SAS/STAT® 14.2 User's Guide
Introduction to Clustering Procedures

This document is an individual chapter from *SAS/STAT® 14.2 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS/STAT® 14.2 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 14.2 User's Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2016

Chapter 11
Introduction to Clustering Procedures

Contents

Overview: Clustering Procedures

You can use SAS clustering procedures to cluster the observations or the variables in a SAS data set. Both hierarchical and disjoint clusters can be obtained. Only numeric variables can be analyzed directly by the procedures, although the DISTANCE procedure can compute a distance matrix that uses character or numeric variables.

The purpose of cluster analysis is to place objects into groups, or clusters, suggested by the data, not defined a priori, such that objects in a given cluster tend to be similar to each other in some sense, and objects in different clusters tend to be dissimilar. You can also use cluster analysis to summarize data rather than to find "natural" or "real" clusters; this use of clustering is sometimes called *dissection* (Everitt 1980).

Any generalization about cluster analysis must be vague because a vast number of clustering methods have been developed in several different fields, with different definitions of clusters and similarity among objects. The variety of clustering techniques is reflected by the variety of terms used for cluster analysis: botryology, classification, clumping, competitive learning, morphometrics, nosography, nosology, numerical taxonomy, partitioning, Q-analysis, systematics, taximetrics, taxonorics, typology, unsupervised pattern recognition, vector quantization, and winner-take-all learning. Good (1977) has also suggested aciniformics and agminatics.

Several types of clusters are possible:

- Disjoint clusters place each object in one and only one cluster.
- Hierarchical clusters are organized so that one cluster can be entirely contained within another cluster, but no other kind of overlap between clusters is allowed.
- Overlapping clusters can be constrained to limit the number of objects that belong simultaneously to two clusters, or they can be unconstrained, allowing any degree of overlap in cluster membership.
- Fuzzy clusters are defined by a probability or grade of membership of each object in each cluster. Fuzzy clusters can be disjoint, hierarchical, or overlapping.

The data representations of objects to be clustered also take many forms. The most common are as follows:

- a square distance or similarity matrix, in which both rows and columns correspond to the objects to be clustered. A correlation matrix is an example of a similarity matrix.
- a coordinate matrix, in which the rows are observations and the columns are variables, as in the usual SAS multivariate data set. The observations, the variables, or both can be clustered.

The SAS procedures for clustering are oriented toward disjoint or hierarchical clusters from coordinate data, distance data, or a correlation or covariance matrix. The following procedures are used for clustering:

CLUSTER	performs hierarchical clustering of observations by using eleven agglomerative methods applied to coordinate data or distance data and draws tree diagrams, which are also called *dendrograms* or *phenograms*.
FASTCLUS	finds disjoint clusters of observations by using a *k*-means method applied to coordinate data. PROC FASTCLUS is especially suitable for large data sets.
MODECLUS	finds disjoint clusters of observations with coordinate or distance data by using nonparametric density estimation. It can also perform approximate nonparametric significance tests for the number of clusters.
VARCLUS	performs both hierarchical and disjoint clustering of variables by using oblique multiple-group component analysis and draws tree diagrams, which are also called *dendrograms* or *phenograms*.
TREE	draws tree diagrams, also called *dendrograms* or *phenograms*, by using output from the CLUSTER or VARCLUS procedure. PROC TREE can also create a data set indicating cluster membership at any specified level of the cluster tree.

The following procedures are useful for processing data prior to the actual cluster analysis:

ACECLUS	attempts to estimate the pooled within-cluster covariance matrix from coordinate data without knowledge of the number or the membership of the clusters (Art, Gnanadesikan, and Kettenring 1982). PROC ACECLUS outputs a data set containing canonical variable scores to be used in the cluster analysis proper.

DISTANCE	computes various measures of distance, dissimilarity, or similarity between the observations (rows) of a SAS data set. PROC DISTANCE also provides various nonparametric and parametric methods for standardizing variables. Different variables can be standardized with different methods.
PRINCOMP	performs a principal component analysis and outputs principal component scores.
STDIZE	standardizes variables by using any of a variety of location and scale measures, including mean and standard deviation, minimum and range, median and absolute deviation from the median, various M-estimators and A-estimators, and some scale estimators designed specifically for cluster analysis.

Massart and Kaufman (1983) is the best elementary introduction to cluster analysis. Other important texts are Anderberg (1973); Sneath and Sokal (1973); Duran and Odell (1974); Hartigan (1975); Titterington, Smith, and Makov (1985); McLachlan and Basford (1988); Kaufman and Rousseeuw (1990). Hartigan (1975); Spath (1980) give numerous FORTRAN programs for clustering. Any prospective user of cluster analysis should study the Monte Carlo results of Milligan (1980); Milligan and Cooper (1985); Cooper and Milligan (1988). Important references on the statistical aspects of clustering include MacQueen (1967); Wolfe (1970); Scott and Symons (1971); Hartigan (1977, 1978, 1981, 1985a); Symons (1981); Everitt (1981); Sarle (1983); Bock (1985); Thode, Mendell, and Finch (1988). Bayesian methods have important advantages over maximum likelihood; see Binder (1978, 1981); Banfield and Raftery (1993); Bensmail et al. (1997). For fuzzy clustering, see Bezdek (1981); Bezdek and Pal (1992). The signal-processing perspective is provided by Gersho and Gray (1992). For a discussion of the fragmented state of the literature on cluster analysis, see Blashfield and Aldenderfer (1978).

Clustering Variables

Factor rotation is often used to cluster variables, but the resulting clusters are fuzzy. It is preferable to use PROC VARCLUS if you want hard (nonfuzzy), disjoint clusters. Factor rotation is better if you want to be able to find overlapping clusters. It is often a good idea to try both PROC VARCLUS and PROC FACTOR with an oblique rotation, compare the amount of variance explained by each, and see how fuzzy the factor loadings are and whether there seem to be overlapping clusters.

You can use PROC VARCLUS to harden a fuzzy factor rotation; use PROC FACTOR to create an output data set containing scoring coefficients and initialize PROC VARCLUS with this data set as follows:

```
proc factor rotate=promax score outstat=fact;
run;

proc varclus initial=input proportion=0;
run;
```

You can use any rotation method instead of the PROMAX method. The SCORE and OUTSTAT= options are necessary in the PROC FACTOR statement. PROC VARCLUS reads the correlation matrix from the data set created by PROC FACTOR. The INITIAL=INPUT option tells PROC VARCLUS to read initial scoring coefficients from the data set. The option PROPORTION=0 keeps PROC VARCLUS from splitting any of the clusters.

Clustering Observations

PROC CLUSTER is easier to use than PROC FASTCLUS because one run produces results from one cluster up to as many as you like. You must run PROC FASTCLUS once for each number of clusters.

The time required by PROC FASTCLUS is roughly proportional to the number of observations, whereas the time required by PROC CLUSTER with most methods varies with the square or cube of the number of observations. Therefore, you can use PROC FASTCLUS with much larger data sets than PROC CLUSTER.

If you want to hierarchically cluster a data set that is too large to use with PROC CLUSTER directly, you can have PROC FASTCLUS produce, for example, 50 clusters, and let PROC CLUSTER analyze these 50 clusters instead of the entire data set. The MEAN= data set produced by PROC FASTCLUS contains two special variables:

- The variable _FREQ_ gives the number of observations in the cluster.
- The variable _RMSSTD_ gives the root mean square across variables of the cluster standard deviations.

These variables are automatically used by PROC CLUSTER to give the correct results when clustering clusters. For example, you could specify Ward's minimum variance method Ward (1963):

```
proc fastclus maxclusters=50 mean=temp;
   var x y z;
run;

ods graphics on;
proc cluster method=ward outtree=tree;
   var x y z;
run;
```

Or you could specify Wong's hybrid method (Wong 1982):

```
proc fastclus maxclusters=50 mean=temp;
   var x y z;
run;

ods graphics on;
proc cluster method=density hybrid outtree=tree;
   var x y z;
run;
```

More detailed examples are given in Chapter 34, "The CLUSTER Procedure."

