Chapter
4

# Cluster Analysis

*Leland Wilkinson, Laszlo Engelman, James Corter, and Mark Coward*
*(Revised by Siva Athreya, Mousum Dutta, and Goutam Peri)*

SYSTAT provides a variety of cluster analysis methods on rectangular or symmetric data matrices. Cluster analysis is a multivariate procedure for detecting natural groupings in data. It resembles discriminant analysis in one respect—the researcher seeks to classify a set of objects into subgroups although neither the number nor members of the subgroups are known.

CLUSTER provides three procedures for clustering: Hierarchical Clustering, K-Clustering, and Additive Trees. The Hierarchical Clustering procedure comprises hierarchical linkage methods. The K-Clustering procedure splits a set of objects into a selected number of groups by maximizing between-cluster variation and minimizing within-cluster variation. The Additive Trees Clustering procedure produces a Sattath-Tversky additive tree clustering.

Hierarchical Clustering clusters cases, variables, or both cases and variables simultaneously; K-Clustering clusters cases only; and Additive Trees clusters a similarity or dissimilarity matrix. Several distance metrics are available with Hierarchical Clustering and K-Clustering including metrics for binary, quantitative and frequency count data. Hierarchical Clustering has ten methods for linking clusters and displays the results as a tree (dendrogram) or a polar dendrogram. When the MATRIX option is used to cluster cases and variables, SYSTAT uses a gray-scale or color spectrum to represent the values.

SYSTAT further provides five indices, viz., statistical criteria by which an appropriate number of clusters can be chosen from the Hierarchical Tree. Options for cutting (or pruning) and coloring the hierarchical tree are also provided.

In the K-Clustering procedure SYSTAT offers two algorithms, KMEANS and KMEDIANS, for partitioning. Further, SYSTAT provides nine methods for selecting initial seeds for both KMEANS and KMEDIANS.

Resampling procedures are available only in Hierarchical Clustering.

# *Statistical Background*

Cluster analysis is a multivariate procedure for detecting groupings in data. The objects in these groups may be:

- **Cases** (observations or rows of a rectangular data file). For example, suppose health indicators (numbers of doctors, nurses, hospital beds, life expectancy, etc.) are recorded for countries (cases), then developed nations may form a subgroup or cluster separate from developing countries.
- **Variables** (characteristics or columns of the data). For example, suppose causes of death (cancer, cardiovascular, lung disease, diabetes, accidents, etc.) are recorded for each U.S. state (case); the results show that accidents are relatively independent of the illnesses.
- **Cases and variables** (individual entries in the data matrix). For example, certain wines are associated with good years of production. Other wines have other years that are better.

## *Types of Clustering*

Clusters may be of two sorts: overlapping or exclusive. Overlapping clusters allow the same object to appear in more than one cluster. Exclusive clusters do not. All of the methods implemented in SYSTAT are exclusive.

There are three approaches to producing exclusive clusters: hierarchical, partitioned, and additive trees. Hierarchical clusters consist of clusters that completely contain other clusters that in turn completely contain other clusters, and so on, until there is only one cluster. Partitioned clusters contain no other clusters. Additive trees use a graphical representation in which distances along branches reflect similarities among the objects.

The cluster literature is diverse and contains many descriptive synonyms: hierarchical clustering (McQuitty, 1960; Johnson, 1967); single linkage clustering (Sokal and Sneath, 1963), and joining (Hartigan, 1975). Output from hierarchical methods can be represented as a tree (Hartigan, 1975) or a dendrogram (Sokal and Sneath, 1963). Density estimates (Hartigan 1975; Wong and Lane, 1983) can be used for clustering. Silverman (1986) provides several methods for density estimation.

## *Correlations and Distances*

To produce clusters, we must be able to compute some measure of dissimilarity between objects. Similar objects should appear in the same cluster, and dissimilar objects, in different clusters. All of the methods available in CORR for producing matrices of association can be used in cluster analysis, but each has different implications for the clusters produced. Incidentally, CLUSTER converts correlations to dissimilarities by negating them.

In general, the correlation measures (Pearson, Mu2, Spearman, Gamma, Tau) are not influenced by differences in scales between objects. For example, correlations between states using health statistics will not in general be affected by some states having larger average numbers or variation in their numbers. Use correlations when you want to measure the similarity in patterns across profiles regardless of overall magnitude.

On the other hand, the other measures such as Euclidean and City (city-block distance) are significantly affected by differences in scale. For health data, two states will be judged to be different if they have differing overall incidences even when they follow a common pattern. Generally, you should use the distance measures when variables are measured on common scales.

### *Standardizing Data*

Before you compute a dissimilarity measure, you may need to standardize your data across the measured attributes. Standardizing puts measurements on a common scale. In general, standardizing makes overall level and variation comparable across measurements. Consider the following data:

| OBJECT | X1 | X2 | X3 | X4 |
|---|---|---|---|---|
| A | 10 | 2 | 11 | 900 |
| B | 11 | 3 | 15 | 895 |
| C | 13 | 4 | 12 | 760 |
| D | 14 | 1 | 13 | 874 |

If we are clustering the four cases (*A* through *D*), variable *X4* will determine almost entirely the dissimilarity between cases, whether we use correlations or distances. If we are clustering the four variables, whichever correlation measure we use will adjust for the larger mean and standard deviation on *X4*. Thus, we should probably

standardize within columns if we are clustering rows and use a correlation measure if we are clustering columns.

In the example below, case *A* will have a disproportionate influence if we are clustering columns.

| OBJECT | X1 | X2 | X3 | X4 |
|---|---|---|---|---|
| A | 410 | 311 | 613 | 514 |
| B | 1 | 3 | 2 | 4 |
| C | 10 | 11 | 12 | 10 |
| D | 12 | 13 | 13 | 11 |

We should probably standardize within rows before clustering columns. This requires transposing the data before standardization. If we are clustering rows, on the other hand, we should use a correlation measure to adjust for the larger mean and standard deviation of case *A*.

These are not immutable laws. The suggestions are only to make you realize that scales can influence distance and correlation measures.

## *Hierarchical Clustering*

In Hierarchical Clustering, initially, each object (case or variable) is considered as a separate cluster. Then two 'closest' objects are joined as a cluster and this process is continued (in a stepwise manner) for joining an object with another object, an object with a cluster, or a cluster with another cluster until all objects are combined into one single cluster. This Hierarchical clustering is then displayed pictorially as a tree referred to as the Hierarchical tree.

The term 'closest' is identified by a specified rule in each of the Linkage methods. Hence in different linkage methods, the corresponding distance matrix (or dissimilarity measure) after each merger is computed by a different formula. These formulas are briefly explained below.

### *Linkage Methods*

SYSTAT provides the following linkage methods: Single, Complete, Average, Centroid, Median, Ward's (Ward, 1963), Weighted Average and Flexible Beta. As explained above, each method differs in how it measures the distance between two clusters and consequently it influences the interpretation of the word 'closest'. Initially,

the distance matrix gives the original distance between clusters as per the input data. The key is to compute the new distance matrix every time any two of the clusters are merged. This is illustrated via a recurrence relationship and a table.

Suppose $R$, $P$, $Q$ are existing clusters and $P+Q$ is the cluster formed by merging cluster $P$ and cluster $Q$, and $n_X$ is the number of objects in the Cluster X. The distance between the two clusters $R$ and $P+Q$ is calculated by the following relationship:

$$d(R,P+Q) = w_1 d(R, P) + w_2 d(R,Q) + w_3 d(P,Q) + w_4 |d(R, P) - d(R,Q)|$$

where the weights $w_1$, $w_2$, $w_3$, $w_4$ are method specific, provided by the table below:

| Name | $w_1$ | $w_2$ | $w_3$ | $w_4$ |
|---|---|---|---|---|
| Single | 1/2 | 1/2 | 0 | -1/2 |
| Complete | 1/2 | 1/2 | 0 | 1/2 |
| Average | $n_P/(n_P+n_Q)$ | $n_Q/(n_P+n_Q)$ | 0 | 0 |
| Weighted | 1/2 | 1/2 | 0 | 0 |
| Centroid | $n_P/(n_P+n_Q)$ | $n_Q/(n_P+n_Q)$ | $-(n_P n_Q)/(n_P+n_Q)^2$ | 0 |
| Median | 1/2 | 1/2 | -1/4 | 0 |
| Ward | $(n_R+n_P)/(n_R+n_P+n_Q)$ | $(n_R+n_Q)/(n_P+n_P+n_Q)$ | $n_R/(n_R+n_P+n_Q)$ | 0 |
| Felxibeta | $(1-\beta)/2$ | $(1-\beta)/2$ | $\beta$ | 0 |

From the above table it can be easily inferred that in a single linkage the distance between two clusters is the minimum of the distance between all the objects in the two clusters. Once the distances between the clusters are computed, the closest two are merged. The other methods can be suitably interpreted as well. Further descriptive details of the methods are given in the dialog-box description section.

### *Density Linkage Method*

SYSTAT provides two density linkage methods: the Uniform Kernel method and the $k^{th}$ Nearest Neighborhood method. In these methods a probability density estimate on the cases is obtained. Using this and the given dissimilarity matrix, a new dissimilarity matrix is constructed. Finally the single linkage cluster analysis is performed on the cases using the new dissimilarity measure.

For the uniform kernel method, you provide a value for the radius $r$. Using this, the density at a case $x$ is estimated as the proportion of the cases in the sphere of radius $r$, centered at $x$. In the $k^{th}$ nearest neighborhood method, you provide the value of $k$ upon which SYSTAT estimates the density at a case $x$ as the proportion of cases in the sphere centered at $x$ and the radius given by the distance to $k^{th}$ nearest neighbor of $x$. In each

of the above methods, the new dissimilarity measure between two cases is given by the average of the reciprocal of the density values of the two cases if they both lie within the same sphere of reference; otherwise, they are deemed to be infinite.

To understand the cluster displays of hierarchical clustering, it is best to look at an example. The following data reflect various attributes of selected performance cars.

| ACCEL | BRAKE | SLALOM | MPG | SPEED | NAME$ |
|---|---|---|---|---|---|
| 5.0 | 245 | 61.3 | 17.0 | 153 | Porsche 911T |
| 5.3 | 242 | 61.9 | 12.0 | 181 | Testarossa |
| 5.8 | 243 | 62.6 | 19.0 | 154 | Corvette |
| 7.0 | 267 | 57.8 | 14.5 | 145 | Mercedes 560 |
| 7.6 | 271 | 59.8 | 21.0 | 124 | Saab 9000 |
| 7.9 | 259 | 61.7 | 19.0 | 130 | Toyota Supra |
| 8.5 | 263 | 59.9 | 17.5 | 131 | BMW 635 |
| 8.7 | 287 | 64.2 | 35.0 | 115 | Civic CRX |
| 9.3 | 258 | 64.1 | 24.5 | 129 | Acura Legend |
| 10.8 | 287 | 60.8 | 25.0 | 100 | VW Fox GL |
| 13.0 | 253 | 62.3 | 27.0 | 95 | Chevy Nova |

### *Cluster Displays*

SYSTAT displays the output of hierarchical clustering in several ways. For joining rows or columns, SYSTAT prints a tree. For matrix joining, it prints a shaded matrix.

**Trees.** A tree is printed with a unique ordering in which every branch is lined up such that the most similar objects are closest to each other. If a perfect seriation (one-dimensional ordering) exists in the data, the tree reproduces it. The algorithm for ordering the tree is given in Gruvaeus and Wainer (1972). This ordering may differ from that of trees printed by other clustering programs if they do not use a seriation algorithm to determine how to order branches. The advantage of using seriation is most apparent for single linkage clustering.

If you join rows, the end branches of the tree are labeled with case numbers or labels. If you join columns, the end branches of the tree are labeled with variable names.

**Direct display of a matrix.** As an alternative to trees, SYSTAT can produce a shaded display of the original data matrix in which rows and columns are permuted according to an algorithm in Gruvaeus and Wainer (1972). Different characters represent the

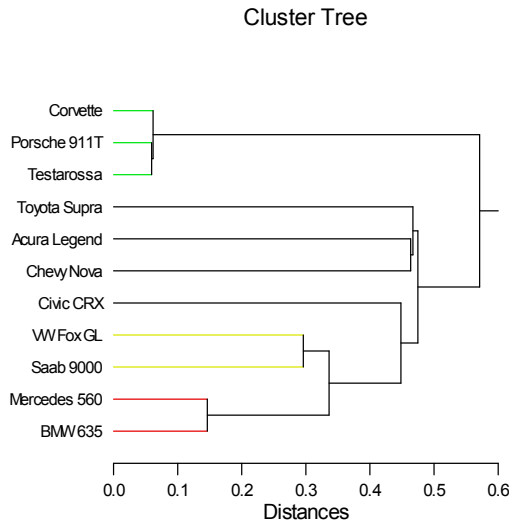
magnitude of each number in the matrix (see Ling, 1973). A legend showing the range of data values that these characters represent appears with the display.

Cutpoints between these values and their associated characters are selected to heighten contrast in the display. The method for increasing contrast is derived from techniques used in computer pattern recognition, in which gray-scale histograms for visual displays are modified to heighten contrast and enhance pattern detection. To find these cutpoints, we sort the data and look for the largest gaps between adjacent values. Tukey's gapping method (See Wainer and Schacht, 1978) is used to determine how many gaps (and associated characters) should be chosen to heighten contrast for a given set of data. This procedure, time consuming for large matrices, is described in detail in Wilkinson (1979).

If you have a course to grade and are looking for a way to find rational cutpoints in the grade distribution, you might want to use this display to choose the cutpoints. Cluster the $n \times 1$ matrix of numeric grades (*n* students by 1 grade) and let SYSTAT choose the cutpoints. Only cutpoints asymptotically significant at the 0.05 level are chosen. If no cutpoints are chosen in the display, give everyone an *A*, flunk them all, or hand out numeric grades (unless you teach at Brown University or Hampshire College).

