NE=NE-1
DO 21 K=1,NE
DO 21 K=1,NE
IF (MENBR(K).EQ.NE) GO TO 21
K1=K
DO 21 K=K1,K2,1
IF (MENBR(K).EQ.NE) GO TO 21
WII(K,K1)=.TRUE.
WII(K,K1)=.TRUE.
CONTINUE
21 CONTINUE
CONTINUE
CONTINUE

COMPUTE INTERPOINT DISTANCES
IEVEC=0
DO 40 J=2,50
II=I-1
DO 40 J=I,11
IEVEC=IEVEC+1
CONTINUE
DIST = 0.0
GO TO 177

7
DO 3 I = 1, NV
J = NV (J - 1) * K
J = NV (J - 1) + 1
DIST = DIST + DATA (J2) - DATA (J1)
DATA (J2) = DATA (J1)
3 CONTINUE

8
S = 0.0
S = S + DATA (J1)
8 CONTINUE

DIST = DIST + S
DATA (J1) = DATA (J1) + S
S = S + S
STOP

111
GO TO 201

121
GO TO 201

701
COMPUTE RAND AND SINGLE PARTITION CRITERION STATISTICS

C
SUM = 0.0
SUM1 = 0.0
SUM2 = 0.0
SUM3 = 0.0
SUM4 = 0.0
IVEC = 0
DO 106 I = 1, 125
106 SUM = SUM + IVEC(I)
105 SUM1 = SUM1 + IVEC(I)
104 SUM2 = SUM2 + IVEC(I)
103 SUM3 = SUM3 + IVEC(I)
102 SUM4 = SUM4 + IVEC(I)
101 CONTINUE

700
GO TO 700

701
IF (RANU(JJ)) GO TO 700

702
IF (RANU(JJ)) SUM = SUM1

703
CONTINUE

C
PRINT AND PUNCH RESULTS AS APPROPRIATE

C
PRINT 1006, SET, TYPE, RAND (TYPE), PCOK (TYPE)
IF (TYPE = 'E') GO TO 5
1006 CONTINUE

1010
IF (TYPE = 'E') GO TO 5

1020
CONTINUE
DIMENSION ISSEED(1), NMBERN(1), CHIMED(4), TOTAL(76), CENTRI(50)
DATA CHIMEU /0.0,0.0,0.3357,0.0,0.5348,0.0,0.7348/
INTEGER TYPE
C
ESTABLISH INITIAL PARTITION
C
CALL RANDOM(ISSEED, ASEED, NC)
DO 10 IM=1,NC
NBER(IM)=INT(ASEED(IM)+(IM-1)+1)
10 IM=IM+2,NC
IF(J2=1,IM)
IF(NMBER(IM1+1)=NMBER(IM2)) GO TO 21
CONTINUE
C
READ THE DATA SET
K=1
DO 11 J=1,NE
KM=M=K=N=1
READ (1,110) (DATA(KM),KM=K,MM)
K=M=N+1
11 CONTINUE
C
SET UP THE SEED POINTS - THE DATA UNIT WITH SEQUENCE NUMBER NUMBP(J)
IS USED AS THE J-TH SEED POINT
FIRST MOVE THE SELECTED RANDOM SEEDS TO THE FIRST NC POSITIONS
IN THE DATA LIST
C
DO 15 J=1,NC
DO 15 J=1,NC
TEMP=DATA(J)*NMBER(J-1)+1
DATA(J)=DATA(J-1)+1
15 CONTINUE
C
SET THE INITIAL SEED POINTS AT THE FIRST NC DATA UNITS
K=1
DO 20 K=1,NC
CENTRI(K)=DATA(K)
TOTAL(K)=DATA(K)
20 CONTINUE
C
SET THE NUMBER OF ENTITIES IN EACH CLUSTER TO 1
DO 30 J=1,NC
NUMBP(J)=1
30 CONTINUE
C
ASSIGN EACH DATA UNIT TO NEAREST CENTROID AND UPDATE AFTER EACH
ASSIGNMENT
KK=NC+1
DO 40 K=KK,NE
J2=1
40 CONTINUE
C
COMPUTE DISTANCE TO FIRST CLUSTER CENTROID
DIST=G.G
DO 50 J=1,1,1,1,1,1,1,1,1,1,9,11,1,1,1,1,1,1,1,1,1,1,1,1,1
DO 50 K=1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
DIST=SQRT(DIST+DATA(K3)-CENTRI(J3))DATA(K3)-CENTRI(J3)
50 DIST=SQRT(DIST)
GO TO 70
C
DO 60 K=1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
DO 60 J=1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
C
COMPUTE DISTANCE TO FIRST CLUSTER CENTROID
DIST=G.G
DO 50 J=1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
DO 50 K=1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
DIST=SQRT(DIST+DATA(K3)-CENTRI(J3))DATA(K3)-CENTRI(J3)
50 DIST=SQRT(DIST)
GO TO 70
C
END OF PROGRAM
DO 12 J=1,NV
  K3=K2+1
  J3=J2+1
12  DIST=DIST*(DATA(K3)-CENTR(J3))*(DATA(K3)-CENTR(J3))
    DIST=(DIST+CMINVDIST)/CMINEDIST+DIST-1.0
    JREF=J3
    JREF=J3