Characteristics of Methods for Clustering Observations

Many simulation studies comparing various methods of cluster analysis have been performed. In these studies, artificial data sets containing known clusters are produced using pseudo-random-number generators. The data sets are analyzed by a variety of clustering methods, and the degree to which each clustering method recovers the known cluster structure is evaluated. See Milligan (1981) for a review of such studies. In most of these studies, the clustering method with the best overall performance has been either average linkage or Ward's minimum variance method. The method with the poorest overall performance has almost invariably been single linkage. However, in many respects, the results of simulation studies are inconsistent and confusing.

When you attempt to evaluate clustering methods, it is essential to realize that most methods are biased toward finding clusters possessing certain characteristics related to size (number of members), shape, or dispersion. Methods based on the least squares criterion (Sarle 1982), such as *k*-means and Ward's minimum variance method, tend to find clusters with roughly the same number of observations in each cluster. Average linkage is somewhat biased toward finding clusters of equal variance. Many clustering methods tend to produce compact, roughly hyperspherical clusters and are incapable of detecting clusters with highly elongated or irregular shapes. The methods with the least bias are those based on nonparametric density estimation such as single linkage and density linkage.

Most simulation studies have generated compact (often multivariate normal) clusters of roughly equal size or dispersion. Such studies naturally favor average linkage and Ward's method over most other hierarchical methods, especially single linkage. It would be easy, however, to design a study that uses elongated or irregular clusters in which single linkage would perform much better than average linkage or Ward's method (see some of the following examples). Even studies that compare clustering methods that use "realistic" data might unfairly favor particular methods. For example, in all the data sets used by Mezzich and Solomon (1980), the clusters established by field experts are of equal size. When interpreting simulation or other comparative studies, you must, therefore, decide whether the artificially generated clusters in the study resemble the clusters you suspect might exist in your data in terms of size, shape, and dispersion. If, like many people doing exploratory cluster analysis, you have no idea what kinds of clusters to expect, you should include at least one of the relatively unbiased methods, such as density linkage, in your analysis.

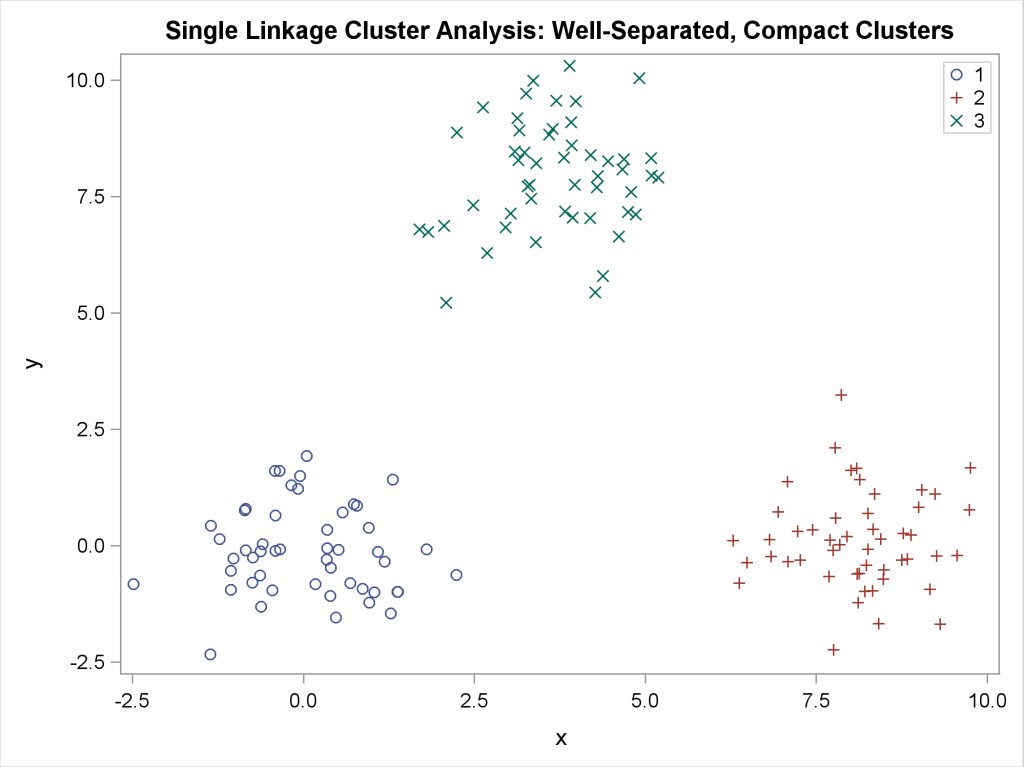
The rest of this section consists of a series of examples that illustrate the performance of various clustering methods under various conditions. The first, and simplest, example shows a case of well-separated clusters. The other examples show cases of poorly separated clusters, clusters of unequal size, parallel elongated clusters, and nonconvex clusters.

Well-Separated Clusters

If the population clusters are sufficiently well separated, almost any clustering method performs well, as demonstrated in the following example, which uses single linkage. In this and subsequent examples, the output from the clustering procedures is not shown, but cluster membership is displayed in scatter plots. The following SAS statements produce Figure 11.1:

```
data compact;
   keep x y;
   n=50; scale=1;
   mx=0; my=0; link generate;
   mx=8; my=0; link generate;
   mx=4; my=8; link generate;
   stop;
generate:
   do i=1 to n;
      x=rannor(1)*scale+mx;
      y=rannor(1)*scale+my;
      output;
   end;
   return;
run;

proc cluster data=compact outtree=tree method=single noprint;
run;

proc tree noprint out=out n=3;
   copy x y;
run;

ods graphics on / attrpriority=none;

proc sgplot noautolegend;
   title 'Single Linkage Cluster Analysis: '
         'Well-Separated, Compact Clusters';
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

Figure 11.1 Well-Separated, Compact Clusters: PROC CLUSTER METHOD=SINGLE



The ATTRPRIORITY=NONE option in the ODS GRAPHICS statement differentiates clusters by both marker shape and color. By default, ATTRPRIORITY=COLOR; so the default markers are circles and clusters are differentiated only by colors.

Poorly Separated Clusters

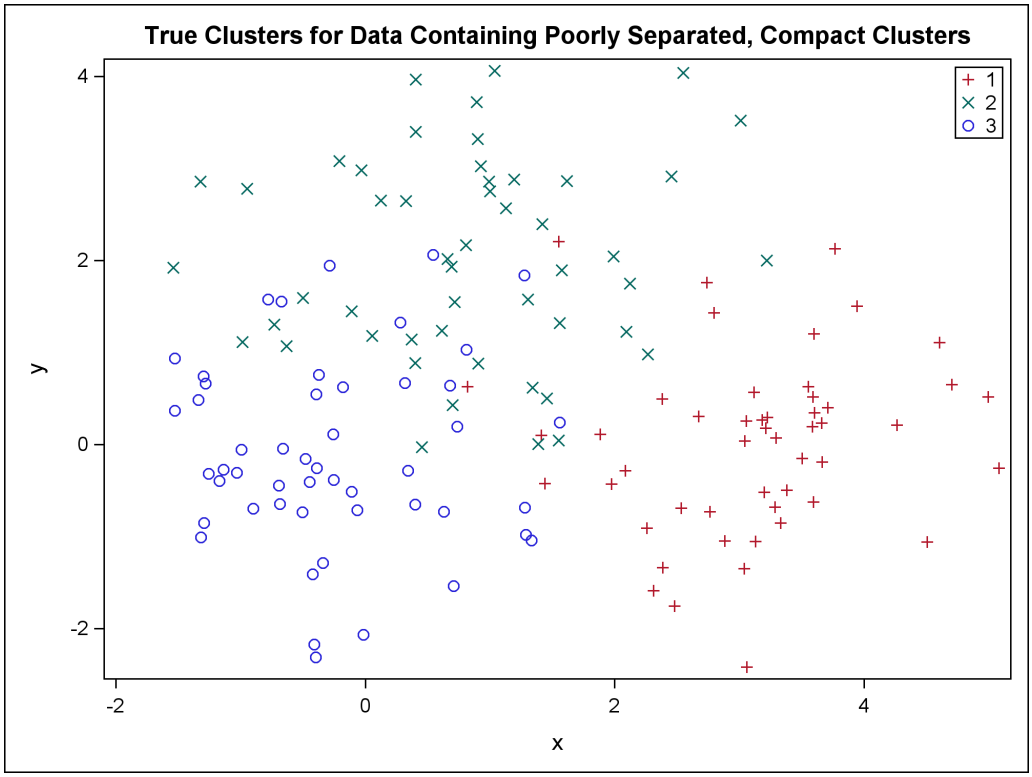
To see how various clustering methods differ, you must examine a more difficult problem than that of the previous example.

The following data set is similar to the first except that the three clusters are much closer together. This example demonstrates the use of PROC FASTCLUS and five hierarchical methods available in PROC CLUSTER. To help you compare methods, this example plots true, generated clusters. Also included is a bubble plot of the density estimates obtained in conjunction with two-stage density linkage in PROC CLUSTER.

The following SAS statements produce Figure 11.2:

```
data closer;
   keep x y c;
   n=50; scale=1;
   mx=0; my=0; c=3; link generate;
   mx=3; my=0; c=1; link generate;
   mx=1; my=2; c=2; link generate;
   stop;
generate:
   do i=1 to n;
      x=rannor(9)*scale+mx;
      y=rannor(9)*scale+my;
      output;
   end;
   return;
run;

title 'True Clusters for Data Containing Poorly Separated, Compact Clusters';
proc sgplot noautolegend;
   scatter y=y x=x / group=c;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

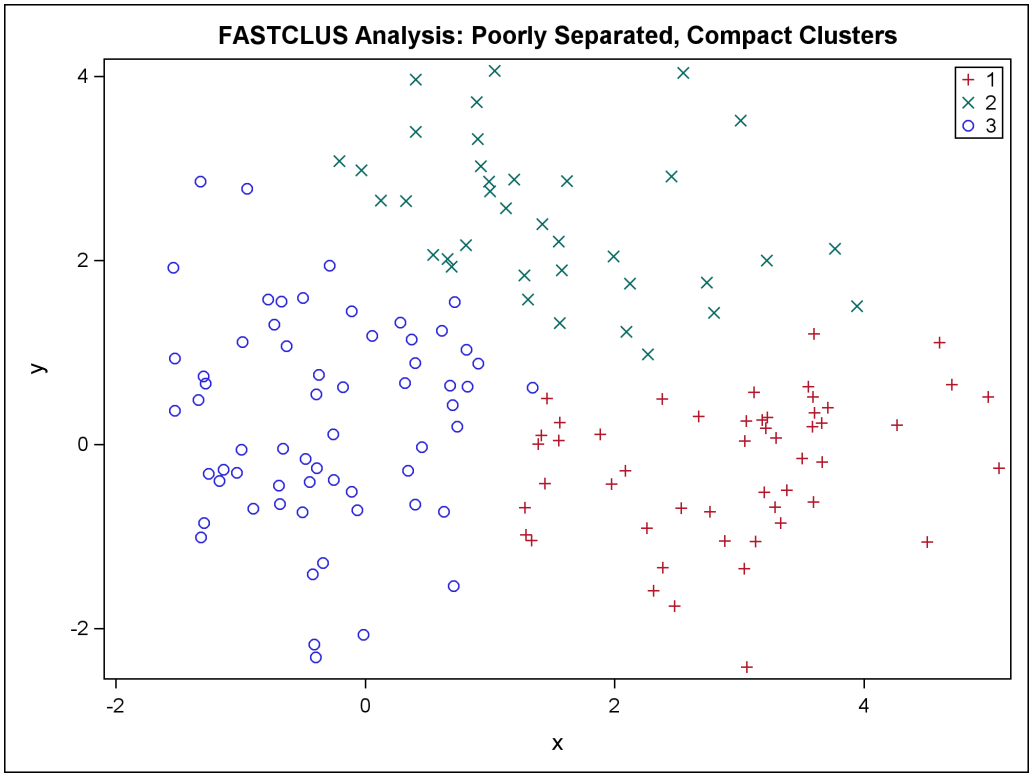
Figure 11.2 Poorly Separated, Compact Clusters: Plot of True Clusters



The following statements use the FASTCLUS procedure to find three clusters and then use the SGPLOT procedure to plot the clusters. The following statements produce Figure 11.3:

```
proc fastclus data=closer out=out maxc=3 noprint;
   var x y;
   title 'FASTCLUS Analysis: '
         'Poorly Separated, Compact Clusters';
run;

proc sgplot noautolegend;
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

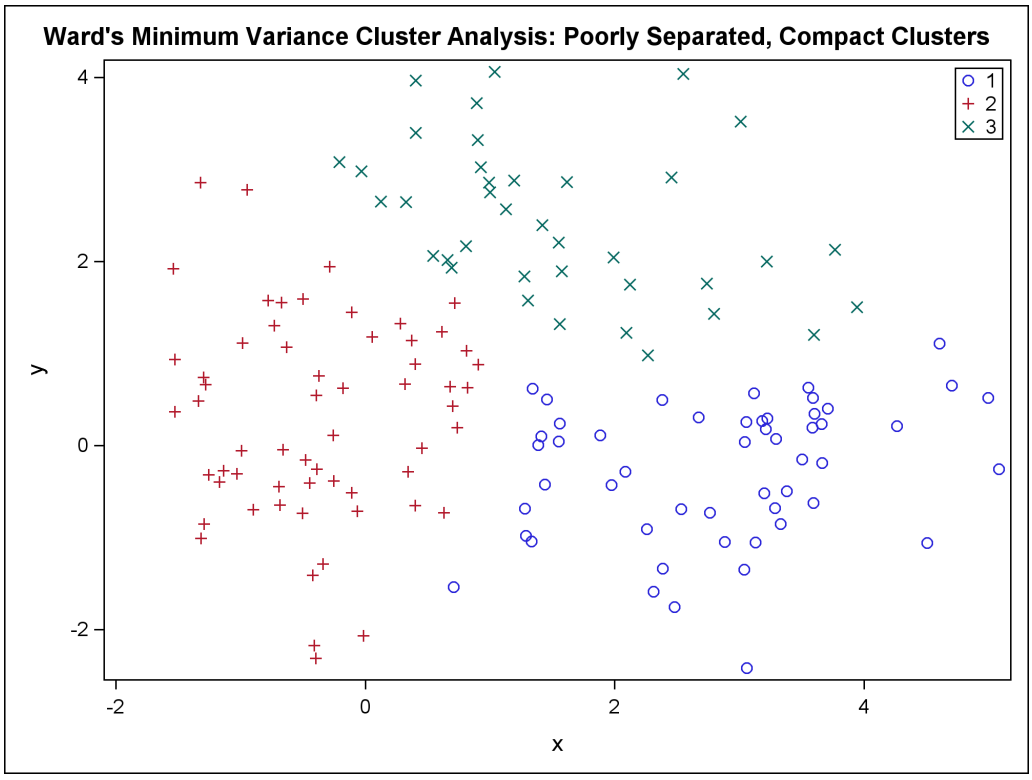
Figure 11.3 Poorly Separated, Compact Clusters: PROC FASTCLUS



The following SAS statements produce Figure 11.4:

```
proc cluster data=closer outtree=tree method=ward noprint;
   var x y;
run;

proc tree noprint out=out n=3;
   copy x y;
   title 'Ward''s Minimum Variance Cluster Analysis: '
         'Poorly Separated, Compact Clusters';
run;

proc sgplot noautolegend;
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

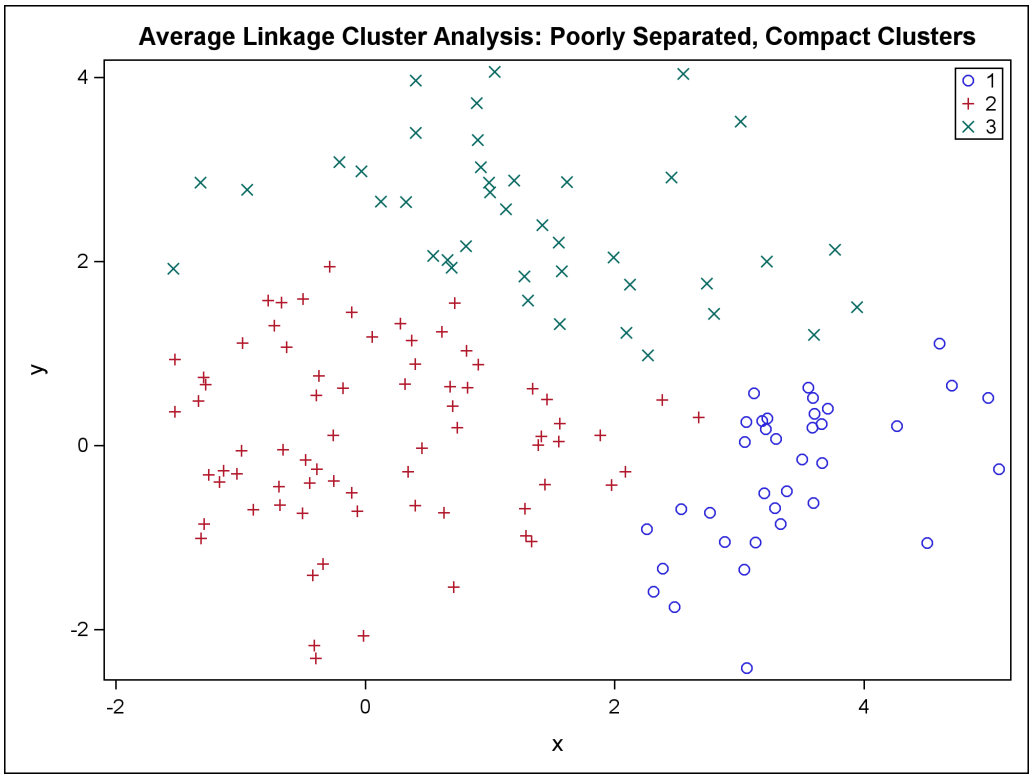
Figure 11.4 Poorly Separated, Compact Clusters: PROC CLUSTER METHOD=WARD



The following SAS statements produce Figure 11.5:

```
proc cluster data=closer outtree=tree method=average noprint;
   var x y;
run;

proc tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Average Linkage Cluster Analysis: '
         'Poorly Separated, Compact Clusters';
run;

proc sgplot noautolegend;
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

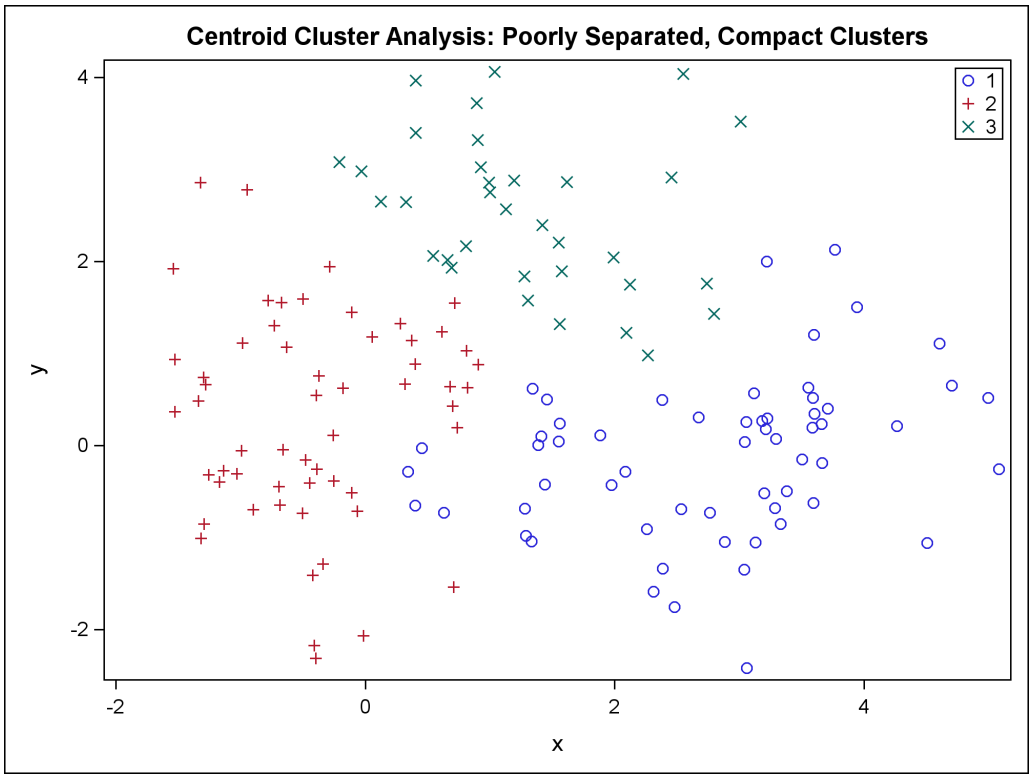
Figure 11.5 Poorly Separated, Compact Clusters: PROC CLUSTER METHOD=AVERAGE



The following SAS statements produce Figure 11.6:

```
proc cluster data=closer outtree=tree method=centroid noprint;
   var x y;
run;

proc tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Centroid Cluster Analysis: '
         'Poorly Separated, Compact Clusters';
run;

proc sgplot noautolegend;
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

Figure 11.6 Poorly Separated, Compact Clusters: PROC CLUSTER METHOD=CENTROID



The following SAS statements produce Figure 11.7 and Figure 11.8:

```
proc cluster data=closer outtree=tree method=twostage k=10 noprint;
   var x y;
run;

proc tree noprint out=out n=3;
   copy x y _dens_;
   title 'Two-Stage Density Linkage Cluster Analysis: '
         'Poorly Separated, Compact Clusters';
run;

proc sgplot noautolegend;
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;

proc sgplot noautolegend;
   title 'Estimated Densities for Data Containing Poorly Separated, '
         'Compact Clusters';
   bubble y=y x=x size=_dens_ / nofill lineattrs=graphdatadefault;
run;
```