### *Clustering Rows*

First, let us look at possible clusters of the cars in the example. Since the variables are on such different scales, we will standardize them before doing the clustering. This will give acceleration comparable influence to braking. Then we select Pearson correlations as the basis for dissimilarity between cars. The result is:



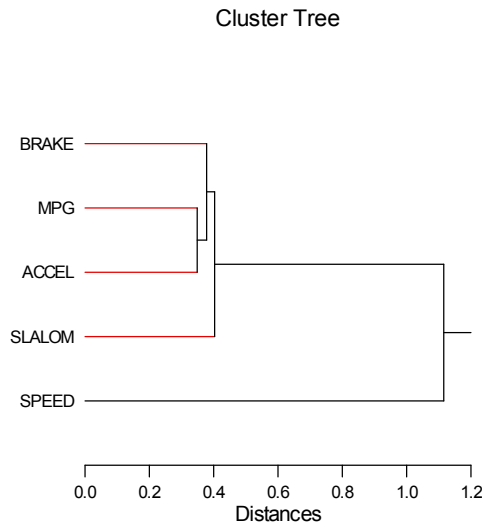

If you look at the correlation matrix for the cars, you will see how these clusters hang together. Cars within the same cluster (for example, Corvette, Testarossa, Porsche) generally correlate highly.

| | **Porsche** | **Testa** | **Corv** | **Merc** | **Saab** |
|---|---|---|---|---|---|
| Porsche | 1.00 | | | | |
| Testa | 0.94 | 1.00 | | | |
| Corv | 0.94 | 0.87 | 1.00 | | |
| Merc | 0.09 | 0.21 | –0.24 | 1.00 | |
| Saab | –0.51 | –0.52 | –0.76 | 0.66 | 1.00 |
| Toyota | 0.24 | 0.43 | 0.40 | –0.38 | –0.68 |
| BMW | –0.32 | –0.10 | –0.56 | 0.85 | 0.63 |
| Civic | –0.50 | –0.73 | –0.39 | –0.52 | 0.26 |
| Acura | –0.05 | –0.10 | 0.30 | –0.98 | –0.77 |
| VW | –0.96 | –0.93 | –0.98 | 0.08 | 0.70 |
| Chevy | –0.73 | –0.70 | –0.49 | –0.53 | –0.13 |

| | **Toyota** | **BMW** | **Civic** | **Acura** | **VW** |
|---|---|---|---|---|---|
| Toyota | 1.00 | | | | |
| BMW | –0.25 | 1.00 | | | |
| Civic | –0.30 | –0.50 | 1.00 | | |
| Acura | 0.53 | –0.79 | 0.35 | 1.00 | |
| VW | –0.35 | 0.39 | 0.55 | –0.16 | 1.00 |
| Chevy | –0.03 | –0.06 | 0.32 | 0.54 | 0.53 |

### *Clustering Columns*

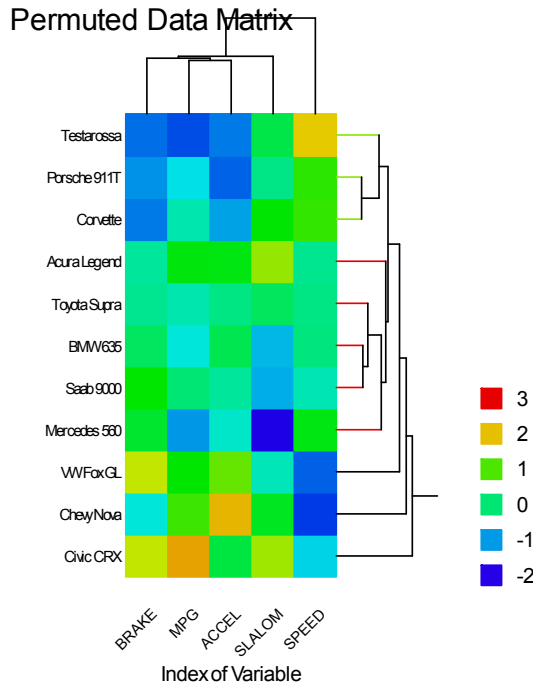
We can cluster the performance attributes of the cars more easily. Here, we do not need to standardize within cars (by rows) because all of the values are comparable between cars. Again, to give each variable comparable influence, we will use Pearson correlations as the basis for the dissimilarities. The result based on the data standardized by variable (column) is:




### *Clustering Rows and Columns*

To cluster the rows and columns jointly, we should first standardize the variables to give each of them comparable influence on the clustering of cars. Once we have standardized the variables, we can use Euclidean distances because the scales are comparable.

Single linkage is used to produce the following result:




This figure displays the standardized data matrix itself with rows and columns permuted to reveal clustering and each data value replaced by one of three symbols. Note that the rows are ordered according to overall performance, with the fastest cars at the top.

Matrix clustering is especially useful for displaying large correlation matrices. You may want to cluster the correlation matrix this way and then use the ordering to produce a scatterplot matrix that is organized by the multivariate structure.

### *Cluster Validity Indices*

The fundamental aim of the cluster validity indices is to enable the user to choose an optimal number of clusters in the data subject to pre-defined conditions. Milligan and Cooper (1985) studied several such indices. In this section we discuss five indices that are provided by SYSTAT for Hierarchical clustering.

**Root Mean Square Standard Deviation (RMSSTD) Index**. This index is the root mean square standard deviation of all the variables within each cluster. This is calculated by

calculating the within-group sum of squares of each cluster and normalizing it by the product of the number of elements in the cluster and the number of variables (Sharma, 1995). More precisely,

$$\text{RMSSTD} = \sqrt{W_k / (v(N_k - 1))}$$

where $W_k$ is the within-group sum of squares of cluster *k*, $N_k$ is the number of elements in cluster *k* and *v* is the number of variables. SYSTAT calculates the index at each step of the Hierarchical algorithm providing a measure of homogeneity of the clusters that have been formed. Thus, the smaller the value of the RMSSTD, the better is the cluster formed. At any hierarchical step, if the RMSSTD value rises then the new clustering scheme is worse.

SYSTAT provides a plot of RMSSTD for a number of steps in the hierarchical clustering. You can then determine the number of clusters that exist in a data set by spotting the 'knee' (in other words, the steep jump of the index value from higher to smaller numbers of clusters) in the graph. This index is valid for rectangular data. If a dissimilarity matrix is available, then the index is valid only if the methods used are average, centroid or Ward.

**Dunn's Index.** This cluster validity index was proposed by Dunn (1973). Suppose the number of clusters at a given level in the hierarchical cluster tree is *k*. For any two clusters $X_i$ and $X_j$ let $\delta(X_i, X_j)$ be the distance between two clusters and $\Delta(X_i)$ be the diameter of cluster $X_i$. Dunn's index is defined as the minimum of the ratio of the dissimilarity measure between two clusters to the diameter of cluster, where the minimum is taken over all the clusters in the data set. More precisely,

$$\text{Dunn's Index} = \min_{1 \le i \le k} \left\{ \min_{1 \le j \ne i \le k} \frac{\delta(X_i, X_j)}{\max_{1 \le i \le k} \Delta(X_i)} \right\}$$

Originally, the distance between two sets is defined as the minimum distance between two points taken from different sets, whereas the diameter of a set is defined as the maximum distance between two points in the set. A generalization of the above measurement can be found in Bezdek and Pal (1998). If the data set contains close-knit but separated clusters, the distance between the clusters is expected to be large and the diameter of the clusters is expected to be small. So, based on the definition, large values of the index indicate the presence of compact and well-separated clusters. Thus, the clustering which attains the maximum in the plot of Dunn's versus the number of clusters, is the appropriate one. This index is valid for both rectangular and dissimilarity data.

**Davies-Bouldin's (DB's) Index**. Let *k* be the number of clusters at a given step in hierarchical clustering. Let $v_{x_i}$ denote the centre of the cluster $X_i$ and $|X_i|$ the size of the cluster $X_i$ .

Define $S_i = \left( \frac{1}{|X_i|} \sum_{x \in X_i} d^2(x, v_{x_i}) \right)^{\frac{1}{2}}$ as the measure of dispersion of cluster $X_i$ ,

$d_{ij} = d(v_{x_i}, v_{x_j})$, as the dissimilarity measure between clusters $X_i$ and $X_j$ and

$R_i = \text{Max}_{j,\ j \neq i} \left\{ \frac{S_i + S_j}{d_{ij}} \right\}$

Then the DB (Davies and Bouldin, 1979) Index is defined as DB's Index = $\frac{1}{k} \sum_{1}^{k} R_i$ .

It is clear that DB's index quantifies the average similarity between a cluster and its most similar counterpart. It is desirable for the clusters to be as distinct from each other as possible. So a clustering which minimizes the DB index is the ideal one. This index can be calculated for rectangular data.

**Pseudo *F* Index**. The pseudo *F* statistic describes the ratio of between-cluster variance to within cluster variance (Calinski and Harabasz, 1974):

$$\text{Pseudo } F = \frac{(GSS) / (K - 1)}{(WSS) / (N - K)}$$

where *N* is the number of observations, *K* is the number of clusters at any step in the hierarchical clustering, GSS is the between-group sum of squares, and WSS is the within group sum of squares. Large values of *Pseudo F* indicate close-knit and separated clusters. In particular, peaks in the pseudo *F* statistic are indicators of greater cluster separation. Typically, these are spotted in the plot of the index versus the number of clusters. This index is valid for rectangular data and for any Hierarchical clustering procedure. In the case of dissimilarity data, one can use this index for hierarchical clustering if the methods used are average, centroid or Ward.

**Pseudo *T*-square Index**. Suppose, during a step in the hierarchical clustering, cluster *K* and cluster *L* are merged to form a new cluster. Then, the pseudo *T*-square statistic for the clustering obtained is given by

$$\text{Pseudo } T\text{-square} = \frac{B_{KL}}{((W_K + W_L) / (N_K + N_L - 2))}$$

where $N_K$ and $N_L$ are the number of observations in clusters *k*and *l*, $W_K$ and $W_L$ are within cluster sum of squares of clusters *k* and *l*, and $B_{KL}$ is the between-cluster

sum of squares. This index quantifies the difference between two clusters that are merged at a given step. Thus, if the pseudo *T*-square statistic has a distinct jump at step *k* of the hierarchical clustering, then the clustering in step *k+1* is selected as the optimal cluster. The pseudo *T*-square statistic is closely related to Duda and Hart's (Je(2)/ Je(1)) index.

## *Partitioning via K-Clustering*

To produce partitioned clusters, you must decide in advance how many clusters you want. *K*-Clustering searches for the best way to divide your objects into *K* different sections so that they are separated as well as possible. *K*-Clustering provides two such procedures: *K*-Means and *K*-Medians.

### *K-Means*

*K*-Means, which is the default procedure, begins by picking 'seed' cases, one for each cluster, which are spread apart as much as possible from the centre of all the cases. Then it assigns all cases to the nearest seed. Next, it attempts to reassign each case to a different cluster in order to reduce the within-groups sum of squares. This continues until the within-groups sum of squares can no longer be reduced. The initial seeds can be chosen from nine possible options.

*K*-Means does not search through every possible partitioning of the data, so it is possible that some other solution might have a smaller within-groups sum of squares. Nevertheless, it has performed relatively well on global data separated in several dimensions in Monte Carlo studies of cluster algorithms.

Because it focuses on reducing the within-groups sum of squares, *K*-Means clustering is like a multivariate analysis of variance in which the groups are not known in advance. The output includes analysis of variance statistics, although you should be cautious in interpreting them. Remember, the program is looking for large *F-ratios* in the first place, so you should not be too impressed by large values.

The following is a three-group analysis of the car data. The clusters are similar to those we found by joining. *K*-Means clustering uses Euclidean distances instead of Pearson correlations, so there are minor differences because of scaling.

To keep the influences of all variables comparable, we standardized the data before running the analysis.

**Distance Metric is Euclidean Distance**
**K-Means splitting cases into 3 groups**

**Summary Statistics for All Cases**

| Variable | Between SS | df | Within SS | df | F-ratio |
|---|---|---|---|---|---|
| ACCEL | 7.825 | 2 | 2.175 | 8 | 14.389 |
| BRAKE | 5.657 | 2 | 4.343 | 8 | 5.211 |
| SLALOM | 5.427 | 2 | 4.573 | 8 | 4.747 |
| MPG | 7.148 | 2 | 2.852 | 8 | 10.027 |
| SPEED | 7.677 | 2 | 2.323 | 8 | 13.220 |
| ** TOTAL ** | 33.735 | 10 | 16.265 | 40 | |

**Cluster 1 of 3 Contains 4 Cases**

| Members | | Statistics | | | | |
|---|---|---|---|---|---|---|
| Case | Distance | Variable | Minimum | Mean | Maximum | Standard Deviation |
| Mercedes 560 | 0.596 | ACCEL | -0.451 | -0.138 | 0.174 | 0.260 |
| Saab 9000 | 0.309 | BRAKE | -0.149 | 0.230 | 0.608 | 0.326 |
| Toyota Supra | 0.488 | SLALOM | -1.952 | -0.894 | 0.111 | 0.843 |
| BMW 635 | 0.159 | MPG | -1.010 | -0.470 | -0.007 | 0.423 |
| | | SPEED | -0.338 | 0.002 | 0.502 | 0.355 |

**Cluster 2 of 3 Contains 4 Cases**

| Members | | Statistics | | | | |
|---|---|---|---|---|---|---|
| Case | Distance | Variable | Minimum | Mean | Maximum | Standard Deviation |
| Civic CRX | 0.811 | ACCEL | 0.258 | 0.988 | 2.051 | 0.799 |
| Acura Legend | 0.668 | BRAKE | -0.528 | 0.624 | 1.619 | 1.155 |
| VW Fox GL | 0.712 | SLALOM | -0.365 | 0.719 | 1.432 | 0.857 |
| Chevy Nova | 0.763 | MPG | 0.533 | 1.054 | 2.154 | 0.752 |
| | | SPEED | -1.498 | -0.908 | -0.138 | 0.616 |

**Cluster 3 of 3 Contains 3 Cases**

| Members | | Statistics | | | | |
|---|---|---|---|---|---|---|
| Case | Distance | Variable | Minimum | Mean | Maximum | Standard Deviation |
| Porsche 911T | 0.253 | ACCEL | -1.285 | -1.132 | -0.952 | 0.169 |
| Testarossa | 0.431 | BRAKE | -1.223 | -1.138 | -1.033 | 0.096 |
| Corvette | 0.314 | SLALOM | -0.101 | 0.234 | 0.586 | 0.344 |
| | | MPG | -1.396 | -0.779 | -0.316 | 0.557 |
| | | SPEED | 0.822 | 1.208 | 1.941 | 0.635 |

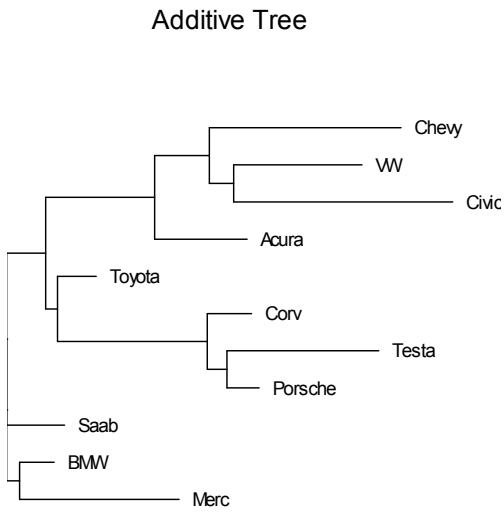
## *K-Medians*

The second approach available in *K*-Clustering is *K*-Medians. The *K*-Medians procedure follows the same amalgamation approach as *K*-Means except for a key difference. It uses the median to reassign each case to a different cluster in order to reduce the within-groups sum of absolute deviations.