TEST DISTANCES TO REMAINING CLUSTER CENTROIDS
DO 40 J=2,NV
  DIST=0
  GO TO 42
4  JUS=(J*J-4)*JUS+JUS
  JREF=J
  GO TO 30
40  DIST=DIST+DATA(K2)-CENTR(J2)
  DIST=SQRT(DIST)
  GO TO 301
5  S1=J
  S2=J
  S3=J
  S4=J
  S5=J
  S6=J
  S7=J
  S8=J
  S9=J
  S10=J
  S11=J
  S12=J
  S13=J
  S14=J
  S15=J
  S16=J
  S17=J
  S18=J
  S19=J
  S20=J
  S21=J
  S22=J
  S23=J
  S24=J
  S25=J
  S26=J
  S27=J
  S28=J
  S29=J
  S30=J
  S31=J
  S32=J
  S33=J
  S34=J
  S35=J
  S36=J
  S37=J
  S38=J
  S39=J
  S40=J
  UU=J
  VV=J
  WW=J
  XXX=J
  YYY=J
  ZZZ=J
  RRR=J
  TTT=J
  UU=J
  VV=J
  WW=J
  XXX=J
  YYY=J
  ZZZ=J
  RRR=J
  TTT=J

14  DIST=DIST+DATA(K3)-CENTR(J3)
    DIST=SQRT(DIST)
    GO TO 301
6  J3=J3+1
14  DIST=DIST+DATA(K3)-CENTR(J3)
    DIST=SQRT(DIST)
    GO TO 301
DO 50 J=1,NV
  JUS=(J*J-4)*JUS+JUS
  JREF=J
  GO TO 30
50  JUS=(J*J-4)*JUS+JUS
  JREF=J
  GO TO 30

ALLOCATE DATA UNIT K TO CLUSTER JREF
NUMBK(JREF)=NUMBK(JREF)+1
UJ=UJ+1
UU=UU+1
JREF=JREF
TOTAL(J)=TOTAL(J)+DATA(K2)
CENTR(J)=TOTAL(J)/NUMBK(JREF)
CONTINUE

ALLOCATE DATA UNITS TO INITIAL CONFIGURATION
REALLOCATE DATA UNITS TO FIXED SEED POINTS
DO 60 J=2,NV
  NUMBK(J)=0
  K1=NC+J
60  UJ=UJ+1
  TOTAL(K)=0
  JREF=J
  REALLOCATE DATA UNITS
  K1=K
  UJ=UJ+1
  K2=K+1
  JREF=J
  COMPUTE DISTANCE TO FIXED CLUSTER CENTROID
  DIST=0
  GO TO 72
7  DIST=0
  GO TO 72
  K3=K2+1