Figure 11.7 Poorly Separated, Compact Clusters: PROC CLUSTER METHOD=TWOSTAGE

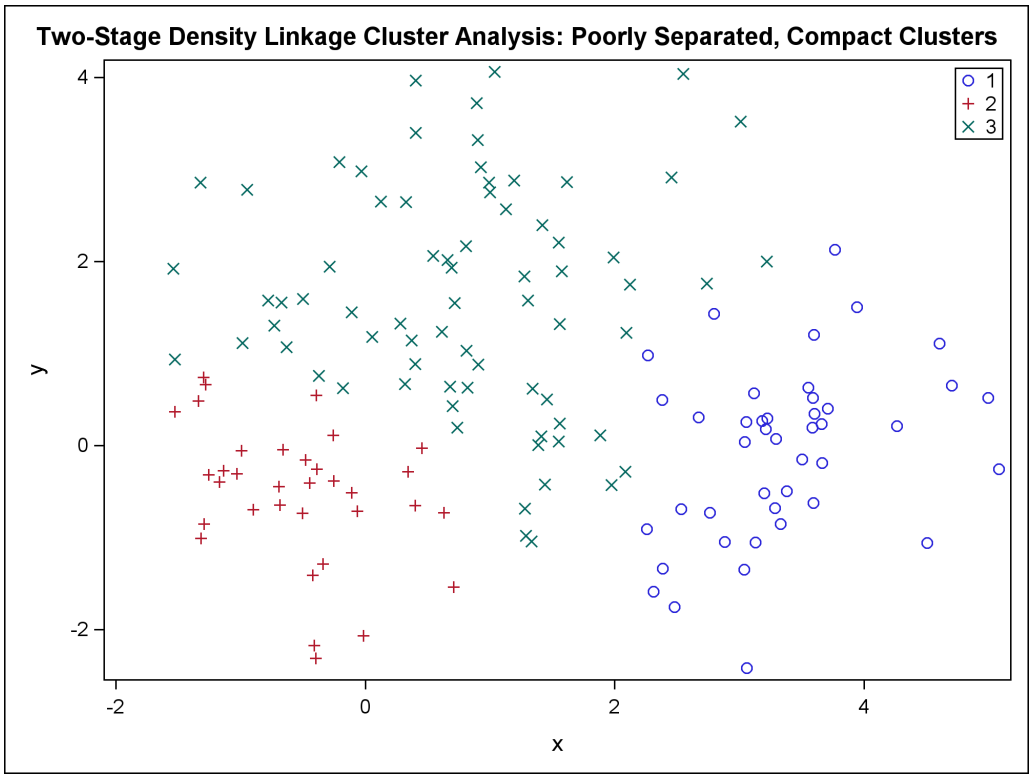
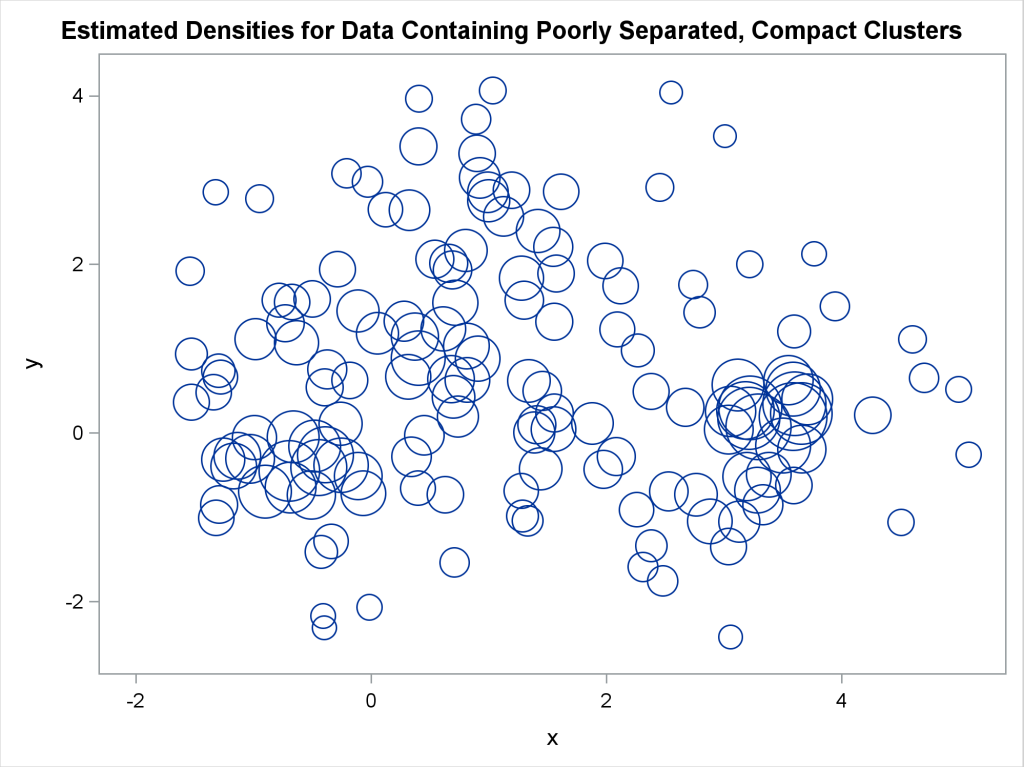


Figure 11.8 Poorly Separated, Compact Clusters: PROC CLUSTER METHOD=TWOSTAGE

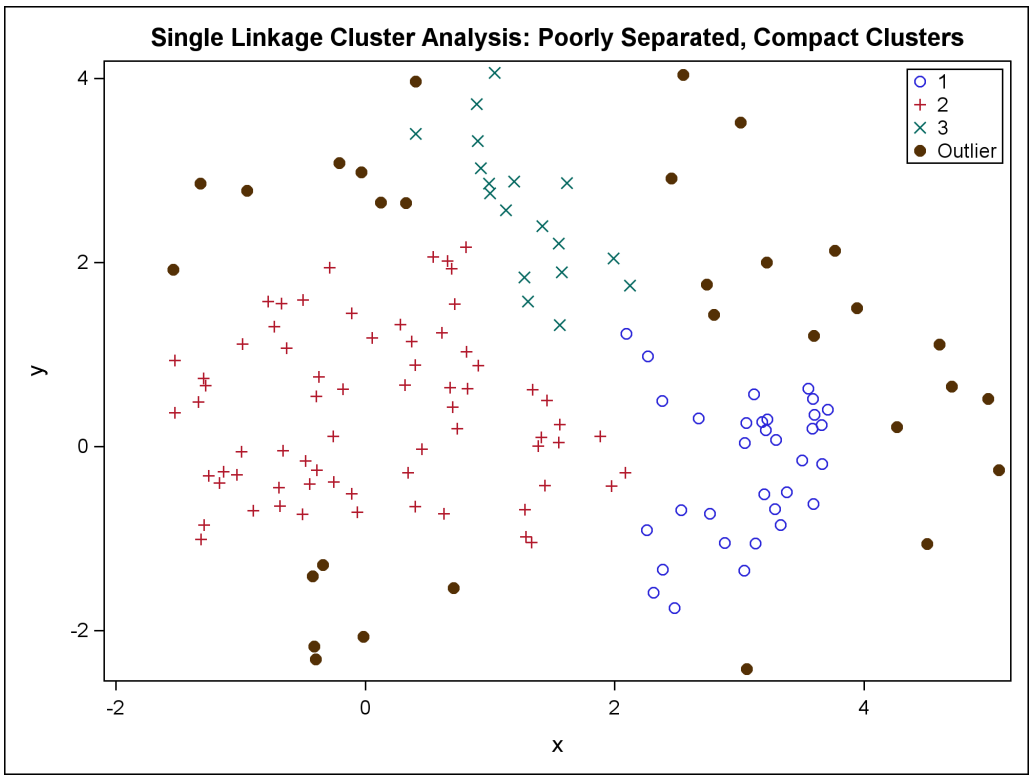


In two-stage density linkage, each cluster is a region surrounding a local maximum of the estimated probability density function. If you think of the estimated density function as a landscape with mountains and valleys, each mountain is a cluster, and the boundaries between clusters are placed near the bottoms of the valleys.

The following SAS statements produce Figure 11.9:

```
proc cluster data=closer outtree=tree method=single noprint;
   var x y;
run;

proc tree data=tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Single Linkage Cluster Analysis: '
         'Poorly Separated, Compact Clusters';
run;

proc format;
   value out . = 'Outlier';
run;

proc sgplot noautolegend;
   styleattrs datasymbols=(circle plus x circlefilled);
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
   format cluster out.;
run;
```

Figure 11.9 Poorly Separated, Compact Clusters: PROC CLUSTER METHOD=SINGLE



The two least squares methods, PROC FASTCLUS and Ward's, yield the most uniform cluster sizes and the best recovery of the true clusters. This result is expected since these two methods are biased toward recovering compact clusters of equal size. With average linkage, the lower-left cluster is too large; with the centroid method, the lower-right cluster is too large; and with two-stage density linkage, the top cluster is too large. The single linkage analysis resembles average linkage except for the large number of outliers resulting from the DOCK= option in the PROC TREE statement; the outliers (which are designated by missing values in the Cluster variable) are plotted as filled circles.