## *Additive Trees*

Sattath and Tversky (1977) developed additive trees for modeling similarity/dissimilarity data. Hierarchical clustering methods require objects in the same cluster to have identical distances to each other. Moreover, these distances must be smaller than the distances between clusters. These restrictions prove problematic for similarity data, and, as a result, hierarchical clustering cannot fit this data set well.

In contrast, additive trees use the tree branch length to represent distances between objects. Allowing the within-cluster distances to vary yields a tree diagram with varying branch lengths. Objects within a cluster can be compared by focusing on the horizontal distance along the branches connecting them. The additive tree for the car data is as follows:



The distances between nodes of the graph are:

| Node | Length | Child |
|---|---|---|
| 1 | 0.10 | Porsche |
| 2 | 0.49 | Testa |
| 3 | 0.14 | Corv |
| 4 | 0.52 | Merc |
| 5 | 0.19 | Saab |
| 6 | 0.13 | Toyota |
| 7 | 0.11 | BMW |
| 8 | 0.71 | Civic |
| 9 | 0.30 | Acura |
| 10 | 0.42 | VW |
| 11 | 0.62 | Chevy |
| 12 | 0.06 | 1,2 |
| 13 | 0.08 | 8,10 |
| 14 | 0.49 | 12,3 |
| 15 | 0.18 | 13,11 |
| 16 | 0.35 | 9,15 |
| 17 | 0.04 | 14,6 |
| 18 | 0.13 | 17,16 |
| 19 | 0.0 | 5,18 |
| 20 | 0.04 | 4,7 |
| 21 | 0.0 | 20,19 |

Each object is a node in the graph. In this example, the first 11 nodes represent the cars. Other graph nodes correspond to "groupings" of the objects. Here, the 12th node represents *Porsche* and *Testa*.

The distance between any two nodes is the sum of the (horizontal) lengths between them. The distance between *Chevy* and *VW* is $0.62 + 0.08 + 0.42 = 1.12$. The distance between *Chevy* and *Civic* is $0.62 + 0.08 + 0.71 = 1.41$. Consequently, *Chevy* is more similar to *VW* than to *Civic*.

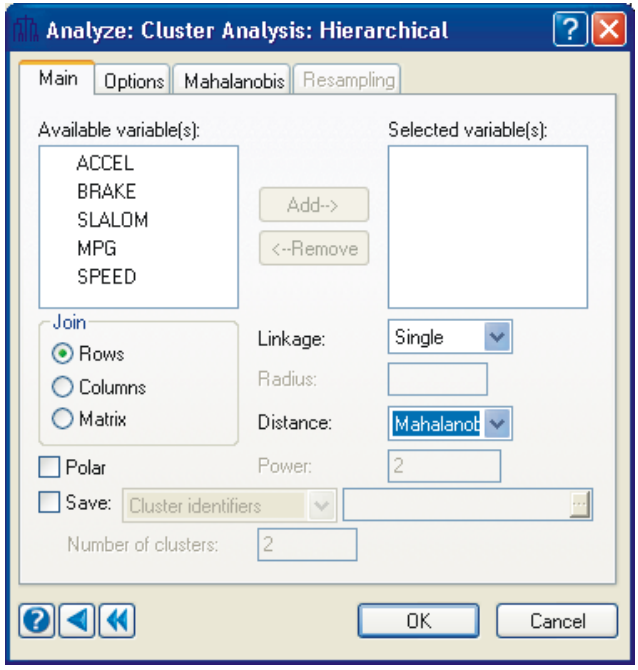
# *Cluster Analysis in SYSTAT*

## *Hierarchical Clustering Dialog Box*

Hierarchical clustering produces hierarchical clusters that are displayed in a tree. Initially, each object (case or variable) is considered a separate cluster. SYSTAT begins by joining the two "closest" objects as a cluster and continues (in a stepwise manner) joining an object with another object, an object with a cluster, or a cluster with another cluster until all objects are combined into one cluster.

To open the Hierarchical Clustering dialog box, from the menus choose:

Analyze
  Cluster Analysis
    Hierarchical…




You must select the elements of the data file to cluster (Join):

- **Rows.** Rows (cases) of the data matrix are clustered.

- **Columns.** Columns (variables) of the data matrix are clustered.
- **Matrix.** In Matrix, rows and columns of the data matrix are clustered—they are permuted to bring similar rows and columns next to one another.

Linkage allows you to specify the type of joining algorithm used to amalgamate clusters (that is, define how distances between clusters are measured).

- **Average.** Average linkage averages all distances between pairs of objects in different clusters to decide how far apart they are.
- **Centroid.** Centroid linkage uses the average value of all objects in a cluster (the cluster centroid) as the reference point for distances to other objects or clusters.
- **Complete.** Complete linkage uses the most distant pair of objects in two clusters to compute between-cluster distances.This method tends to produce compact, globular clusters. If you use a similarity or dissimilarity matrix from a SYSTAT file, you get Johnson's "max" method.
- **Flexibeta.** Flexible beta linkage uses a weighted average distance between pairs of objects in different clusters to decide how far apart they are.You can choose the value of the weight $\beta$ . The range of $\beta$ is between -1 and 1.
- ***K*-nbd.** $K^{th}$ nearest neighborhood method is a density linkage method. The estimated density is proportional to the number of cases in the smallest sphere containing the $k^{th}$ nearest neighbor. A new dissimilarity matrix is then constructed using the density estimate. Finally the single linkage cluster analysis is performed. You can specify the number *k*; its range is between 1 and the total number of cases in the data set.
- **Median.** Median linkage uses the median distances between pairs of objects in different clusters to decide how far apart they are.
- **Single.** Single linkage defines the distance between two objects or clusters as the distance between the two closest members of those clusters. This method tends to produce long, stringy clusters. If you use a SYSTAT file that contains a similarity or dissimilarity matrix, you get clustering via Johnson's "min" method.
- **Uniform.** Uniform Kernel method is a density linkage method. The estimated density is proportional to the number of cases in a sphere of radius *r*. A new dissimilarity matrix is then constructed using the density estimate. Finally, single linkage cluster analysis is performed. You can choose the number *r*; its range is the positive real line.
- **Ward.** Ward's method averages all distances between pairs of objects in different clusters, with adjustments for covariances, to decide how far apart the clusters are.

- **Weighted**. Weighted average linkage uses a weighted average distance between pairs of objects in different clusters to decide how far apart they are. The weights used are proportional to the size of the cluster.

For some data, some methods cannot produce a hierarchical tree with strictly increasing amalgamation distances. In these cases, you may see stray branches that do not connect to others. If this happens, you should consider Single or Complete linkage. For more information on these problems, see Fisher and Van Ness (1971). These reviewers concluded that these and other problems made Centroid, Average, Median, and Ward (as well as *K*-Means) "inadmissible" clustering procedures. In practice and in Monte Carlo simulations, however, they sometimes perform better than Single and Complete linkage, which Fisher and Van Ness considered "admissible." Milligan (1980) tested all of the hierarchical joining methods in a large Monte Carlo simulation of clustering algorithms. Consult his paper for further details.

In addition, the following options can be specified:

**Distance.** Specifies the distance metric used to compare clusters**.**

**Polar.** Produces a polar (circular) cluster tree.

**Save.** Save provides two options either to save cluster identifiers or to save cluster identifiers along with data. You can specify the number of clusters to identify for the saved file. If not specified, two clusters are identified.

### *Clustering Distances*

Both Hierarchical Clustering and *K*-Clustering allow you to select the type of distance metric to use between objects. From the Distance drop-down list, you can select:

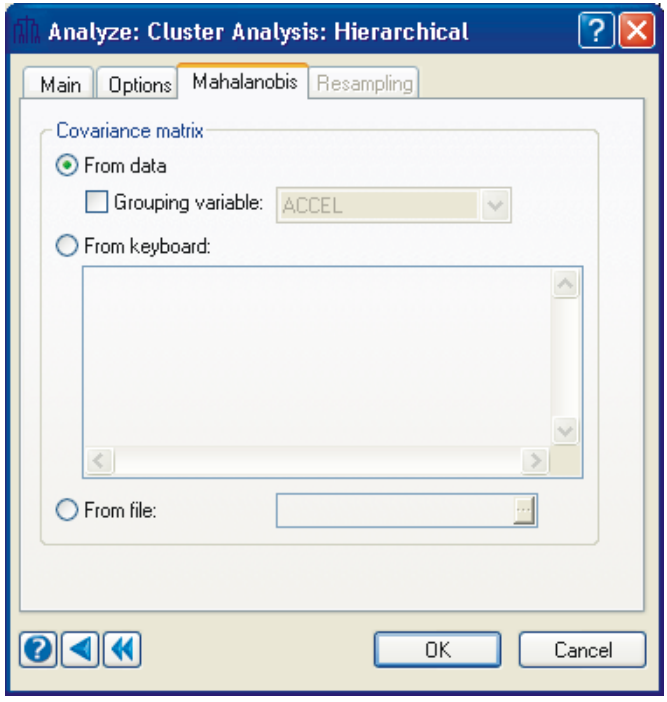
- **Absolute.** Distances are computed using absolute differences. Use this metric for quantitative variables. The computation excludes missing values.
- **Anderberg.** Distances are computed using a dissimilarity form of Anderberg's similarity coefficients for binary data. Anderberg distance is available for hierarchical clustering only.
- **Chi-square.** Distances are computed as the chi-square measure of independence of rows and columns on 2-by-*n* frequency tables, formed by pairs of cases (or variables). Use this metric when the data are counts of objects or events.
- **Euclidean.** Clustering is computed using normalized Euclidean distance (root mean squared distances). Use this metric with quantitative variables. Missing values are excluded from computations.

- **Gamma.** Distances are computed using one minus the Goodman-Kruskal gamma correlation coefficient. Use this metric with rank order or ordinal scales. Missing values are excluded from computations.
- **Jaccard.** Clustering is computed using the dissimilarity form of Jaccard's similarity coefficient for binary data. Jaccard distance is only available for hierarchical clustering.
- **Mahalanobis.** Distances are computed using the square root of the quadratic form of the deviations among two random vectors using the inverse of their variance-covariance matrix. This metric can also be used to cluster groups. Use this metric with quantitative variables. Missing values are excluded from computations.
- **Minkowski.** Clustering is computed using the *pth* root of the mean *pth* powered distances of coordinates. Use this metric for quantitative variables. Missing values are excluded from computations. Use the Power text box to specify the value of *p*.
- **MW** (available for *K*-Clustering only)**.** Distances are computed as the increment in within sum of squares of deviations, if the case would belong to a cluster. The case is moved into the cluster that minimizes the within sum of squares of deviations. Use this metric with quantitative variables. Missing values are excluded from computations.
- **Pearson.** Distances are computed using one minus the Pearson product-moment correlation coefficient for each pair of objects. Use this metric for quantitative variables. Missing values are excluded from computations.
- **Percent** (available for hierarchical clustering only)**.** Clustering uses a distance metric that is the percentage of comparisons of values resulting in disagreements in two profiles. Use this metric with categorical or nominal scales.
- **Phi-square.** Distances are computed as the phi-square (chi-square/total) measure on 2-by-*n* frequency tables, formed by pairs of cases (or variables). Use this metric when the data are counts of objects or events.
- **Rsquared.** Distances are computed using one minus the square of the Pearson product-moment correlation coefficient for each pair of objects. Use this metric with quantitative variables. Missing values are excluded from computations.
- **RT.** Clustering uses the dissimilarity form of Rogers and Tanimoto's similarity coefficient for categorical data. *RT* distance is available only for hierarchical clustering.
- **Russel.** Clustering uses the dissimilarity form of Russel's similarity coefficient for binary data. Russel distance is available only for hierarchical clustering.

- **SS.** Clustering uses the dissimilarity form of Sneath and Sokal's similarity coefficient for categorical data. *SS* distance is available only for hierarchical clustering.

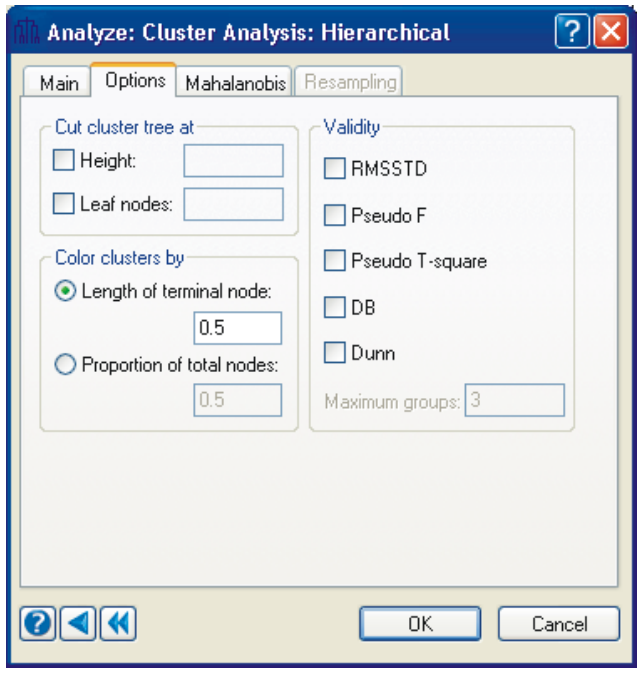
### *Mahalanobis*

In the Mahalanobis tab, you can specify the covariance matrix to compute Mahalanobis distance.




**Covariance matrix.** Specify the covariance matrix to compute the Mahalanobis distance. Enter the covariance matrix either through the keyboard or from a SYSTAT file. Otherwise, by default SYSTAT computes the matrix from the data. Select a grouping variable for inter-group distance measures.

### *Options*




The following options are available:

**Cut cluster tree at.** You can choose the following options for cutting the cluster tree:

- **Height.** Provides the option of cutting the cluster tree at a specified distance.
- **Leaf nodes.** Provides the option of cutting the cluster tree by number of leaf nodes.

**Color clusters by.** The colors in the cluster tree can be assigned by two different methods:

- **Length of terminal node.** As you pass from node to node in order down the cluster tree, the color changes when the length of a node on the distance scale changes between less than and greater than the specified length of terminal nodes (on a scale of 0 to 1).
- **Proportion of total nodes.** Colors are assigned based on the proportion of members in a cluster.

**Validity.** Provides five validity indices to evaluate the partition quality. In particular, it is used to find out the appropriate number of clusters for the given data set.