J3=J3+1-1
DIST=JIST*(DATA(K3)-CENTR(J3))+(DATA(K3)-CENTR(J3))
DIST=SQRT(DIST)
GO TO 401

K=K+1-NV
J=J+1-NV
S1=S1+1
S2=S2+1
S3=S3+1
S4=S4+1
S5=S5+1
K3=K3+1
J3=J3+1
S1=S1+1
S2=S2+1
S3=S3+1
S4=S4+1
S5=S5+1

TEST DISTANCES TO REMAINING CENTROIDS

DO J=2,NC
J3=J3+1-NV
DIST=0.0
GO TO 30,NV
K3=K3+1
J3=J3+1
DIST=JIST*(DATA(K3)-CENTR(J3))+(DATA(K3)-CENTR(J3))
DIST=SQRT(DIST)
GO TO 201

S1=S1+1
S2=S2+1
S3=S3+1
S4=S4+1
S5=S5+1
K3=K3+1
J3=J3+1
S1=S1+1
S2=S2+1
S3=S3+1
S4=S4+1
S5=S5+1

ALLOCATE DATA UNIT K TO CLUSTER JREF
N=NUM(JREF)=N=NUM(JREF)+1
MEM(K)=JREF
TDIST=TDIST+1

CONTINUE

RESET DATA AND MEMBR ARRAY BACK TO ORIGINAL VALUES
DO 100 J=1,NC
  TEMP=MEMB(J)
  MEMB(J)=MEMB(NMBER(J))
  MMBER(NMBER(J))=TEMP
DO 100 J=1,NV
  TEMP=DATA(NV*(MBER(J)-1)+1)
  DATA(NV*(MBER(J)-1)+1)=DATA(NV*(J-1)+1)
CONTINUE
100 FORMAT(FUP8.3)
RETURN
END
**HIERARCHICAL CLUSTERING ALGORITHMS - K-MEANS PROCEDURES**

**THE METHOD CODE IS:**

1. THE FORGET'S PROTOCOL

**COMMON VARIABLES:**
- IHMR(70), JUAR(70), JUAR(70), JUAR(70)

**DIMENSION PROPERTIES:**
- IHMR(1), JUAR(1), JUAR(1), JUAR(1)

**DATA INPUT:**
- DATA ALPHABETICAL ORDER, DATA FIXED

**INTEGER TYPE:**
- IHMR, JUAR, JUAR, JUAR, JUAR

**THE METHOD CODE IS:**

1. THE FORGET'S PROTOCOL

**COMMON VARIABLES:**
- IHMR(70), JUAR(70), JUAR(70), JUAR(70)

**DIMENSION PROPERTIES:**
- IHMR(1), JUAR(1), JUAR(1), JUAR(1)

**DATA INPUT:**
- DATA ALPHABETICAL ORDER, DATA FIXED

**INTEGER TYPE:**
- IHMR, JUAR, JUAR, JUAR, JUAR

**SET UP ARRAYS FOR CRITERION STATISTICS**

**CALL THE K-MEANS ROUTINE**

**UPDATE THE LOGICAL ARRAY**

**COMPUTE INTERPOINT DISTANCES**

**END**
DIST = u, (G)
GU TO (7, 7, 7, 7, 7, 7, 7, 7, 7, 7, TYPE)
7 DIST = u, (G)
11 DIST = u, (G)
11 DIST = u, (G)
COMMON NUMB(K), NM=MBR(K), DATA(70),
INTEGER AS, D, R, I, J, K, M, N, NE, K1, K2, K3, K4, N1, N2, N3,
DATA (1,2,3,4,5,6,7,8,9)/
INTEGER TYPE