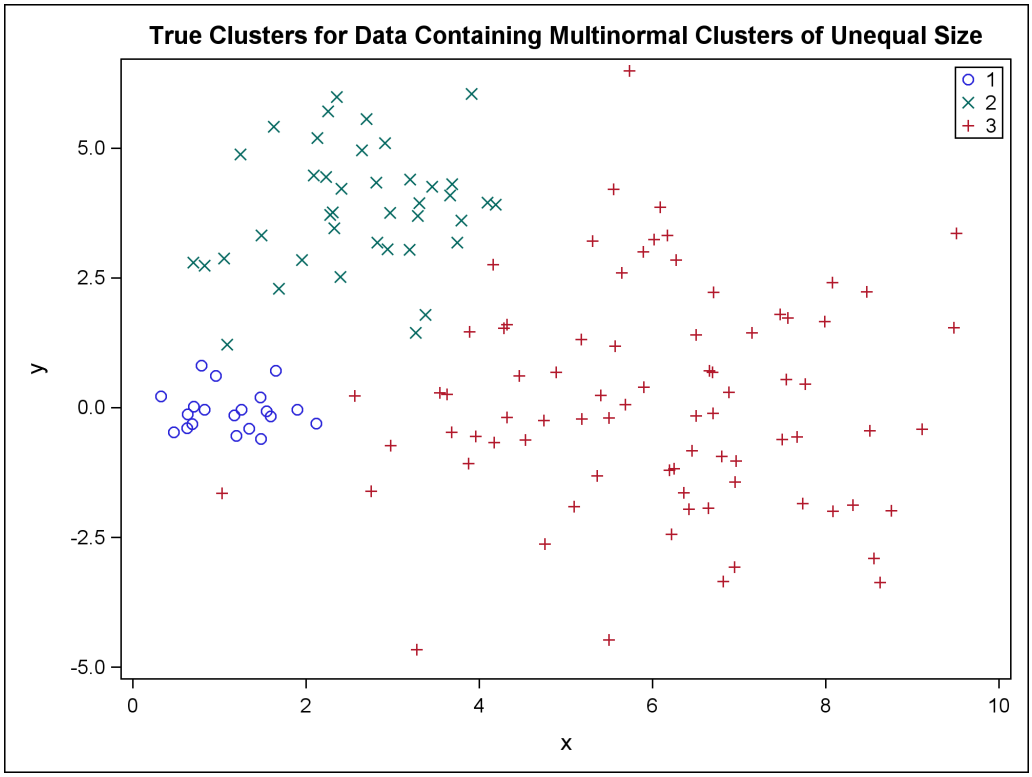
Multinormal Clusters of Unequal Size and Dispersion

In this example, there are three multinormal clusters that differ in size and dispersion. PROC FASTCLUS and five of the hierarchical methods available in PROC CLUSTER are used. To help you compare methods, the true, generated clusters are plotted.

The following SAS statements produce Figure 11.10:

```
data unequal;
   keep x y c;
   mx=1; my=0; n=20; scale=.5; c=1; link generate;
   mx=6; my=0; n=80; scale=2.; c=3; link generate;
   mx=3; my=4; n=40; scale=1.; c=2; link generate;
   stop;
generate:
   do i=1 to n;
      x=rannor(1)*scale+mx;
      y=rannor(1)*scale+my;
      output;
   end;
   return;
run;

title 'True Clusters for Data Containing Multinormal Clusters of Unequal Size';
proc sgplot noautolegend;
   scatter y=y x=x / group=c;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

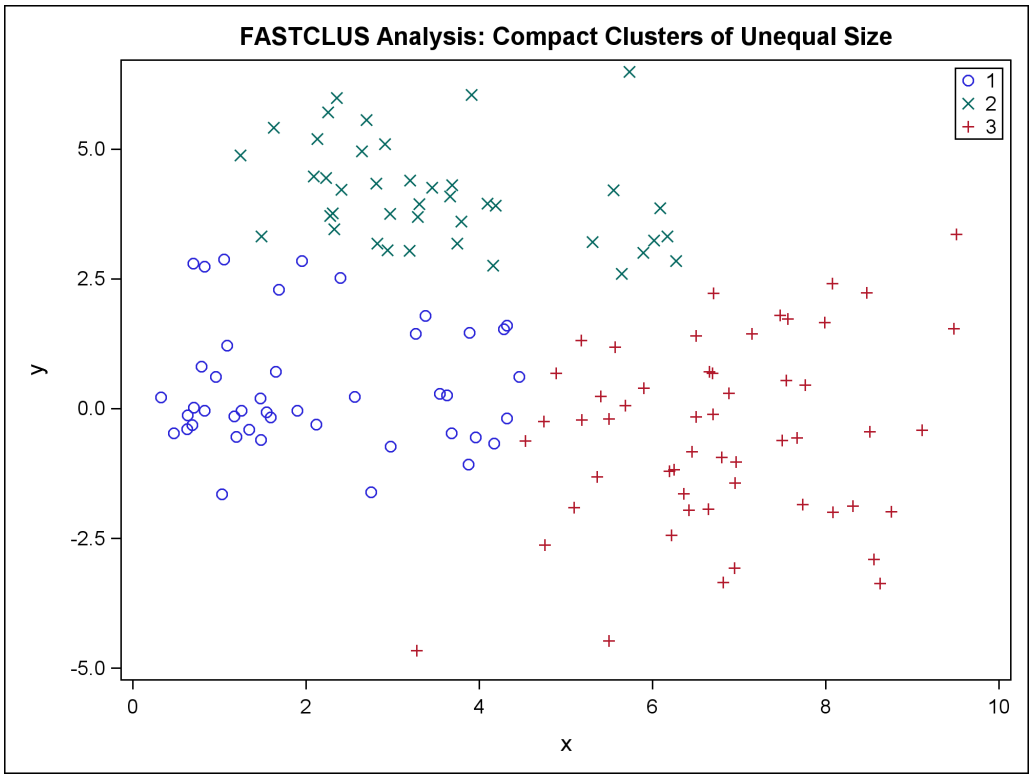
Figure 11.10 Generated Clusters of Unequal Size



The following statements use the FASTCLUS procedure to find three clusters and then use the SGPLOT procedure to plot the clusters. The following statements produce Figure 11.11:

```
proc fastclus data=unequal out=out maxc=3 noprint;
   var x y;
   title 'FASTCLUS Analysis: Compact Clusters of Unequal Size';
run;

proc sgplot noautolegend;
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

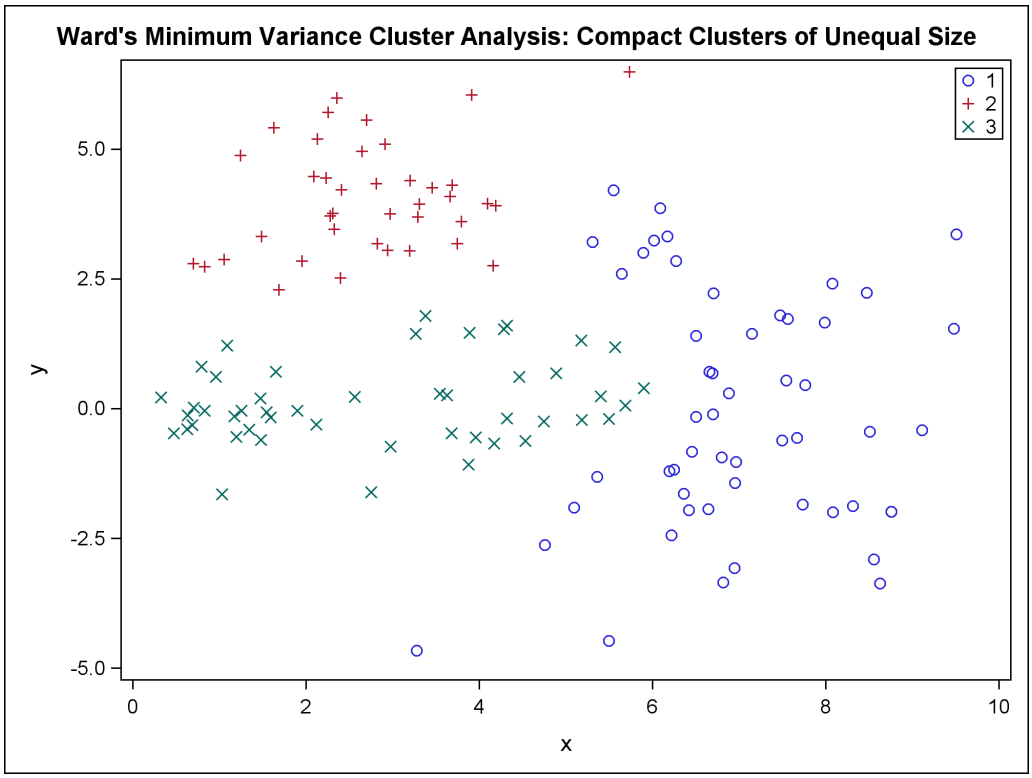
Figure 11.11 Compact Clusters of Unequal Size: PROC FASTCLUS