- **RMSSTD.** Provides root-mean-square standard deviation of the clusters at each step in hierarchical clustering.
- **Pseudo F.** Provides pseudo *F-ratio* for the clusters at each step in hierarchical clustering.
- **Pseudo T-square.** Provides pseudo T-square statistic for cluster assessment.
- **DB.** Provides Davies-Bouldin's index for each hierarchy of clustering. This index is applicable for rectangular data only.
- **Dunn.** Provides Dunn's cluster separation measure.
- **Maximum groups.** Performs the computation of indices up to this specified number of clusters. The default value is the square-root of number of objects.

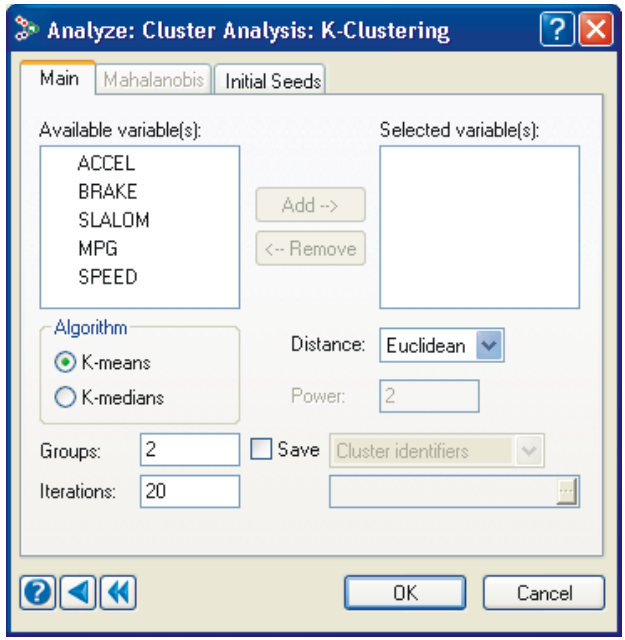
## *K-Clustering Dialog Box*

*K*-Clustering dialog box provides options for *K*-Means clustering and *K*-Medians clustering. Both clustering methods split a set of objects into a selected number of groups by maximizing between-cluster variation relative to within-cluster variation. It is similar to doing a one-way analysis of variance where the groups are unknown and the largest *F* value is sought by reassigning members to each group.

By default, the algorithms start with one cluster and splits it into two clusters by picking the case farthest from the center as a seed for a second cluster and assigning each case to the nearest center. It continues splitting one of the clusters into two (and reassigning cases) until a specified number of clusters are formed. The reassigning of cases continues until the within-groups sum of squares can no longer be reduced. The initial seeds or partitions can be chosen from a possible set of nine options.

To open the *K*-Clustering dialog box, from the menus choose:

Analyze
  Cluster Analysis
    K-Clustering…




**Algorithm**. Provides *K*-Means and *K*-Medians clustering options.

- ***K*-means.** Requests *K*-Means clustering.
- ***K*-medians.** Requests *K*-Medians clustering.

**Groups.** Enter the number of desired clusters. Default number (Groups) is two.

**Iterations.** Enter the maximum number of iterations. If not stated, the maximum is 20.

**Distance.** Specifies the distance metric used to compare clusters**.**

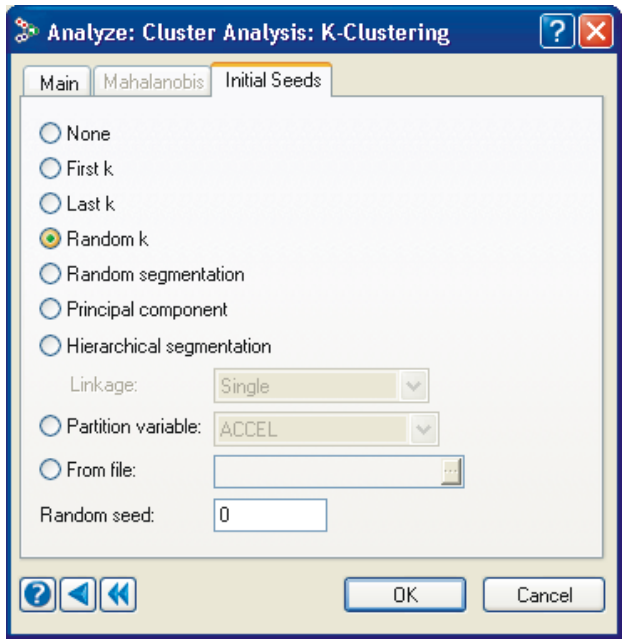
**Save.** Save provides three options to save either cluster identifiers, cluster identifiers along with data, or final cluster seeds, to a SYSTAT file.

***Mahalanobis.***

See the Mahalanobis tab in Hierarchical clustering.

***Initial Seeds.***

To specify the initial seeds for clustering, click on the Initial Seeds tab.




The following initial seeds options are available:

- **None.** Starts with one cluster and splits it into two clusters by picking the case farthest from the center as a seed for the second cluster and then assigning each case optimally.
- **First *K*.** Considers the first *K* non-missing cases as initial seeds.
- **Last *K*.** Considers the last *K* non-missing cases as initial seeds.
- **Random *K*.** Chooses randomly (without replacement) *K* non-missing cases as initial seeds.
- **Random segmentation.** Assigns each case to any of *K* partitions randomly. Computes seeds from each initial partition taking the mean or the median of the observations, whichever is applicable.

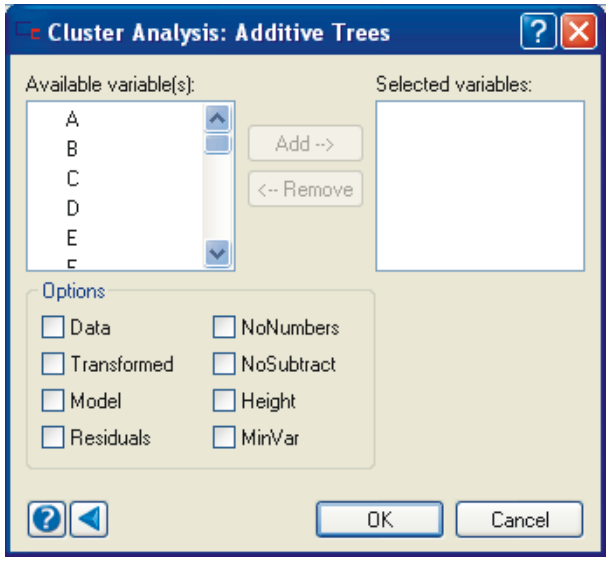
- **Principal component.** Uses the first principal component as a single variable. Sorts all cases based on this single variable. It creates partitions taking the first *n/K* cases in the first partition, the next *n/K* cases in the second partition and so on.
- **Hierarchical segmentation.** Makes the initial *K* partitions from hierarchical clustering with the specified linkage method.
- **Partition variable.** Makes initial partitions from a specified variable.
- **From file.** Specify the SYSTAT file where seeds are written in case by case.
- **Linkage.** Specify the linkage method for hierarchical segmentation.
- **Random seed.** Specify the seed for random number generation.

## *Additive Trees Clustering Dialog Box*

Additive trees were developed by Sattath and Tversky (1977) for modeling similarity/dissimilarity data, which hierarchical joining trees do not fit well. Hierarchical trees imply that all within-cluster distances are smaller than all between-cluster distances and that within-cluster distances are equal. This so-called "ultrametric" condition seldom applies to real similarity data from direct judgment. Additive trees, on the other hand, represent similarities with a network model in the shape of a tree. Distances between objects are represented by the lengths of the branches connecting them in the tree.

To open the Additive Trees Clustering dialog box, from the menus choose:

Analyze
  Cluster Analysis
    Additive Trees…




At least three variables should be selected to perform Additive Tree Clustering. The following options can be specified:

**Data.** Display the raw data matrix.

**Transformed.** Include the transformed data (distance-like measures) with the output.

**Model.** Display the model (tree) distances between the objects.

**Residuals.** Show the differences between the distance-transformed data and the model distances.

**NoNumbers.** Objects in the tree graph are not numbered.

**NoSubtract.** Use of an additive constant. Additive Trees assumes interval-scaled data, which implies complete freedom in choosing an additive constant, so it adds or subtracts to exactly satisfy the triangle inequality. Use this NoSubtract option to allow strict inequality and not subtract a constant.

**Height.** Prints the distance of each node from the root.

**MinVar.** Combines the last few remaining clusters into the root node by searching for the root that minimizes the variances of the distances from the root to the leaves.

## *Using Commands*

### *For the Hierarchical tree method:*

```
CLUSTER
  USE filename
  IDVAR var$
  SAVE filename / NUMBER=n DATA
  JOIN varlist / ROWS or COLUMNS or MATRIX POLAR DISTANCE=metric
                 POWER=p COV=matrix or 'filename' GROUP=var
                 LINKAGE=method RADIUS=r K=k BETA=b MAX=n
                 VALIDITY= RMSSTD, CHF, PTS, DB, DUNN, HEIGHT=r,
                 LEAF=n, LENGTH=r PROP=r
                 SAMPLE = BOOT(m,n) or SIMPLE(m,n) or JACK
```

The distance *metric* is ABSOLUTE, ANDERBERG, CHISQUARE, EUCLIDEAN, GAMMA, JACCARD, MAHALANOBIS, MINKOWSKI, PEARSON, PERCENT, PHISQUARE, RSQUARED, RT, RUSSEL, SS. For MINKOWSKI, specify the root using POWER=*p*. For COV=*matrix*, separate columns by space and separate rows by semicolon. Use GROUP=*var*, to compute inter-group distances.

The linkage methods include AVERAGE, CENTROID, COMPLETE, MEDIAN, SINGLE, KNBD, UNIFORM, FLEXIBETA, WARD and WEIGHT.

More than one validity index can be specified at a time.

Resampling is available only in joining columns.

### *For the K-Means clustering method:*

```
CLUSTER
  USE filename
  IDVAR var$
  SAVE filename / NUMBER=n DATA
  KMEANS varlist / NUMBER=n ITER=n DISTANCE=metric POWER=p
                   COV=matrix or 'filename' GROUP=var
                   INITIAL=option INIFILE='filename'
                   PARTITION=var LINKAGE=method
```

The distance *metric* is ABSOLUTE, CHISQUARE, EUCLIDEAN, GAMMA, MAHALANOBIS, MINKOWSKI, MW, PEARSON, PHISQUARE or RSQUARED. For MINKOWSKI, specify the root using POWER=*p*. For COV=*matrix*, separate columns by space and separate rows by semicolon. Use GROUP=*var*, to compute inter-group distances.

The options for initial seeds are NONE, FIRSTK, LASTK, RANDOMK, RANDSEG, PCA and HIERSEG. Initial seeds can also be specified from a file or through a variable. For HIERSEG, specify the linkage method using LINKAGE=*method*. The linkage methods are mentioned below:

AVERAGE, CENTROID, COMPLETE, MEDIAN, SINGLE, WARD, WEIGHTED.

***For the K-Medians clustering method:***

```
CLUSTER
  USE filename
  IDVAR var$
  SAVE filename / NUMBER=n DATA or SEEDS
  KMEDIANS varlist / NUMBER=n ITER=n DISTANCE=metric POWER=p
                     COV=matrix or 'filename' GROUP=var
                     INITIAL=option INIFILE='filename'
                     PARTITION=var LINKAGE=method
```

The distance metric is ABSOLUTE, CHISQUARE, EUCLIDEAN, GAMMA, MAHALANOBIS, MINKOWSKI, MW, PEARSON, PHISQUARE or RSQUARED. For MINKOWSKI, specify the root using POWER=*p*. For COV=*matrix*, separate columns by space and separate rows by semi colon. Use GROUP=*var*, to compute inter-group distances.

The options for initial seeds are NONE, FIRSTK, LASTK, RANDOMK, RANDSEG, PCA and HIERSEG. Initial seeds can also be specified from a file or through a variable. For HIERSEG, specify the linkage method using LINKAGE=*method*. The linkage methods are mentioned below:

AVERAGE, CENTROID, COMPLETE, MEDIAN, SINGLE, WARD, WEIGHTED.

***For the Additive trees:***

```
CLUSTER
  USE filename
  ADD varlist / DATA TRANSFORMED MODEL RESIDUALS
                TREE NUMBERS NOSUBTRACT HEIGHT
                MINVAR ROOT = n1, n2
```

## *Usage Considerations*

**Types of data.** Hierarchical Clustering works on either rectangular SYSTAT files or files containing a symmetric matrix, such as those produced with Correlations. *K*-Clustering works only on rectangular SYSTAT files. Additive Trees works only on symmetric (similarity or dissimilarity) matrices.

**Print options.** PLENGTH options are effective only in Additive Trees.

**Quick Graphs.** Cluster analysis includes Quick Graphs for each procedure. Hierarchical Clustering and Additive Trees have tree diagrams. For each cluster, *K*-Clustering displays a profile plot of the data, a parallel coordinates display and a display of the variable means and standard deviations. Also, *K*-Clustering produces a scatterplot matrix with different colors and symbols based on final cluster identifiers. To omit Quick Graphs, specify GRAPH NONE.

**Saving files.** CLUSTER saves cluster indices as a new variable.

**BY groups.** CLUSTER analyzes data by groups.

**Labeling output.** For Hierarchical Clustering and *K*-Clustering, be sure to consider using the ID Variable (on the Data menu) for labeling the output.

# *Examples*

## *Example 1*
## *K-Means Clustering*

The data in the file *SUBWORLD* are a subset of cases and variables from the *OURWORLD* file:

| | |
|---|---|
| *URBAN* | Percentage of the population living in cities |
| *BIRTH_RT* | Births per 1000 people |
| *DEATH_RT* | Deaths per 1000 people |
| *B_TO_D* | Ratio of births to deaths |
| *BABYMORT* | Infant deaths during the first year per 1000 live births |
| *GDP_CAP* | Gross domestic product per capita (in U.S. dollars) |
| *LIFEEXPM* | Years of life expectancy for males |
| *LIFEEXPF* | Years of life expectancy for females |
| *EDUC* | U.S. dollars spent per person on education |
| *HEALTH* | U.S. dollars spent per person on health |
| *MIL* | U.S. dollars spent per person on the military |
| *LITERACY* | Percentage of the population who can read |

The distributions of the economic variables (*GDP_CAP*, *EDUC*, *HEALTH*, and *MIL*) are skewed with long right tails, so these variables are analyzed in log units.

This example clusters countries (cases).

The input is:

```
CLUSTER
  USE SUBWORLD
  IDVAR COUNTRY$
  LET (GDP_CAP, EDUC, MIL, HEALTH) = L10(@)
  STANDARDIZE / SD
  KMEANS URBAN BIRTH_RT DEATH_RT BABYMORT LIFEEXPM,
         LIFEEXPF GDP_CAP B_TO_D LITERACY EDUC,
         MIL HEALTH / NUMBER=4
```

Note that KMEANS must be specified last.