READ THE DATA SET
30  K1=1
   DO 10 J=1,N
      K1=K1+1
      CALL RNDMR(N,1,AS)
      DATA(K1),K1=K1+1
10   CONTINUE

ESTABLISH INITIAL PARTITION
11  CALL RNDPMR(1,SEED,NG)
   DO 20 I=1,NG
      NUMS(I)=INITIALIZE(I)*(NG-1)+1
   DO 20 I=1,NG
      NUMS(I)=NUMS(I)+1
20   CONTINUE

SET UP THE SEED POINTS - THE DATA UNIT WITH SEQUENCE NUMBER NUMP(I) IS USED AS THE J-TH SEED POINT
   DO 30 J=1,NG
      J1=J(J=1)*N1
      J2=J(J=1)*N2
      CENTR(J1+1)=DATA(I=J+1)
30   CONTINUE

INITIALIZE ARRAY:
   DO 50 I=1,NG
      M=255
      DO 50 I=1,NG
      M=255
50   CONTINUE

BEGINNING OF MAIN LOOP
   J1=0
   DO 60 I=1,NG
      NUMS(J1)=I
      J1=J1+1
60   CONTINUE

TOTAL (J1)=NG
   N1=N1+1

ALLOCATE EACH DATA UNIT TO THE NEAREST CLUSTER CENTROID
   K1=0
   DO 80 J=1,NE
      K2=1
50   CONTINUE

COMPUTE DISTANCE TO FIRST CLUSTER CENTROID
   DIS1=0.0
   GO TO 110
1  0   J1=J1+1
   K3=K3+1
110  J1=J1+1
   DIS1=SQRT(DIST)
   GO TO 7
2
   S4=0.0
   S5=0.0
   S6=0.0
   S7=0.0
   K3=K3+1
   DIS1=DIS1+DATA(K3)*CENTR(J3)
   GO TO 110
7
   S4=S4+CENTR(J3)*CENTR(J3)
   S5=S5+DATA(K3)*CENTR(J3)
   S6=S6+DATA(K3)*DATA(K3)
   S7=S7+DATA(K3)*DATA(K3)
   TP=S4-S1*S3/FLOAT(NV)
DIST=0.0
GO TO 7
K=1
NV=1
J=NV

1 DIST=1+(DATA(J2)-DATA(J1))*(DATA(J2)-DATA(J1))
DIST=SQR(DIST)
GO TO 2,1
5
NV=NV+1
NK=NV
S=NV
K=S

10 I=1
K=1
J=NV
J=NV
J=J+1
GO TO 10
20 DIST=1+(DATA(J2)-DATA(J1))*(DATA(J2)-DATA(J1))
DIST=SQR(DIST)
GO TO 2
10 NV=NV+1
J=NV
J=NV
GO TO 2
20 DIST=1+(DATA(J2)-DATA(J1))*(DATA(J2)-DATA(J1))
DIST=SQR(DIST)
GO TO 2
20

C PROGRAM RAND AND SINGLE PARTITION CRITERION STATISTICS
C
C SUM=0.0
SUM=0.0
SUM=0.0
C LEVEL=
105 LEVEL=0
LEVEL=0
LEVEL=0
106 LEVEL=LEVEL+1
LEVEL=LEVEL+1
LEVEL=LEVEL+1
107 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
108 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
109 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
110 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
111 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
112 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
113 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
114 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
115 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
116 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
117 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
118 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
119 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
120 SUM=SUM+LEVEL
LEVEL=LEVEL
LEVEL=LEVEL
C PRINT AND PUNCH RESULTS AS APPROPRIATE
C
C PRINT 100, SET, TYPE, RAND(TYPE), PCORR(TYPE)
C IF TYPE=NC.11) GO TO 5
C GO 100, 101
100 PRINT 100, SET, TYPE, RAND(TYPE), PCORR(TYPE)
101 IF TYPE=NC.11) GO TO 5