The following SAS statements produce Figure 11.12:

```
proc cluster data=unequal outtree=tree method=ward noprint;
   var x y;
run;

proc tree noprint out=out n=3;
   copy x y;
   title 'Ward''s Minimum Variance Cluster Analysis: '
         'Compact Clusters of Unequal Size';
run;

proc sgplot noautolegend;
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

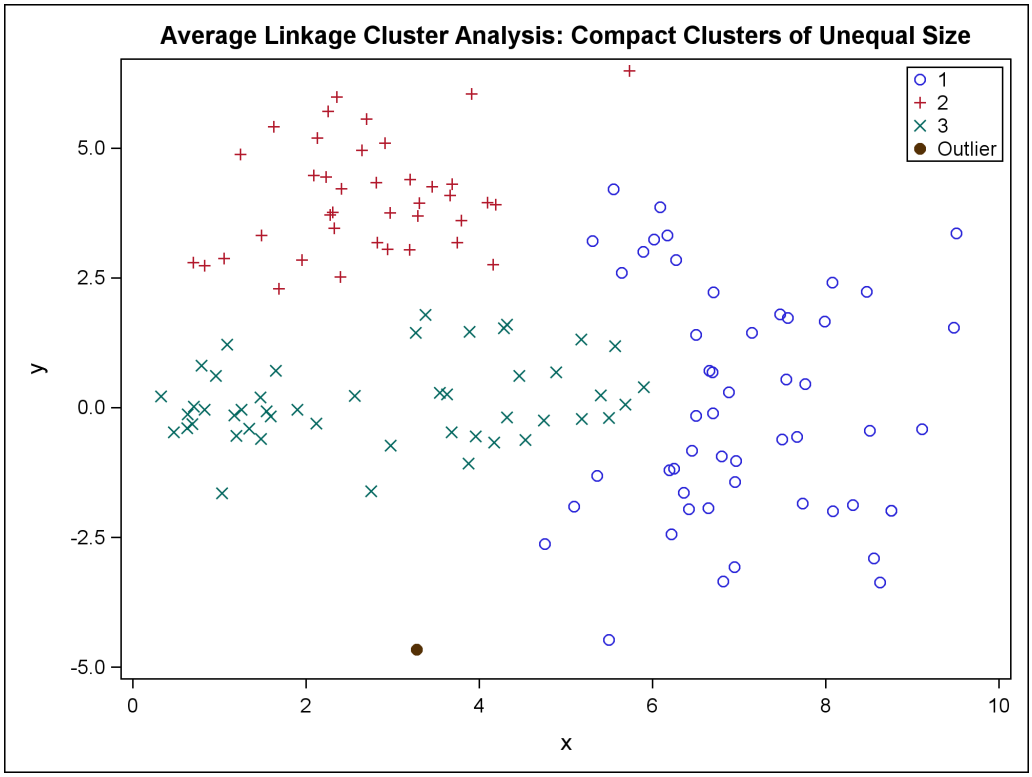
Figure 11.12 Compact Clusters of Unequal Size: PROC CLUSTER METHOD=WARD



The following SAS statements produce Figure 11.13:

```
proc cluster data=unequal outtree=tree method=average noprint;
   var x y;
run;

proc tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Average Linkage Cluster Analysis: '
         'Compact Clusters of Unequal Size';
run;

proc sgplot noautolegend;
   styleattrs datasymbols=(circle plus x circlefilled);
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
   format cluster out.;
run;
```

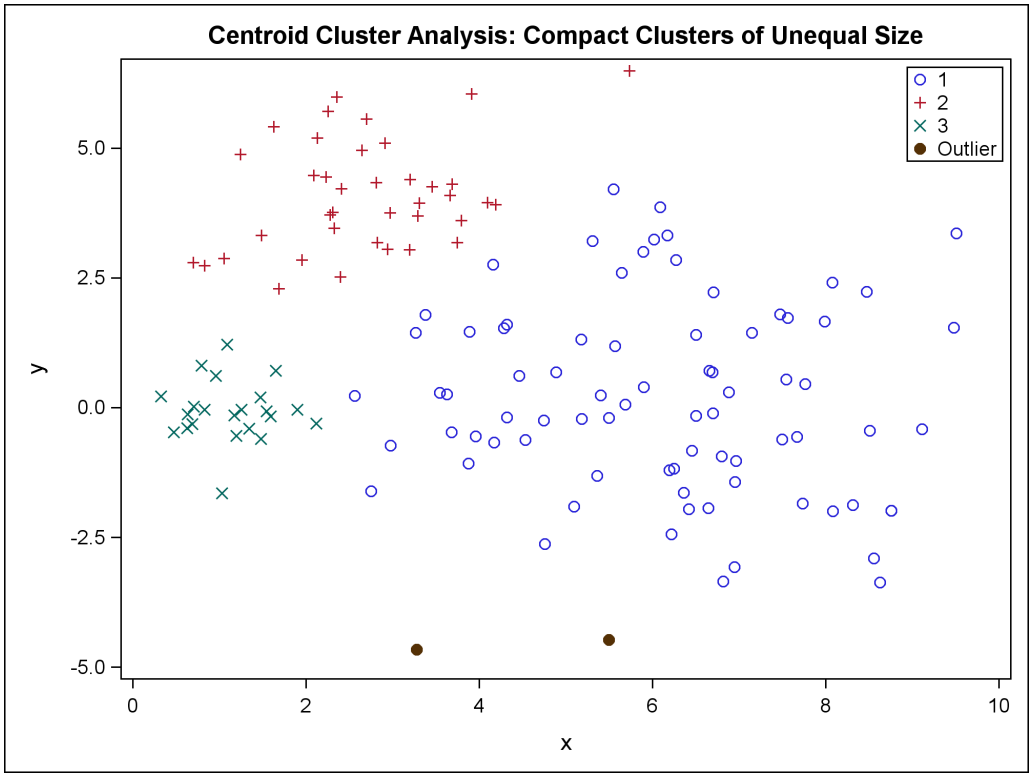
Figure 11.13 Compact Clusters of Unequal Size: PROC CLUSTER METHOD=AVERAGE



The following SAS statements produce Figure 11.14:

```
proc cluster data=unequal outtree=tree method=centroid noprint;
   var x y;
run;

proc tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Centroid Cluster Analysis: '
         'Compact Clusters of Unequal Size';
run;

proc sgplot noautolegend;
   styleattrs datasymbols=(circle plus x circlefilled);
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
   format cluster out.;
run;
```

Figure 11.14 Compact Clusters of Unequal Size: PROC CLUSTER METHOD=CENTROID



The following SAS statements produce Figure 11.15 and Figure 11.16:

```
proc cluster data=unequal outtree=tree method=twostage k=10 noprint;
   var x y;
run;

proc tree noprint out=out n=3;
   copy x y _dens_;
   title 'Two-Stage Density Linkage Cluster Analysis: '
         'Compact Clusters of Unequal Size';
run;

proc sgplot noautolegend;
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;