The output is:

**Distance Metric is Euclidean Distance**
**Single Linkage Method (Nearest Neighbor)**
**K-Means splitting cases into 4 groups**

**Summary Statistics for All Cases**

| Variable | Between SS | df | Within SS | df | F-ratio |
|---|---|---|---|---|---|
| URBAN | 18.606 | 3 | 9.394 | 25 | 16.506 |
| BIRTH_RT | 26.204 | 3 | 2.796 | 26 | 81.226 |
| DEATH_RT | 23.663 | 3 | 5.337 | 26 | 38.422 |
| BABYMORT | 26.028 | 3 | 2.972 | 26 | 75.887 |
| LIFEEXPM | 24.750 | 3 | 4.250 | 26 | 50.473 |
| LIFEEXPF | 25.927 | 3 | 3.073 | 26 | 73.122 |
| GDP_CAP | 26.959 | 3 | 2.041 | 26 | 114.447 |
| B_TO_D | 22.292 | 3 | 6.708 | 26 | 28.800 |
| LITERACY | 24.854 | 3 | 4.146 | 26 | 51.947 |
| EDUC | 25.371 | 3 | 3.629 | 26 | 60.593 |
| MIL | 24.787 | 3 | 3.213 | 25 | 64.289 |
| HEALTH | 24.923 | 3 | 3.077 | 25 | 67.488 |
| ** TOTAL ** | 294.362 | 36 | 50.638 | 309 | |

**Cluster 1 of 4 Contains 12 Cases**

| Members | | Statistics | | | | |
|---|---|---|---|---|---|---|
| Case | Distance | Variable | Minimum | Mean | Maximum | Standard Deviation |
| Austria | 0.283 | URBAN | -0.166 | 0.602 | 1.587 | 0.540 |
| Belgium | 0.091 | BIRTH_RT | -1.137 | -0.934 | -0.832 | 0.105 |
| Denmark | 0.189 | DEATH_RT | -0.770 | 0.000 | 0.257 | 0.346 |
| France | 0.140 | BABYMORT | -0.852 | -0.806 | -0.676 | 0.052 |
| Switzerland | 0.260 | LIFEEXPM | 0.233 | 0.745 | 0.988 | 0.230 |
| UK | 0.137 | LIFEEXPF | 0.430 | 0.793 | 1.065 | 0.182 |
| Italy | 0.160 | GDP_CAP | 0.333 | 1.014 | 1.275 | 0.257 |
| Sweden | 0.228 | B_TO_D | -1.092 | -0.905 | -0.462 | 0.180 |
| WGermany | 0.310 | LITERACY | 0.540 | 0.721 | 0.747 | 0.059 |
| Poland | 0.391 | EDUC | 0.468 | 0.947 | 1.281 | 0.277 |
| Czechoslov | 0.265 | MIL | 0.285 | 0.812 | 1.109 | 0.252 |
| Canada | 0.301 | HEALTH | 0.523 | 0.988 | 1.309 | 0.234 |

**Cluster 2 of 4 Contains 5 Cases**

| Members | | Statistics | | | | |
|---|---|---|---|---|---|---|
| Case | Distance | Variable | Minimum | Mean | Maximum | Standard Deviation |
| Ethiopia | 0.397 | URBAN | -2.008 | -1.694 | -1.289 | 0.305 |
| Guinea | 0.519 | BIRTH_RT | 1.458 | 1.580 | 1.687 | 0.102 |
| Somalia | 0.381 | DEATH_RT | 1.284 | 1.848 | 3.081 | 0.757 |
| Afghanistan | 0.383 | BABYMORT | 1.384 | 1.883 | 2.414 | 0.440 |
| Haiti | 0.298 | LIFEEXPM | -2.783 | -1.900 | -1.383 | 0.557 |
| | | LIFEEXPF | -2.475 | -1.912 | -1.477 | 0.447 |
| | | GDP_CAP | -1.999 | -1.614 | -1.270 | 0.300 |
| | | B_TO_D | -0.376 | -0.018 | 0.252 | 0.258 |
| | | LITERACY | -2.268 | -1.828 | -0.764 | 0.619 |
| | | EDUC | -2.411 | -1.582 | -1.096 | 0.511 |
| | | MIL | -1.763 | -1.509 | -1.374 | 0.173 |
| | | HEALTH | -2.222 | -1.638 | -1.290 | 0.438 |

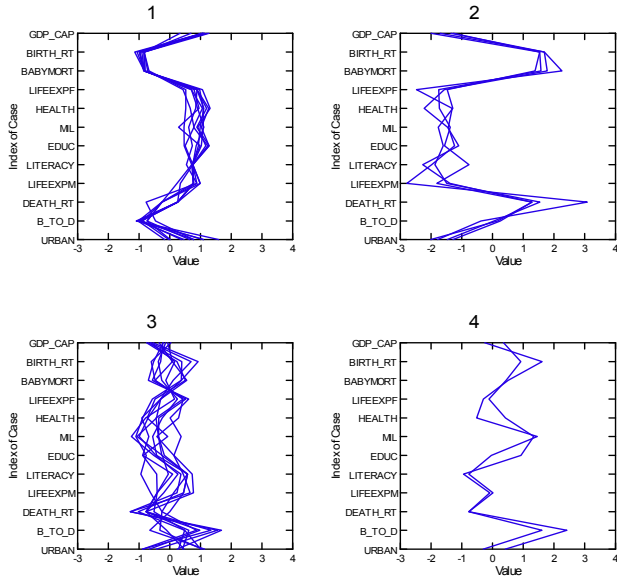
**Cluster 3 of 4 Contains 11 Cases**

| Members | | Statistics | | | | |
|---|---|---|---|---|---|---|
| Case | Distance | Variable | Minimum | Mean | Maximum | Standard Deviation |
| Argentina | 0.450 | URBAN | -0.885 | 0.157 | 1.137 | 0.764 |
| Brazil | 0.315 | BIRTH_RT | -0.603 | 0.070 | 0.923 | 0.490 |
| Chile | 0.397 | DEATH_RT | -1.284 | -0.700 | 0.000 | 0.415 |
| Colombia | 0.422 | BABYMORT | -0.698 | -0.063 | 0.551 | 0.465 |
| Uruguay | 0.606 | LIFEEXPM | -0.628 | 0.057 | 0.772 | 0.492 |
| Ecuador | 0.364 | LIFEEXPF | -0.569 | 0.042 | 0.611 | 0.435 |
| ElSalvador | 0.520 | GDP_CAP | -0.753 | -0.382 | 0.037 | 0.278 |
| Guatemala | 0.646 | B_TO_D | -0.651 | 0.630 | 1.680 | 0.759 |
| Peru | 0.369 | LITERACY | -0.943 | 0.200 | 0.730 | 0.506 |
| Panama | 0.514 | EDUC | -0.888 | -0.394 | 0.135 | 0.357 |
| Cuba | 0.576 | MIL | -1.250 | -0.591 | 0.371 | 0.492 |
| | | HEALTH | -0.911 | -0.474 | 0.284 | 0.382 |

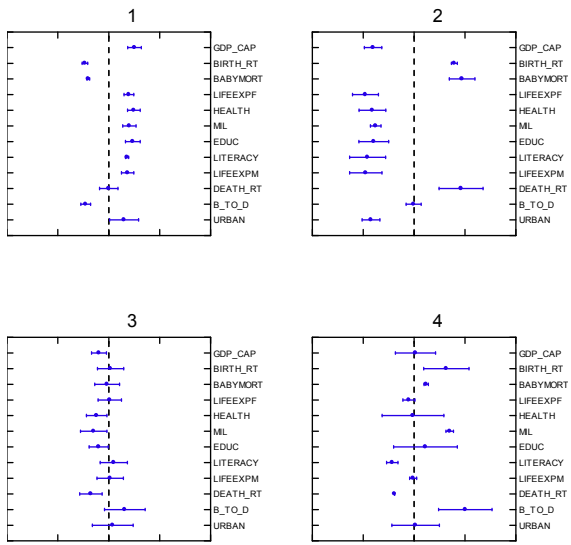
**Cluster 4 of 4 Contains 2 Cases**

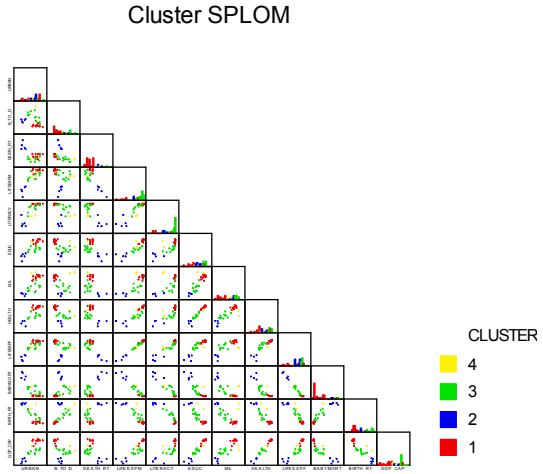
| Members | | Statistics | | | | |
|---|---|---|---|---|---|---|
| Case | Distance | Variable | Minimum | Mean | Maximum | Standard Deviation |
| Iraq | 0.285 | URBAN | -0.301 | 0.059 | 0.418 | 0.508 |
| Libya | 0.285 | BIRTH_RT | 0.923 | 1.267 | 1.610 | 0.486 |
| | | DEATH_RT | -0.770 | -0.770 | -0.770 | 0.000 |
| | | BABYMORT | 0.441 | 0.474 | 0.507 | 0.046 |
| | | LIFEEXPM | -0.090 | -0.036 | 0.018 | 0.076 |
| | | LIFEEXPF | -0.297 | -0.206 | -0.115 | 0.128 |
| | | GDP_CAP | -0.251 | 0.053 | 0.357 | 0.430 |
| | | B_TO_D | 1.608 | 2.012 | 2.417 | 0.573 |
| | | LITERACY | -0.943 | -0.857 | -0.771 | 0.122 |
| | | EDUC | -0.037 | 0.444 | 0.925 | 0.680 |
| | | MIL | 1.344 | 1.400 | 1.456 | 0.079 |
| | | HEALTH | -0.512 | -0.045 | 0.422 | 0.661 |

## Cluster Parallel Coordinate Plots




## Cluster Profile Plots






For each variable, cluster analysis compares the between-cluster mean square (*Between SS/df*) to the within-cluster mean square (*Within SS/df*) and reports the *F-ratio*. However, do not use these *F-ratios* to test significance because the clusters are formed to characterize differences. Instead, use these statistics to characterize relative discrimination. For example, the log of gross domestic product (*GDP_CAP*) and *BIRTH_RT* are better discriminators between countries than *URBAN* or *DEATH_RT*. For a good graphical view of the separation of the clusters, you might rotate the data using the three variables with the highest *F-ratios*.

Following the summary statistics, for each cluster, cluster analysis prints the distance from each case (country) in the cluster to the center of the cluster. Descriptive statistics for these countries appear on the right. For the first cluster, the standard scores for *LITERACY* range from 0.54 to 0.75 with an average of 0.72. *B_TO_D* ranges from –1.09 to –0.46. Thus, for these predominantly European countries, literacy is well above the average for the sample and the birth-to-death ratio is below average. In cluster 2, *LITERACY* ranges from –2.27 to –0.76 for these five countries, and *B_TO_D* ranges from –0.38 to 0.25. Thus, the countries in cluster 2 have a lower literacy rate and a greater potential for population growth than those in cluster 1. The fourth cluster (Iraq and Libya) has an average birth-to-death ratio of 2.01, the highest among the four clusters.

### *Cluster Parallel Coordinates*

The variables in this Quick Graph are ordered by their *F-ratios*. In the top left plot, there is one line for each country in cluster 1 that connects its *z* scores for each of the variables. Zero marks the average for the complete sample. The lines for these 12 countries all follow a similar pattern: above average values for *GDP_CAP*, below average for *BIRTH_RT* and so on. The lines in cluster 3 do not follow such a tight pattern.

### *Cluster Profiles*

The variables in cluster profile plots are ordered by the *F-ratios*. The vertical line under each cluster number indicates the grand mean across all data. A variable mean within each cluster is marked by a dot. The horizontal lines indicate one standard deviation above or below the mean. The countries in cluster 1 have above average means of gross domestic product, life expectancy, literacy, and urbanization, and spend considerable money on health care and the military, while the means of their birth rates, infant mortality rates, and birth-to-death ratios are low. The opposite is true for cluster 2.

### *Scatterplot Matrix*

In the scatterplot matrix (SPLOM), the off-diagonal cells are the scatterplot of two variables at a time and the diagonal cells are the histogram of variables. The off-diagonal cells in the SPLOM are such that observations belonging to the same cluster will have the same color and symbol.