C PROCEDURE (METHOD=14)
C
C THIS SUBROUTINE USES EITHER FORGY'S (METHOD=13) OR JANCEY'S
C
COMMON NUMBR5,MEMBM(7C),DATAm(7DC)
DIMENSION ASED5,SED5M,ITOT1(170),CENTR(50)
DATA CHNED5,KN=0,L=1/3x2.557,2.557,3.5456,0,0,7.3687
INTEGER TYPE

READ THE DATA SET

REAL X(1),Y(1)
RETURN

Establish Initial Partition

CALL RANDOM (1,SEED,ASEED,NC)
DO 10 IM=1,NC
  INTEGER IM,IF,NUMB(IM)=1,(ASEED,IM)*(NE-1)+1
  DO 10 IM=1,NC
    IF(NHM=IM,NC,NUMB(IM)) CONTINUE
10  CONTINUE

SET UP THE SEED POINTS - THE DATA UNIT WITH SEQUENCE NUMBER NUMB(J)
  IS USED AS THE J-TH SEED POINT

DO 10 J=1,NC
  REAL X(J),Y(J)
  INTEGER J,IV,IM,IV=1,J=1
  CONTINUE

Initialize Arrays

DO 10 J=1,NC
  REAL X(J)
  X(J)=1.0
10  CONTINUE

BEGINNINC OF MAIN LOOP

INTEGER J1,J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12
DO 11 J1=1,NC

ALLOCATE EACH DATA UNIT TO THE NEAREST CLUSTER CENTROID

REAL X(1),Y(1)
CONTINUE

compute Distance to First Cluster Centroid

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
CONTINUE

REAL X(1),Y(1)
DO J=1,NV
   DIST=1.0*(TOP-BOTTOM-1.0)
   GO TO 201
10 DIST=DIST+(DATA(K3)-CENTR(J3))*(DATA(K3)-CENTR(J3))
   DIST=1.0/(CHIMD(NV)*DIST)/(CHIMD(NV)*DIST)+1.0
201 JREF=J
C C TEST DISTANCES TO REMAINING CLUSTER CENTROIDS
   DO 40 J=2,NV
      JREF=J
      DIST=1.0
      GO TU (S-5.5,44,54,45,46,47)*TYPE
40   Du=J3=1,NV
      J3=J3+1
      JREF=J1
      DIST=DIST+(DATA(K3)-CENTR(J3))
      DIST=DIST+/DATA(K3)-CENTR(J3))
      GO TO 301
301 IF((JREF,E2,JREF)) GO TO 140
      IF(JREF,JUF,E,JREF)) GO TO 140
      JREF=J
      GO TO 400
400 CONTINUE
C C ALLOCATE DATA UNIT K TO CLUSTER JREF
   NUM=NUM(JREF)*NUM(JREF)+1
   DIST=DIST+DIST+KREF
   IF(JREF,E2,MEMO(K)) GO TO 150
C C THE DATA UNIT CHANGES ITS MEMBERSHIP
500 NUM=NUM+1
   NUM(K)=JREF
J1=JREF+1
   NUM(J1)=NUM
   NUM(J1)=1
   TOTAL(J1)=TOTAL(J1)+DATA(K1)
   CONTINUE
J50 ALL DATA UNITS ALLOCATED, TEST FOR CONVERGENCE
   NPASS=NPASS+1
   JREF=J0
   IF(MOVES,GE,0) GO TO 185
   IF(BRU<.0,14) RETURN
   JREF=1
C C COMPUTE TRUE CLUSTER CENTROIDS - FORGIVE UPDATE
170 J1=0
   Du=J1,J=1,NV
   Du=J1,J=1,NV
   J1=J1+1
100 IF NUMER(J) < 10 GO TO 21
105 CENTER(J) = TOTAL(J) / NUMER(J)
110 IF J = 1 THEN RETURN
115 GO TO 100
120 IF METHOD = NE 1 THEN GO TO 170
130 JACOBY UPDATE:
140 J1 = J
145 DU = JC(J) / NC
150 IN = XCV IN + YV
155 J = J + 1
160 IF NUMER(J) < 0 THEN GO TO 21
165 CENTER(J) = 2 * TOTAL(J) / NUMER(J) - CENTER(J)
170 GO TO 120
180 FORMAT (OFF = 3)
END
**NON-HIERARCHICAL CLUSTERING ALGORITHMS - K-MEANS PROCEDURES**