### *K-Medians Cluster Analysis with Subworld*

The input is:

```
CLUSTER
  USE SUBWORLD
  IDVAR COUNTRY$
  LET (GDP_CAP, EDUC, MIL, HEALTH) = L10(@)
  STANDARDIZE / SD
  KMEDIANS URBAN BIRTH_RT DEATH_RT BABYMORT LIFEEXPM,
           LIFEEXPF GDP_CAP B_TO_D LITERACY EDUC,
           MIL HEALTH / DISTANCE =ABSOLUTE NUMBER=4
```

The output is:

**Distance Metric is Absolute Distance**
**Single Linkage Method (Nearest Neighbor)**
**K-Medians splitting cases into 4 groups**

**Summary Statistics for 4 Clusters**

| Variable | Within Sum of Absolute Deviation |
|---|---|
| URBAN | 12.087 |
| BIRTH_RT | 5.648 |
| DEATH_RT | 8.985 |
| BABYMORT | 3.835 |
| LIFEEXPM | 6.249 |
| LIFEEXPF | 4.902 |
| GDP_CAP | 5.459 |
| B_TO_D | 9.610 |
| LITERACY | 6.247 |
| EDUC | 8.453 |
| MIL | 10.136 |
| HEALTH | 6.734 |
| ** TOTAL ** | 88.346 |

**Cluster 1 of 4 Contains 12 Cases**

| Members | | Statistics | | | | |
|---|---|---|---|---|---|---|
| Case | Distance | Variable | Minimum | Median | Maximum | Mean Absolute Deviation |
| Austria | 0.142 | URBAN | -0.166 | 0.643 | 1.587 | 0.425 |
| Belgium | 0.035 | BIRTH_RT | -1.137 | -0.946 | -0.832 | 0.089 |
| Denmark | 0.093 | DEATH_RT | -0.770 | 0.257 | 0.257 | 0.257 |
| France | 0.114 | BABYMORT | -0.852 | -0.830 | -0.676 | 0.027 |
| Switzerland | 0.206 | LIFEEXPM | 0.233 | 0.772 | 0.988 | 0.135 |
| UK | 0.075 | LIFEEXPF | 0.430 | 0.838 | 1.065 | 0.136 |
| Italy | 0.163 | GDP_CAP | 0.333 | 1.079 | 1.275 | 0.139 |
| Sweden | 0.132 | B_TO_D | -1.092 | -0.949 | -0.462 | 0.127 |
| WGermany | 0.130 | LITERACY | 0.540 | 0.747 | 0.747 | 0.026 |
| Poland | 0.384 | EDUC | 0.468 | 0.959 | 1.281 | 0.215 |
| Czechoslov | 0.218 | MIL | 0.285 | 0.847 | 1.109 | 0.189 |
| Canada | 0.224 | HEALTH | 0.523 | 1.007 | 1.309 | 0.183 |

**Cluster 2 of 4 Contains 5 Cases**

| Members | | Statistics | | | | |
|---|---|---|---|---|---|---|
| Case | Distance | Variable | Minimum | Median | Maximum | Mean Absolute Deviation |
| Argentina | 0.169 | URBAN | -0.435 | 1.002 | 1.137 | 0.431 |
| Chile | 0.102 | BIRTH_RT | -0.603 | -0.374 | 0.084 | 0.183 |
| Uruguay | 0.185 | DEATH_RT | -1.284 | -0.770 | 0.000 | 0.411 |
| Panama | 0.453 | BABYMORT | -0.698 | -0.479 | -0.260 | 0.105 |
| Cuba | 0.240 | LIFEEXPM | 0.126 | 0.449 | 0.772 | 0.172 |
| | | LIFEEXPF | 0.248 | 0.430 | 0.611 | 0.337 |
| | | GDP_CAP | -0.347 | -0.216 | 0.037 | 0.345 |
| | | B_TO_D | -0.651 | -0.102 | 1.554 | 0.867 |
| | | LITERACY | 0.437 | 0.575 | 0.730 | 0.271 |
| | | EDUC | -0.427 | -0.175 | 0.135 | 0.487 |
| | | MIL | -0.560 | -0.368 | 0.371 | 0.498 |
| | | HEALTH | -0.448 | -0.243 | 0.284 | 0.525 |

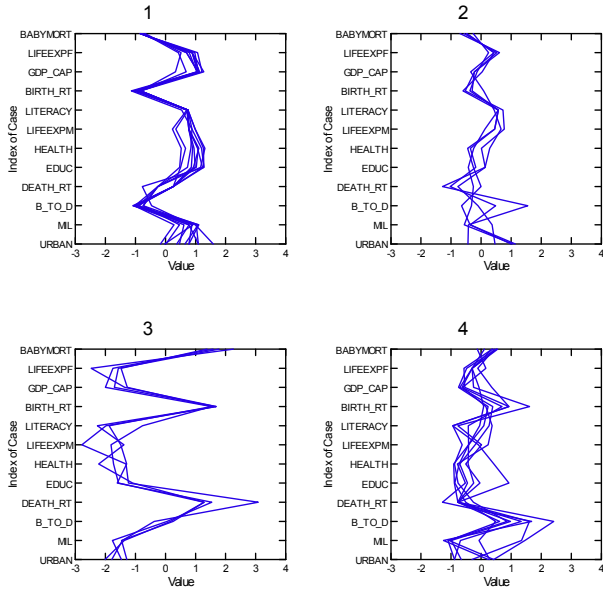
**Cluster 3 of 4 Contains 5 Cases**

| Members | | Statistics | | | | |
|---|---|---|---|---|---|---|
| Case | Distance | Variable | Minimum | Median | Maximum | Mean Absolute Deviation |
| Ethiopia | 0.216 | URBAN | -2.008 | -1.783 | -1.289 | 0.234 |
| Guinea | 0.433 | BIRTH_RT | 1.458 | 1.534 | 1.687 | 0.076 |
| Somalia | 0.266 | DEATH_RT | 1.284 | 1.540 | 3.081 | 0.513 |
| Afghanistan | 0.352 | BABYMORT | 1.384 | 1.778 | 2.414 | 0.342 |
| Haiti | 0.202 | LIFEEXPM | -2.783 | -1.814 | -1.383 | 0.388 |
| | | LIFEEXPF | -2.475 | -1.749 | -1.477 | 0.657 |
| | | GDP_CAP | -1.999 | -1.701 | -1.270 | 0.529 |
| | | B_TO_D | -0.376 | 0.050 | 0.252 | 0.494 |
| | | LITERACY | -2.268 | -1.978 | -0.764 | 0.686 |
| | | EDUC | -2.411 | -1.563 | -1.096 | 0.649 |
| | | MIL | -1.763 | -1.450 | -1.374 | 0.372 |
| | | HEALTH | -2.222 | -1.520 | -1.290 | 0.670 |

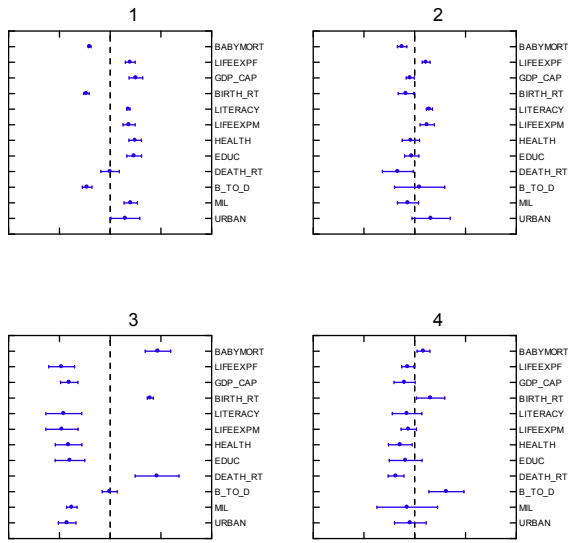
**Cluster 4 of 4 Contains 8 Cases**

| Members | | Statistics | | | | |
|---|---|---|---|---|---|---|
| Case | Distance | Variable | Minimum | Median | Maximum | Mean Absolute Deviation |
| Iraq | 0.585 | URBAN | -0.885 | -0.031 | 0.418 | 0.511 |
| Libya | 0.659 | BIRTH_RT | 0.084 | 0.542 | 1.610 | 0.410 |
| Brazil | 0.263 | DEATH_RT | -1.284 | -0.770 | -0.257 | 0.160 |
| Colombia | 0.364 | BABYMORT | -0.129 | 0.408 | 0.551 | 0.159 |
| Ecuador | 0.160 | LIFEEXPM | -0.628 | -0.305 | 0.233 | 0.229 |
| ElSalvador | 0.215 | LIFEEXPF | -0.569 | -0.297 | 0.157 | 0.136 |
| Guatemala | 0.343 | GDP_CAP | -0.753 | -0.579 | 0.357 | 0.275 |
| Peru | 0.301 | B_TO_D | 0.483 | 1.158 | 2.417 | 0.512 |
| | | LITERACY | -0.943 | -0.236 | 0.368 | 0.733 |
| | | EDUC | -0.888 | -0.487 | 0.925 | 0.680 |
| | | MIL | -1.250 | -0.838 | 1.456 | 0.960 |
| | | HEALTH | -0.911 | -0.721 | 0.422 | 0.545 |

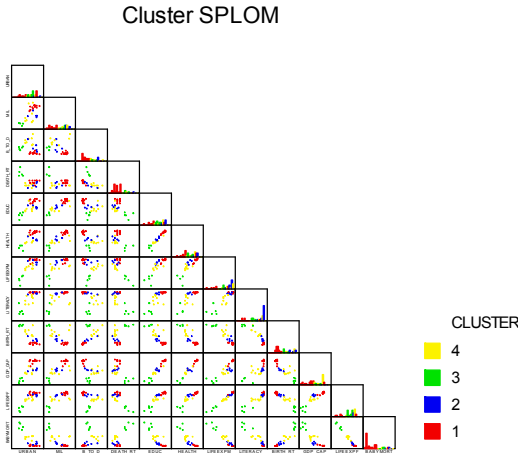
## Cluster Parallel Coordinate Plots




## Cluster Profile Plots



Scatter Plot Matrix




## *Example 2*
## *Hierarchical Clustering: Clustering Cases*

This example uses the *SUBWORLD* data (see the *K*-Means example for a description) to cluster cases.

The input is:

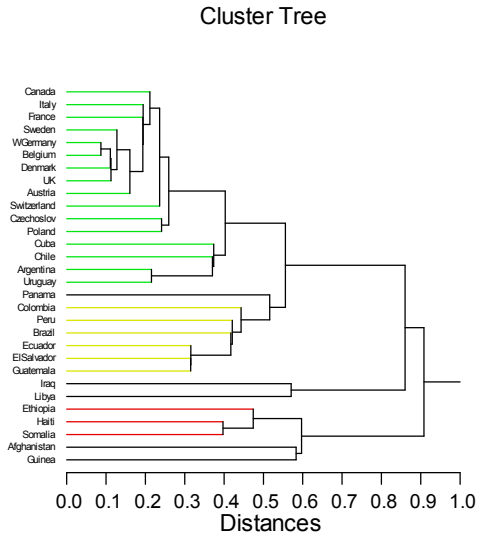
```
CLUSTER
  USE SUBWORLD
  IDVAR COUNTRY$
  LET (GDP_CAP, EDUC, MIL, HEALTH) = L10(@)
  STANDARDIZE / SD
  JOIN URBAN BIRTH_RT DEATH_RT BABYMORT LIFEEXPM,
       LIFEEXPF GDP_CAP B_TO_D LITERACY EDUC MIL HEALTH
```

The output is:

```
Distance Metric is Euclidean Distance
Single Linkage Method (Nearest Neighbor)

     Clusters Joining       at Distance   No. of Members
 ------------------------------------------------------------
  WGermany      Belgium          0.087              2
  WGermany      Denmark          0.111              3
  WGermany      UK               0.113              4
  Sweden        WGermany         0.128              5
  Austria       Sweden           0.161              6
  Austria       France           0.194              7
  Austria       Italy            0.194              8
```

| | | | |
|---|---|---|---|
| Austria | Canada | 0.211 | 9 |
| Uruguay | Argentina | 0.215 | 2 |
| Switzerland | Austria | 0.236 | 10 |
| Czechoslov | Poland | 0.241 | 2 |
| Switzerland | Czechoslov | 0.260 | 12 |
| Guatemala | ElSalvador | 0.315 | 2 |
| Guatemala | Ecuador | 0.316 | 3 |
| Uruguay | Chile | 0.370 | 3 |
| Cuba | Uruguay | 0.374 | 4 |
| Haiti | Somalia | 0.397 | 2 |
| Switzerland | Cuba | 0.403 | 16 |
| Guatemala | Brazil | 0.417 | 4 |
| Peru | Guatemala | 0.421 | 5 |
| Colombia | Peru | 0.443 | 6 |
| Ethiopia | Haiti | 0.474 | 3 |
| Panama | Colombia | 0.516 | 7 |
| Switzerland | Panama | 0.556 | 23 |
| Libya | Iraq | 0.570 | 2 |
| Afghanistan | Guinea | 0.583 | 2 |
| Ethiopia | Afghanistan | 0.597 | 5 |
| Switzerland | Libya | 0.860 | 25 |
| Switzerland | Ethiopia | 0.908 | 30 |



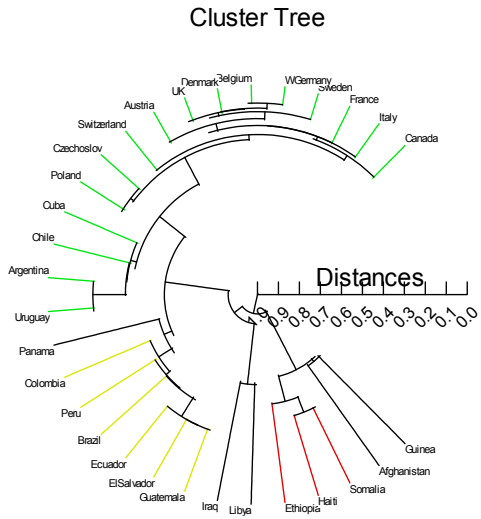

The numerical results consist of the joining history. The countries at the top of the panel are joined first at a distance of 0.087. The last entry represents the joining of the largest two clusters to form one cluster of all 30 countries. Switzerland is in one of the clusters and Ethiopia is in the other.

The clusters are best illustrated using a tree diagram. Because the example joins rows (cases) and uses *COUNTRY* as an *ID* variable, the branches of the tree are labeled with countries. If you join columns (variables), then variable names are used. The scale for the joining distances is printed at the bottom. Notice that Iraq and Libya, which

form their own cluster as they did in the *K*-Means example, are the second-to-last cluster to link with others. They join with all the countries listed above them at a distance of 0.860. Finally, at a distance of 0.908, the five countries at the bottom of the display are added to form one large cluster.

### *Polar Dendrogram*

Adding the POLAR option to JOIN yields a polar dendrogram.



## *Example 3*
## *Hierarchical Clustering: Clustering Variables*

This example joins columns (variables) instead of rows (cases) to see which variables cluster together.

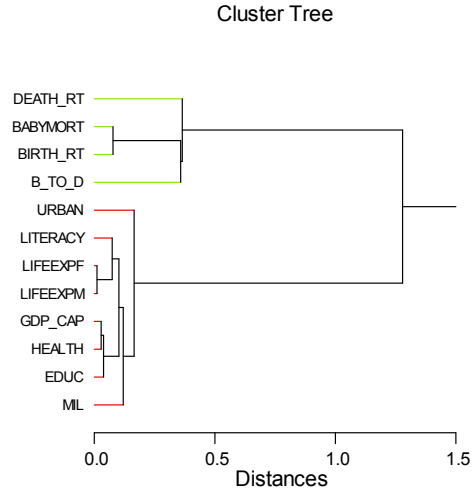
The input is:

```
CLUSTER
  USE SUBWORLD
  IDVAR COUNTRY$
  LET (GDP_CAP, EDUC, MIL, HEALTH) = L10(@)
  STANDARDIZE / SD
  JOIN URBAN BIRTH_RT DEATH_RT BABYMORT LIFEEXPM,
       LIFEEXPF GDP_CAP B_TO_D LITERACY,
        EDUC MIL HEALTH / COLUMNS DISTANCE=PEARSON
```

The output is:

```
Distance Metric is 1-Pearson Correlation Coefficient
Single Linkage Method (Nearest Neighbor)
```

| Clusters Joining | | at Distance | No. of Members |
|---|---|---|---|
| LIFEEXPF | LIFEEXPM | 0.011 | 2 |
| HEALTH | GDP_CAP | 0.028 | 2 |
| EDUC | HEALTH | 0.038 | 3 |
| LIFEEXPF | LITERACY | 0.074 | 3 |
| BABYMORT | BIRTH_RT | 0.077 | 2 |
| EDUC | LIFEEXPF | 0.102 | 6 |
| MIL | EDUC | 0.120 | 7 |
| MIL | URBAN | 0.165 | 8 |
| B_TO_D | BABYMORT | 0.358 | 3 |
| B_TO_D | DEATH_RT | 0.365 | 4 |
| B_TO_D | MIL | 1.279 | 12 |




The scale at the bottom of the tree for the distance (1–r) ranges from 0.0 to 1.5. The smallest distance is 0.011—thus, the correlation of *LIFEEXPM* with *LIFEEXPF* is 0.989.

## *Example 4*
## *Hierarchical Clustering: Clustering Variables and Cases*

To produce a shaded display of the original data matrix in which rows and columns are permuted according to an algorithm in Gruvaeus and Wainer (1972), use the MATRIX option. Different shadings or colors represent the magnitude of each number in the matrix (Ling, 1973).

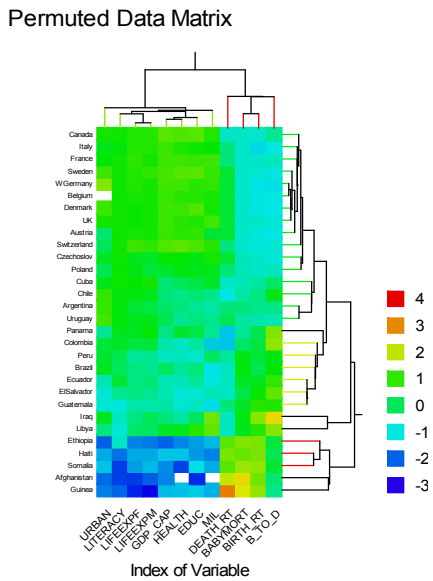
If you use the MATRIX option with Euclidean distance, be sure that the variables are on comparable scales because both rows and columns of the matrix are clustered. Joining a matrix containing inches of annual rainfall and annual growth of trees in feet, for example, would split columns more by scales than by covariation. In cases like this, you should standardize your data before joining.

The input is:

```
CLUSTER
  USE SUBWORLD
  IDVAR COUNTRY$
  LET (GDP_CAP, EDUC, MIL, HEALTH) = L10(@)
  STANDARDIZE / SD
  JOIN URBAN BIRTH_RT DEATH_RT BABYMORT LIFEEXPM,
       LIFEEXPF GDP_CAP B_TO_D LITERACY EDUC,
        MIL HEALTH / MATRIX
```

The output is:

```
Distance Metric is Euclidean Distance
Single Linkage Method (Nearest Neighbor)
```




This clustering reveals three groups of countries and two groups of variables. The countries with more urban dwellers and literate citizens, longest life-expectancies, highest gross domestic product, and most expenditures on health care, education, and the military are on the top left of the data matrix; countries with the highest rates of death, infant mortality, birth, and population growth (see *B_TO_D*) are on the lower right. You can also see that, consistent with the KMEANS and JOIN examples, Iraq and Libya spend much more on military, education, and health than their immediate neighbors.

## *Example 5*
## *Hierarchical Clustering: Distance Matrix Input*

This example clusters a matrix of distances. The data, stored as a dissimilarity matrix in the *CITIES* data file, are airline distances in hundreds of miles between 10 global cities. The data are adapted from Hartigan (1975).

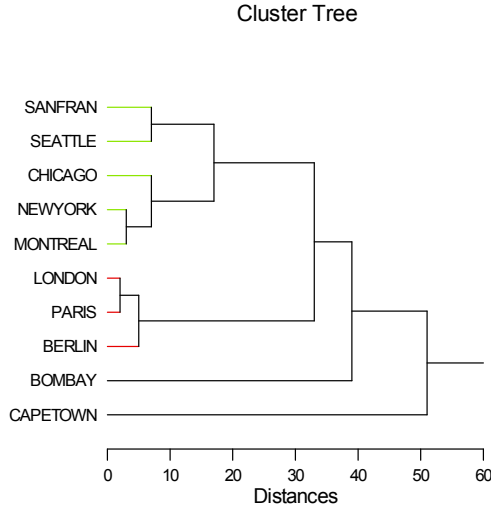
The input is:

```
CLUSTER
  USE CITIES
  JOIN BERLIN BOMBAY CAPETOWN CHICAGO LONDON,
       MONTREAL NEWYORK PARIS SANFRAN SEATTLE
```

The output is:

**Single Linkage Method (Nearest Neighbor)**

| Clusters Joining | | at Distance | No. of Members |
|---|---|---|---|
| PARIS | LONDON | 2.000 | 2 |
| NEWYORK | MONTREAL | 3.000 | 2 |
| BERLIN | PARIS | 5.000 | 3 |
| CHICAGO | NEWYORK | 7.000 | 3 |
| SEATTLE | SANFRAN | 7.000 | 2 |
| SEATTLE | CHICAGO | 17.000 | 5 |
| BERLIN | SEATTLE | 33.000 | 8 |
| BOMBAY | BERLIN | 39.000 | 9 |
| BOMBAY | CAPETOWN | 51.000 | 10 |



The tree is printed in seriation order. Imagine a trip around the globe to these cities. SYSTAT has identified the shortest path between cities. The itinerary begins at San Francisco, leads to Seattle, Chicago, New York, and so on, and ends in Capetown.

Note that the *CITIES* data file contains the distances between the cities; SYSTAT did not have to compute those distances. When you save the file, be sure to save it as a dissimilarity matrix.

This example is used both to illustrate direct distance input and to give you an idea of the kind of information contained in the order of the SYSTAT cluster tree. For distance data, the seriation reveals shortest paths; for typical sample data, the seriation is more likely to replicate in new samples so that you can recognize cluster structure.

## *Example 6*
## *Density Clustering Examples*

### *K-th Nearest Neighbor Density Linkage Clustering*

The data file *CARS* is used for analysis of Hierarchical Clustering using *K*-th Nearest Neighbor density linkage clustering.

The variables in the *CARS* data which are used for analysis are *ACCEL*, *BRAKE*, *SLALOM*, *MPG* and *SPEED*.

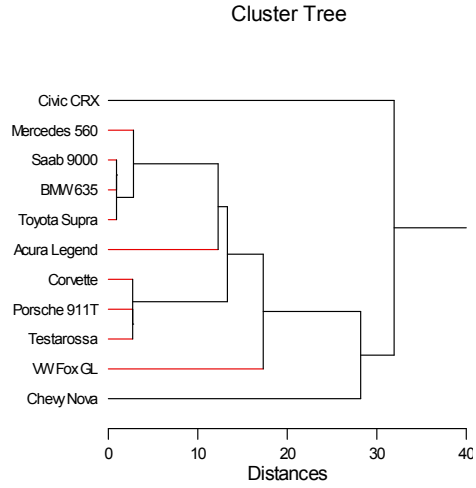
The input is:

```
CLUSTER
  USE CARS
  IDVAR NAME$
  STANDARDIZE ACCEL BRAKE SLALOM MPG SPEED
  JOIN ACCEL BRAKE SLALOM MPG SPEED/LINKAGE=KNBD K=3
```

The output is:

```
Distance Metric is Euclidean Distance
KNBD Density Linkage Method for K = 3
```

| Clusters Joining | | at Distance | No. of Members |
|---|---|---|---|
| BMW 635 | Saab 9000 | 0.914 | 2 |
| Toyota Supra | BMW 635 | 0.914 | 3 |
| Testarossa | Porsche 911T | 2.715 | 2 |
| Corvette | Testarossa | 2.715 | 3 |
| Mercedes 560 | Toyota Supra | 2.808 | 4 |
| Mercedes 560 | Acura Legend | 12.274 | 5 |
| Corvette | Mercedes 560 | 13.309 | 8 |
| VW Fox GL | Corvette | 17.320 | 9 |
| VW Fox GL | Chevy Nova | 28.192 | 10 |
| VW Fox GL | Civic CRX | 31.941 | 11 |




### *Uniform Kernel Density Linkage Clustering*

The data file *CARS* is used for analysis of Hierarchical Clustering using Uniform Kernel density linkage clustering.

The variables in *CARS* data which are used for analysis are *ACCEL*, *BRAKE, SLALOM*, *MPG* and *SPEED*.

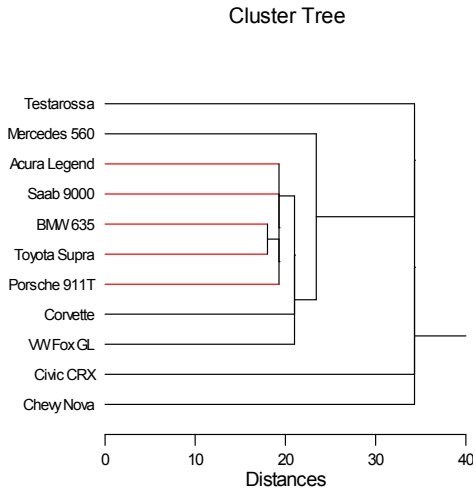
The input is:

```
CLUSTER
  USE CARS
  IDVAR NAME$
  STANDARDIZE ACCEL BRAKE SLALOM MPG SPEED
  JOIN ACCEL BRAKE SLALOM MPG SPEED/LINKAGE=UNIFORM RADIUS=1.2
```

The output is:

**Distance Metric is Euclidean Distance**
**Uniform Density Linkage Method for Radius = 1.200**

| Clusters Joining | | at Distance | No. of Members |
|---|---|---|---|
| BMW 635 | Toyota Supra | 18.010 | 2 |
| BMW 635 | Porsche 911T | 19.296 | 3 |
| Saab 9000 | BMW 635 | 19.296 | 4 |
| Acura Legend | Saab 9000 | 19.296 | 5 |
| Acura Legend | Corvette | 21.011 | 6 |
| VW Fox GL | Acura Legend | 21.011 | 7 |
| VW Fox GL | Mercedes 560 | 23.413 | 8 |
| VW Fox GL | Testarossa | 34.304 | 9 |
| Civic CRX | VW Fox GL | 34.304 | 10 |
| Chevy Nova | Civic CRX | 34.304 | 11 |



## *Example 7*
## *Flexible Beta Linkage Method for Hierarchical Clustering*

The data file *CARS* is used for the analysis of Hierarchical Clustering using Flexible beta linkage clustering.

The variables in *CARS* data which are used for analysis are *ACCEL, BRAKE, SLALOM, MPG* and *SPEED.*

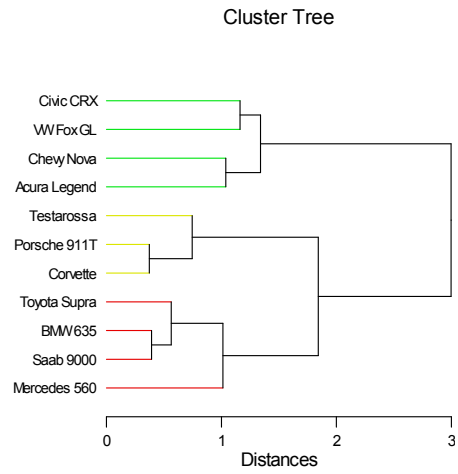
The input is:

```
CLUSTER
  USE CARS
  IDVAR NAME$
  STANDARDIZE ACCEL BRAKE SLALOM MPG SPEED
  JOIN ACCEL BRAKE SLALOM MPG SPEED/LINKAGE=FLEXIBETA BETA=-0.25
```

The output is:

```
Distance Metric is Euclidean Distance
Flexible Beta Linkage Method for Beta = -0.250
```

| Clusters Joining | | at Distance | No. of Members |
|---|---|---|---|
| Corvette | Porsche 911T | 0.373 | 2 |
| BMW 635 | Saab 9000 | 0.392 | 2 |
| Toyota Supra | BMW 635 | 0.563 | 3 |
| Corvette | Testarossa | 0.746 | 3 |
| Mercedes 560 | Toyota Supra | 1.013 | 4 |
| Chevy Nova | Acura Legend | 1.038 | 2 |
| VW Fox GL | Civic CRX | 1.161 | 2 |
| VW Fox GL | Chevy Nova | 1.339 | 4 |
| Corvette | Mercedes 560 | 1.842 | 7 |
| VW Fox GL | Corvette | 2.997 | 11 |



## *Example 8*
## *Validity indices RMSSTD, Pseudo F, and Pseudo T-square with cities*

In this example we have used the *CITIES* data file for the analysis of Hierarchical clustering for the validity of RMSSTD, PSEUDO F and PSEUDO T-SQUARE.

This analysis specifies how many good partitions can be made for the given data in hierarchical clustering.

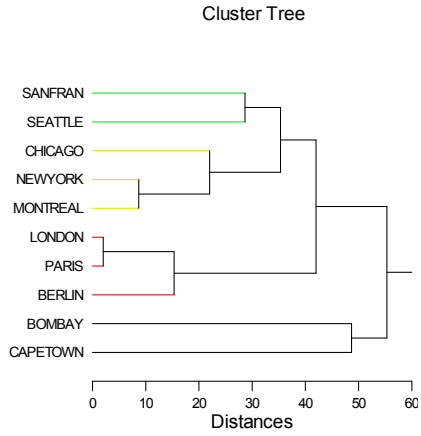
The input is:

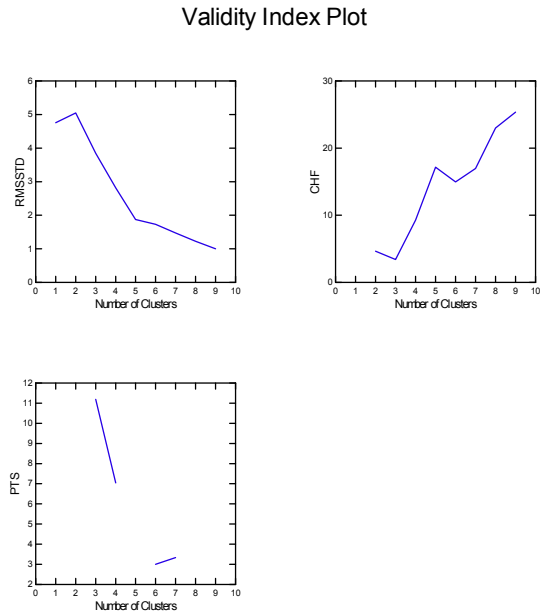
```
CLUSTER
  USE CITIES
  JOIN/LINKAGE=CENTROID VALIDITY = RMSSTD CHF PTS MAX=9
```

The output is:

**Centroid Linkage Method**

| Clusters Joining | | at Distance | No. of Members | RMSSTD | Pseudo F | Pseudo T-square |
|---|---|---|---|---|---|---|
| PARIS | LONDON | 2.000 | 2 | 1.000 | 25.350 | . |
| NEWYORK | MONTREAL | 3.000 | 2 | 1.225 | 23.006 | . |
| BERLIN | PARIS | 5.000 | 3 | 1.472 | 16.969 | 3.333 |
| CHICAGO | NEWYORK | 6.750 | 3 | 1.732 | 14.978 | 3.000 |
| SEATTLE | SANFRAN | 7.000 | 2 | 1.871 | 17.166 | . |
| SEATTLE | CHICAGO | 18.583 | 5 | 2.820 | 9.280 | 7.042 |
| BERLIN | SEATTLE | 35.929 | 8 | 3.845 | 3.392 | 11.186 |
| CAPETOWN | BOMBAY | 51.000 | 2 | 5.050 | 4.639 | . |
| CAPETOWN | BERLIN | 46.750 | 10 | 4.759 | . | 4.639 |




We observe that there is a “knee” in the RMSSTD plot at 5, a jump in the plot of the pseudo *T*-square also at 5 and a peak at the same point in the graph of pseudo *F*. Hence the appropriate number of clusters appears to be 5. In some data sets the indices may not all point to the same clustering; you must then choose the appropriate clustering scheme based on the type of data.