PROGRAMMED BY GLORIA W. MILLIGAN

**DOCTORAL DISSERTATION - FEBRUARY 1976**

**INPUT FROM FILE 1 (PTI1IN1) - PROGRAM READS DATA EXACTLY AS WRITTEN BY THE GENERATOR PROGRAM**

**PUNCH OUTPUT CONSISTS OF EXTERNAL AND INTERNAL CRITERION STATISTICS**

**THE METHOD CODE IS**

**3 - CONVERGENT K-MEANS PROCEDURE**

```
COMMON NUMBR(5),MEMBR1(70),DATA(701)
DIMENSION PCOR(11),RND(11),SPEC(2515),CHIMED(18),LIST(701)
DATA CHIMED(/12356,237,23735,2652,234820,23735/)
LOGICAL MD(70,70),WRAND(150,50)
INTEGER SET,TYPE,NUMNR,CLUSNR,DENSIT

CALL LRAND
ISCI=DS54356399
PRINT 1
PRINT 5
PRINT 3
METHOD=16
DO 5 SET=1,1104
KEW(1104) SET,TYPE,NE,NUMNR,CLUSNR,NC,DFYSIT,(MEMBR1(I),I=1,10)
NV=MIN(2,NV)
IF(TYPE.EQ.6) NV=NV+1
IF(TYPE.EQ.7) NV=NV+2
5 CONTINUE

SET UP ARRAYS FOR CRITERION STATISTICS

IF(TYPE.EQ.1) GO TO 15
JCLUSR=NL+W
J=J+1
DO 13 J=1,50
WRANDJ(J+1)=J
M=MEMBR(J)
W=MENR(1,J)
N=M+1
DO 13 I=1,N,M
J=J+1
WRANDJ(J)=J
CONTINUE
13 CONTINUE
DO 15 J=50,1
DO 15 J=1,50
WRAND(J)=J
15 CONTINUE

CALL THE KMEAN SUBROUTINE

CALL KMEAN (NE,NV,NC,METHOD,ISEED,TYPE)

UPDATE THE LOGICAL ARRAY

NE=NE-1
DO 20 J=1,NC
IF MEMBR(J)=NE GO TO 21
20 CONTINUE
K2=K+1
DO 30 K=2,NE
IF MEMBR(K)=NE GO TO 30
30 CONTINUE
40 CONTINUE

COMPUTE INTERPOINT DISTANCES

1VAC=0
DO 201 J=2,50
1I=I-1
DO 201 J=1,11
1VAC=1VAC+J
201 CONTINUE
```
DIST=0.0
GO TO 7,7,7,7,7,7,7,7,7,7,7), TYPE
7 DD J=K1, NV
J1=NV*(J-1)+1
J2=NV*(J-1)+K
1.0 DIST=DIS^T*(DATA(J2)-DATA(J1))*DATA(J2)-DATA(J1)
DIST=SQRT(DIST)
GO TO Z01
8 S=0.3
S3=0.3
S4=0.3
S0=0.3
DO 11 J=1,NV
J1=NV*(J-1)+1
J2=NV*(J-1)+K
S1=S*DATA(J1)
S2=S*DATA(J2)
S3=S3*DATA(J1)
S4=S4*DATA(J2)
11 S=S+S1+S2+S3+S4+DATA(J1)*DATA(J2)
TOP=S5=SI*5/FLOAT(NV)
BOTTOM=SUR1*S3*SI/FLOAT(NV))*S4-S3*SI/FLOAT(NV))
DIST=1.0*(TOP/BOTTOM-1.0)
GO TO 201
4 DD J=K1,NV
J=NV*(J-1)+1
J2=NV*(J-1)+K
1.0 DIST=DIS^T*(DATA(J2)-DATA(J1))*DATA(J2)-DATA(J1)
DIST=1.0*(CHI_MEDINV-DIST)/(CHI_MEDINV+DIST)-1.0)
201 EVEC(I,IEC)=DIS
C COMPUTE RAND AND SINGLE PARTITION CRITERION STATISTICS
C SUM=0.0
SUM1=0.0
SUM2=0.0
SUM3=0.0
SUM4=0.0
IVEC=0.0
DO 109 I=2,50
I=1
D/J=VEC(J)
IVEC=IVEC+1
SUM=SUM+IVEC(IVEC)
SUM1=SUM1+IVEC(IVEC)
109 SUM=SUM1+IVEC(IVEC)
IF (MOD(J,11)) GO TO 700
SUM=SUM1+1
GO TO 109
700 SUM=SUM1+1
C PRINT AND PUNCH RESULTS AS APPROPRIATE
C PRINT 1096, STR, RAND(TYPE), PCORR(TYPE)
IF (TYPE=6, 6, 11) GO TO 3
D/J=VEC(J)
MEMR(J)=INT(RAND(I)*1000.0)+5
LIST(J)=INT(PCORR(I)*1000.0)+2
PUNCH 1VEC(J),MEMR(J),LIST(J),M125
CONTINUE
C HDRI=HIERARCHICAL CLUSTERING ALGORITHMS
C FORMAT(*SET TYPE RAND INTERNAL.1.*/)
C FORMAT(*CONVERGENT K-MEANS PROCEDURE*)
C FORMAT(*CONVERGENT KMEAN PROCEDURE METHOD=15)
C COMMON NUMR(5),MEMR(70),DATA(700)
DIMENSION ASEED(5),CHIME(8),TOTAL(70),CENTR(50)
DATA CHIMED/0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0/ 
INTEGER TYPE