## *Example 9*
## *Hierarchical Clustering with Leaf Option*

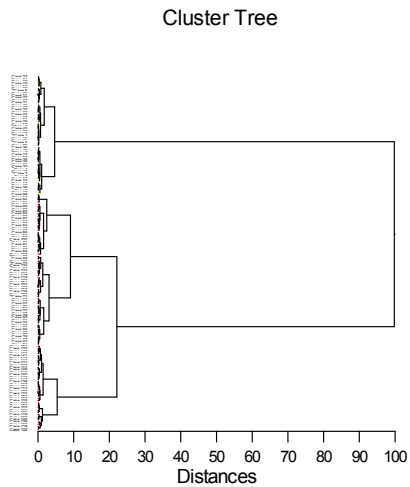
In this example we have used the *IRIS* data file for leaf option analysis in hierarchical clustering. The IRIS data file contains the following variables: *SPECIES*, *SEPALLEN*, *SEPALWID*, *PETALLEN* and *PETALWID*. It becomes difficult to understand the substructures of the data from the cluster trees when there are a large number of objects. In such cases, the LEAF option helps the user to concentrate on the upper part of the tree. In the following example with LEAF =13, SYSTAT provides another tree with 13 leaf nodes along with a partition table. The table shows the content of each node.

The input is:

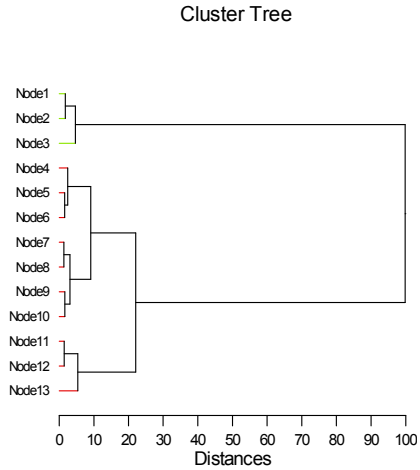
```
CLUSTER
  USE IRIS
  JOIN SEPALLEN SEPALWID PETALLEN PETALWID/LINKAGE=WARD LEAF=13
```

The following is a part of the output:

```
Distance Metric is Euclidean Distance
Ward Minimum Variance Method
```




```
Cluster Tree and Partition Table for LEAF = 13
```




| Node1 | Node2 | Node3 | Node4 | Node5 | Node6 | Node7 |
|---|---|---|---|---|---|---|
| Case 6 | Case 1 | Case 2 | Case 58 | Case 54 | Case 56 | Case 69 |
| Case 11 | Case 5 | Case 3 | Case 61 | Case 60 | Case 67 | Case 73 |
| Case 15 | Case 8 | Case 4 | Case 94 | Case 63 | Case 85 | Case 84 |
| Case 16 | Case 18 | Case 7 | Case 99 | Case 65 | Case 89 | Case 88 |
| Case 17 | Case 20 | Case 9 | | Case 68 | Case 91 | Case 120 |
| Case 19 | Case 22 | Case 10 | | Case 70 | Case 95 | Case 124 |
| Case 21 | Case 24 | Case 12 | | Case 80 | Case 96 | Case 127 |
| Case 32 | Case 25 | Case 13 | | Case 81 | Case 97 | Case 134 |
| Case 33 | Case 27 | Case 14 | | Case 82 | Case 100 | Case 147 |
| Case 34 | Case 28 | Case 23 | | Case 83 | Case 107 | |
| Case 37 | Case 29 | Case 26 | | Case 90 | | |
| Case 49 | Case 38 | Case 30 | | Case 93 | | |
| | Case 40 | Case 31 | | | | |
| | Case 41 | Case 35 | | | | |
| | Case 44 | Case 36 | | | | |
| | Case 45 | Case 39 | | | | |
| | Case 47 | Case 42 | | | | |
| | Case 50 | Case 43 | | | | |
| | | Case 46 | | | | |
| | | Case 48 | | | | |

| Node8 | Node9 | Node10 | Node11 | Node12 | Node13 |
|---|---|---|---|---|---|
| Case 71 | Case 52 | Case 51 | Case 101 | Case 104 | Case 103 |
| Case 102 | Case 57 | Case 53 | Case 111 | Case 105 | Case 106 |
| Case 114 | Case 62 | Case 55 | Case 113 | Case 109 | Case 108 |
| Case 115 | Case 64 | Case 59 | Case 116 | Case 112 | Case 110 |
| Case 122 | Case 72 | Case 66 | Case 121 | Case 117 | Case 118 |
| Case 128 | Case 74 | Case 76 | Case 125 | Case 129 | Case 119 |
| Case 139 | Case 75 | Case 77 | Case 137 | Case 133 | Case 123 |
| Case 143 | Case 79 | Case 78 | Case 140 | Case 135 | Case 126 |
| Case 150 | Case 86 | Case 87 | Case 141 | Case 138 | Case 130 |
| | Case 92 | | Case 142 | | Case 131 |
| | Case 98 | | Case 144 | | Case 132 |
| | | | Case 145 | | Case 136 |
| | | | Case 146 | | |
| | | | Case 148 | | |
| | | | Case 149 | | |

## *Example 10*
## *Additive Trees*

These data are adapted from an experiment by Rothkopf (1957) in which 598 subjects were asked to judge whether Morse code signals presented two in succession were the same. All possible ordered pairs were tested. For multidimensional scaling, the data for letter signals is averaged across the sequence and the diagonal (pairs of the same signal) is omitted. The variables are *A* through *Z*.

The input is:

```
CLUSTER
  USE ROTHKOPF
  ADD A .. Z
```

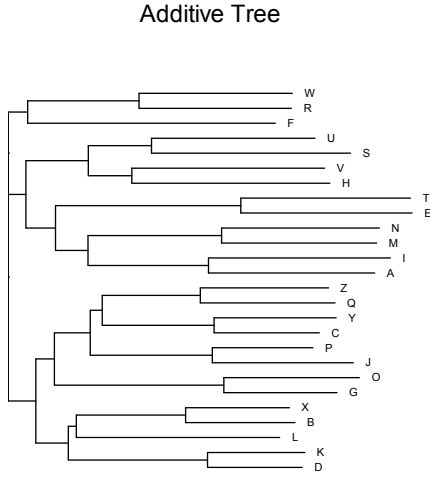
The output is:

```
Similarities linearly transformed into distances
77.000 needed to make distances positive
104.000 added to satisfy triangle inequality
Checking 14950 quadruples
Checking 1001 quadruples
Checking 330 quadruples
Checking 70 quadruples
Checking 1 quadruples
Stress Formula 1                          : 0.061
Stress Formula 2                          : 0.399
R-squared(Monotonic)                      : 0.841
R-squared (Present Value Annuity Factor) : 0.788
```

| Node | Length | Child |
|---|---|---|
| 1 | 23.396 | A |
| 2 | 15.396 | B |
| 3 | 14.813 | C |
| 4 | 13.313 | D |
| 5 | 24.125 | E |
| 6 | 34.837 | F |
| 7 | 15.917 | G |
| 8 | 27.875 | H |
| 9 | 25.604 | I |
| 10 | 19.833 | J |
| 11 | 13.688 | K |
| 12 | 28.620 | L |
| 13 | 21.813 | M |
| 14 | 22.188 | N |
| 15 | 19.083 | O |
| 16 | 14.167 | P |
| 17 | 18.958 | Q |
| 18 | 21.438 | R |
| 19 | 28.000 | S |
| 20 | 23.875 | T |
| 21 | 23.000 | U |
| 22 | 27.125 | V |
| 23 | 21.563 | W |
| 24 | 14.604 | X |
| 25 | 17.188 | Y |
| 26 | 18.042 | Z |
| 27 | 16.943 | 1,9 |
| 28 | 15.380 | 2,24 |

```
29   15.716   3,25
30   19.583   4,11
31   26.063   5,20
32   23.843   7,15
33    6.114   8,22
34   17.175   10,16
35   18.807   13,14
36   13.784   17,26
37   15.663   18,23
38    8.886   19,21
39    4.562   27,35
40    1.700   29,36
41    8.799   33,38
42    4.180   39,31
43    1.123   12,28
44    5.049   34,40
45    2.467   42,41
46    4.585   30,43
47    2.616   32,44
48    2.730   6,37
49    0.000   45,48
50    3.864   46,47
51    0.000   50,49
```

(SYSTAT also displays the raw data, as well as the model distances.)



# *Computation*

## *Algorithms*

JOIN follows the standard hierarchical amalgamation method described in Hartigan (1975). The algorithm in Gruvaeus and Wainer (1972) is used to order the tree. The $K^{th}$-Nearest Neighborhood method and the Uniform Kernel method use the algorithm prescribed in Wong and Lane (1983).

KMEANS follows the algorithm described in Hartigan (1975). Its speed can be improved using modifications proposed by Hartigan and Wong (1979). There is an important difference between SYSTAT's KMEANS algorithm and implementations of Hartigan's algorithm in BMDP, SAS, and SPSS: in SYSTAT, by default, seeds for new clusters are chosen by finding the case farthest from the centroid of its cluster; in Hartigan's algorithm, seeds forming new clusters are chosen by splitting on the variable with largest variance. KMEDIANS essentially follows the same algorithm but uses the median instead of the mean. The median is determined by a modification of binary search.

## *Missing Data*

In cluster analysis, all distances are computed with pairwise deletion of missing values. Since missing data are excluded from distance calculations by pairwise deletion, they do not directly influence clustering when you use the MATRIX option for JOIN. To use the MATRIX display to analyze patterns of missing data, create a new file in which missing values are recoded to 1, and all other values to 0. Then use JOIN with MATRIX to see whether missing values cluster together in a systematic pattern.

# *References*

Bezdek, J.C and Pal, N. R. (1998). Some new indexes of cluster validity. *IEEE Trans. Systems, Man and Cybernetics*, *Part B: Cybernetics*, 28, 301-315.

Calinski, T., Harabasz, J. (1974): A dendrite method for cluster analysis. *Communications in Statistics*, 3, 1-27.

* Campbell, D. T. and Fiske, D. W. (1959). Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological Bulletin*, 56, 81–105.

Davies, D.L. and Bouldin, D.W. (1979). A cluster separation measure. *IEEE Trans. Pattern Anal. Machine Intell.*, 1, 4, 224-227.

Dunn, J.C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, *Journal of Cybernetics*, 3, 32-57.

Fisher, L. and Van Ness, J. W. (1971). Admissible clustering procedures. *Biometrika*, 58, 91–104.

* Gower, J. C. (1967). A comparison of some methods of cluster analysis. *Biometrics,* 23, 623–637.

Gruvaeus, G. and Wainer, H. (1972). Two additions to hierarchical cluster analysis. *The British Journal of Mathematical and Statistical Psychology*, 25, 200–206.

* Guttman, L. (1944). A basis for scaling qualitative data. *American Sociological Review*, 139–150.

Hartigan, J. A. (1975). *Clustering algorithms*. New York: John Wiley & Sons.

Hartigan, J.A. and Wong, M. A. (1979), A K-Means Clustering Algorithm. *Applied Statistics*, 28, 100-108.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika,* 32, 241–254.

* Lance, G.N., and Williams, W.T. (1967), A general theory of classificatory sorting strategies, I. Hierarchical Systems, *Computer Journal*, 9, 373-380.

Ling, R. F. (1973). A computer generated aid for cluster analysis. *Communications of the ACM,* 16, 355–361.

* MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *5th Berkeley symposium on mathematics, statistics, and probability*, 1, 281–298.

McQuitty, L. L. (1960). Hierarchical syndrome analysis. *Educational and Psychological Measurement*, 20, 293–303.

Milligan, G. W. (1980). An examination of the effects of six types of error perturbation on fifteen clustering algorithms. *Psychometrika,* 45, 325–342.

* Milligan, G.W. (1987), A study of beta-flexible clustering method, *College of Administrative Science Working Paper Series*, 87-61 Columbus, OH: The Ohio State University.

Milligan, G.W. and Cooper, M. C. (1985). An examination of procedures for determining number of clusters in a data set. *Psychometrika,* 50, 159-179.

* Preparata, G. and Shamos, M. (1985). *Computational geometry: An introduction*. New York: Springer-Verlag.

Rothkopf, E. Z. (1957). A measure of stimulus similarity and errors in some paired associate learning tasks. *Journal of Experimental Psychology*, 53, 94-101.

Sattath, S. and Tversky, A. (1977). Additive similarity trees. *Psychometrika*, 42, 319–345.

Sharma, S.C. (1995). *Applied multivariate techniques*. New York: John Wiley & Sons.

Silverman, B.W. (1986), *Density estimation*, New York: Chapman & Hall.

* Sokal, R. R. and Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin,* 38, 1409–1438.

Sokal, R. R. and Sneath, P. H. A. (1963). *Principles of numerical taxonomy*. San Francisco: W. H. Freeman and Company.

* Vizirgiannis, M., Haldiki, M. and Gunopulos, D. (2003). *Uncertainity handling and quality assessment in data mining*. London: Springer-Varlag.

Wainer, H. and Schacht, S. (1978). Gappint. *Psychometrika,* 43, 203–212.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association,* 58, 236–244.

Wilkinson, L. (1979). Permuting a matrix to a simple structure. *Proceedings of the American Statistical Association,* 409–412.

Wong, M.A. and Lane, T. (1983), A kth nearest neighbor clustering procedure, *Journal of Royal Statistical Society*, Series B, 45 362-368.

(*indicates additional references.)