ESTABLISH INITIAL PARTITION

CALL RANDOM (ISEED,ASeed,NC)
DO 10 IM=1,NC
10 NUMBR(IM)=INTASEED(IM)*((NE-1)+1)
DO 20 IM=2,NC
IM=IM-1
DO 20 IM=1,IM
IF(NUMBR(IM).EQ.NUMBR(IM2)) GO TO 21
CONTINUE

READ THE DATA SET

K1=1
DO 40 K=1,NE
KM=K*NV-1
READ (1,101) (DATA(KM),KMM)
K1=K+NV

SET UP THE SEED POINT - THE DATA UNIT WITH SEQUENCE NUMBER NUMBR(J) IS USED AS THE J-TH SEED POINT

DO 30 J=1,NC
NJ=(NUMBR(J)-1)*NV
JJ=(J-1)*NV
DL 30 J=1,NC
CONTINUE

CONSTRUCT AN INITIAL PARTITION

DO 52 K=1,NE
MEMB(K)=0
J1=1
DO 53 J=1,NC
NUMBR(J)=0
DO 54 J=1,NC
J1=J+1
TOTAL(J1)=U
53 CONTINUE

ALLOCATE EACH DATA UNIT TO THE NEAREST SEED POINT

K1=0
DO 65 K=1,NE
K2=K1+1
J2=1

COMPUTE DISTANCE TO FIRST SEED POINT

DIST=0.0
GO TO 100
100 DIST=(DATAK3-CENTR(J3))*C(K3-CENTR(J3))
DIST=SQRT(DIST)
GOTO 201
10
DIST=0.0
DO 11 K=1,NC
K3=K2-1
J3=J2-1
11 CONTINUE

DIST=0.0
DO 12 K=1,NC
K3=K2-1
J3=J2-1
12 CONTINUE

DIST = 1.0 * ((CHIMED(INV) - DIST) / (CHIMED(INV) + DIST) - 1.0)
DREF = DIST
JREF = 1

C TEST DISTANCES TO REMAINING SEED POINTS

DO 5 J = 2, NC
   JREF = J
   S = DIST
   K = 1
   DO 4 K = 1, NV
      S = S + DATA(K) - CENTR(J)
   4 CONTINUE
   DIST = S / NV
   GO TO 301
5  CONTINUE

C

C COMPUTE THM CENTROIDS

C

C INITIALIZE ARRAYS

C

C ALLOCATE EACH DATA UNIT TO THE NEAREST CLUSTER CENTROID

C

C COMPUTE DISTANCE TO FIRST CLUSTER CENTROID
DIST=0.0
GO TO 17
END=1+NV
K=K+1
J=J+1
DIST=DIST+DATA(K3)*DATA(K3)*CENTR(J3)*CENTR(J1)
GO TO 43
K=K+1
J=J+1
DIST=DIST+DATA(K3)*DATA(K3)*CENTR(J3)*CENTR(J1)
TOP=5*SIN(S3/2)*FLOAT(NV)
BOTTOM=5*SIN(S3/2)*FLOAT(NV)
DIST=1.0*(TOP/BOTTOM)-1.0
GO TO 91
K=K+1
J=J+1
DIST=DIST+DATA(K3)*DATA(K3)*CENTR(J3)*CENTR(J1)
DIST=1.0*(CHI(NEW)-DIST)/CHI(NEW)+DIST)-1.0
GO TO 91
DIST=DIST+DATA(K3)*DATA(K3)*CENTR(J3)*CENTR(J1)
GO TO 501
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STO
NUMBR(JREF) = NUMBR(JREF) + 1
JREF(JREF) = JREF
J1 = JREF + 1
J = J(J1-J) + 1
J1 = J1 + 1
K = J1 + 1
TOTAL(J1) = TOTAL(J1) + DATA(K1)
CENTR(J1) = TOTAL(J1) / NUMBR(JREF)
TOTAL(J3) = TOTAL(J3) + DATA(K3)
CENTR(J3) = TOTAL(J3) / NUMBR(J3)
CONTINUE
ALL DATA UNITS ALLOCATED - TEST FOR CONVERGENCE
NPASS = NPASS + 1
IF(MV = 5.0) RETURN
END
BIBLIOGRAPHY

Aldenderfer, M. S. A consumer report on cluster analysis software: (2) Hierarchical methods (NSF grant DCR #74-20007). State College, Pa.: The Pennsylvania State University, Department of Anthropology, July 1977.


Baron, D. N., & Fraser, P. M. Medical applications of taxonomic techniques. British Medical Bulletin, 1968, 24, 236-240.


200


Blashfield, R. K. A consumer report on cluster analysis software: (3) Iterative partitioning methods (NSF grant DCR #74-20007). State College, Pa.: The Pennsylvania State University, Department of Psychology, March 1977. (a)


Dudewicz, E. J.  IRCCRAND - The Ohio State University random number generator package (Tech. Rep. No. 104). Columbus, Ohio: The Ohio State University, Department of Statistics, 1974.


Milligan, G. W. Problems associated with the use of the Baker and Hubert cluster statistic. Manuscript submitted for publication, 1978. (a)

Milligan, G. W. Ultrametric clustering algorithms. Manuscript submitted for publication, 1978. (b)


Sorensen, T.  A Method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons.  *Biological Skrifter*, 1948, 5, 1-34.


Tryon, R. C. *Cluster analysis: Correlation profile and orthometric analysis for the isolation of unities in mind and personality*. Ann Arbor: Edward Brothers, 1939.


Wolfe, J. H. *NORMIX 360 Computer program (SRM-72-4).* San Diego, California: U.S. Naval Personnel and Training Research Laboratory, 1971. (NTIS No. AD 731 037)