proc sgplot noautolegend;
   title 'Estimated Densities for Data Containing '
         'Compact Clusters of Unequal Size';
   bubble y=y x=x size=_dens_ / nofill lineattrs=graphdatadefault;
run;
```

Figure 11.15 Compact Clusters of Unequal Size: PROC CLUSTER METHOD=TWOSTAGE

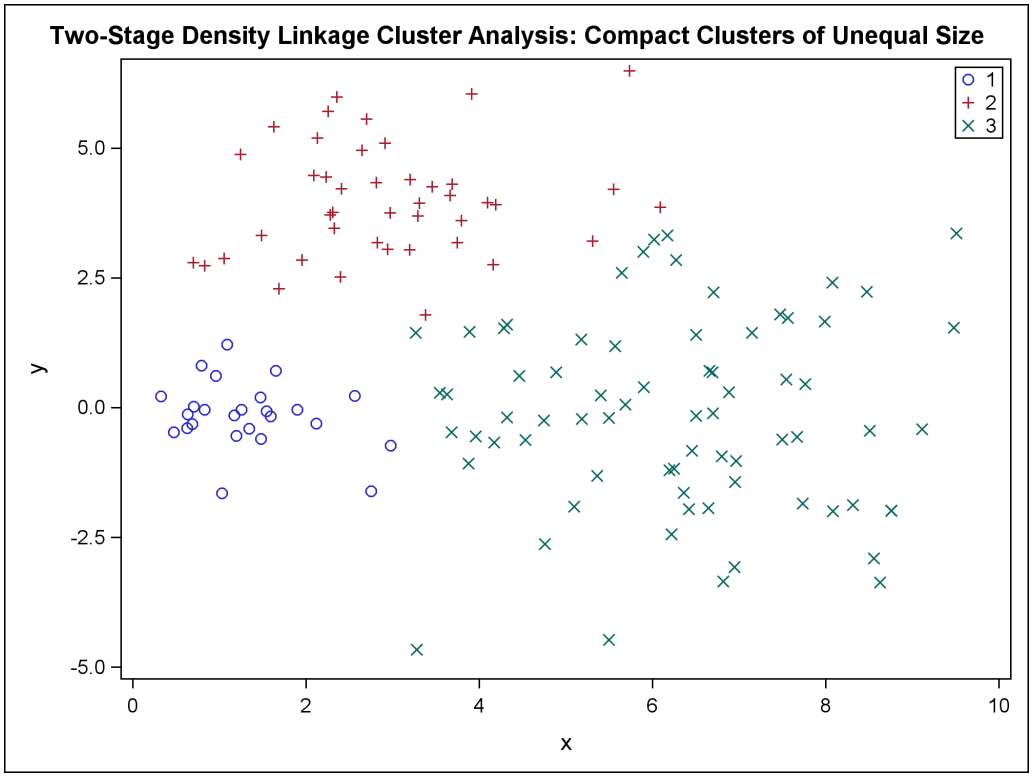
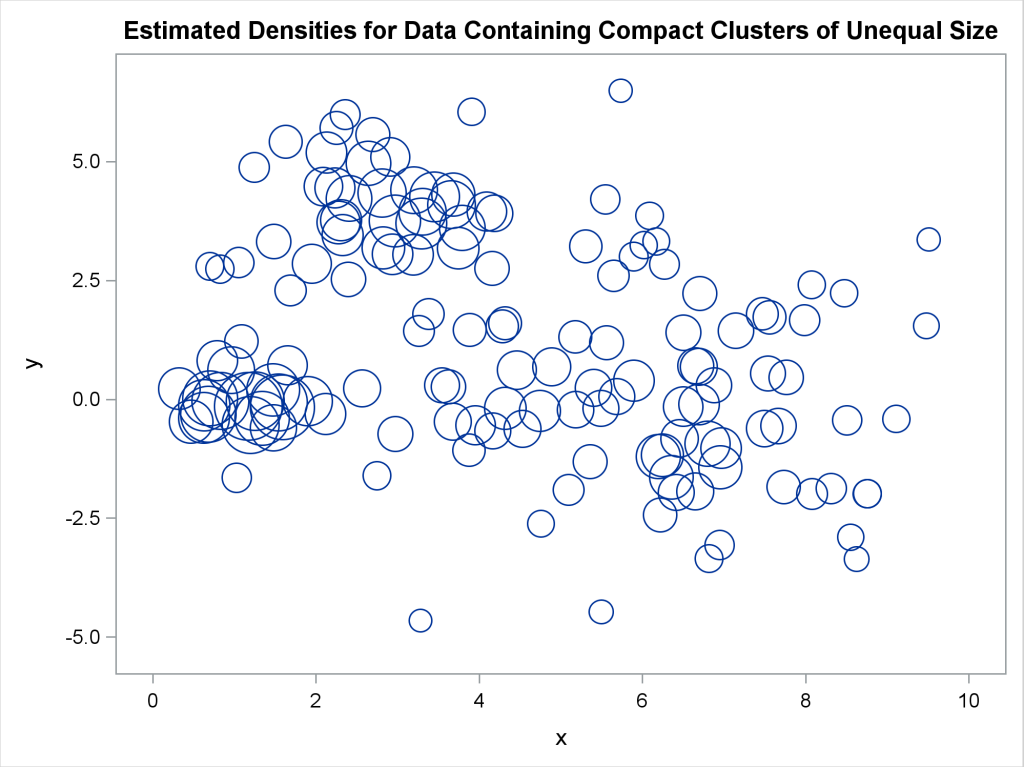


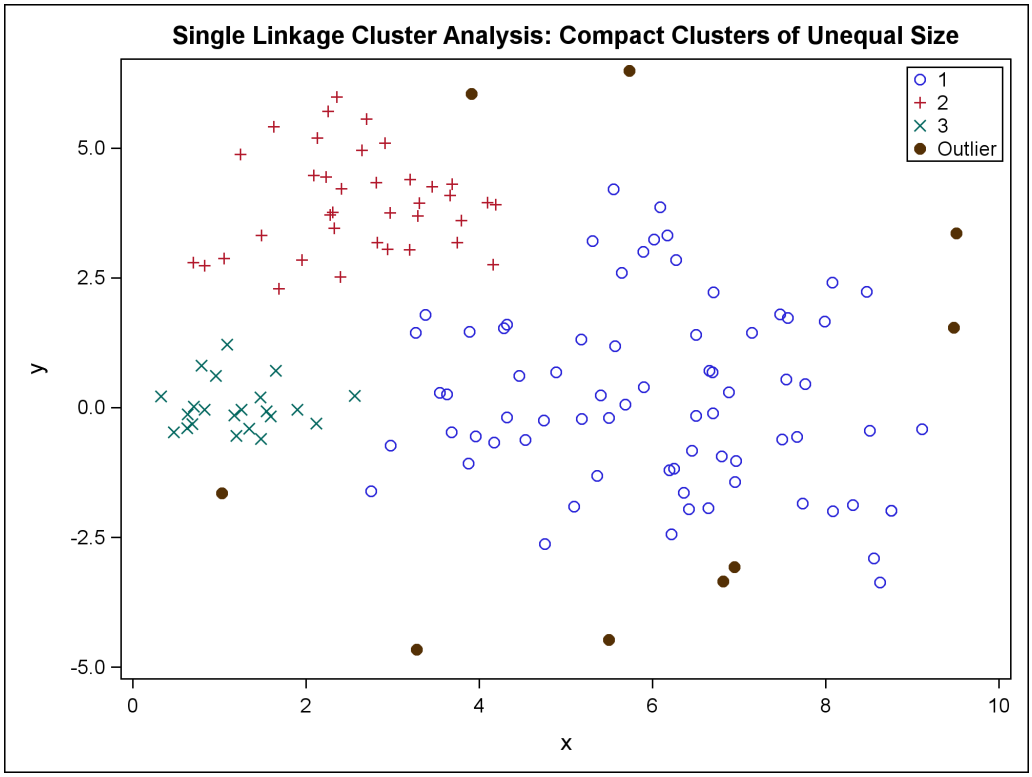
Figure 11.16 Compact Clusters of Unequal Size: PROC CLUSTER METHOD=TWOSTAGE



The following SAS statements produce Figure 11.17:

```
proc cluster data=unequal outtree=tree method=single noprint;
   var x y;
run;

proc tree data=tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Single Linkage Cluster Analysis: '
         'Compact Clusters of Unequal Size';
run;

proc sgplot noautolegend;
   styleattrs datasymbols=(circle plus x circlefilled);
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
   format cluster out.;
run;
```

Figure 11.17 Compact Clusters of Unequal Size: PROC CLUSTER METHOD=SINGLE



In the PROC FASTCLUS analysis, the smallest cluster, in the bottom-left portion of the plot, has stolen members from the other two clusters, and the upper-left cluster has also acquired some observations that rightfully belong to the larger, lower-right cluster. With Ward's method, the upper-left cluster is separated correctly, but the lower-left cluster has taken a large bite out of the lower-right cluster. For both of these methods, the clustering errors are in accord with the biases of the methods to produce clusters of equal size. In the average linkage analysis, both the upper-left and lower-left clusters have encroached on the lower-right cluster, thereby making the variances more nearly equal than in the true clusters. The centroid method, which lacks the size and dispersion biases of the previous methods, obtains an essentially correct partition.

Two-stage density linkage does almost as well, even though the compact shapes of these clusters favor the traditional methods. Single linkage also produces excellent results.

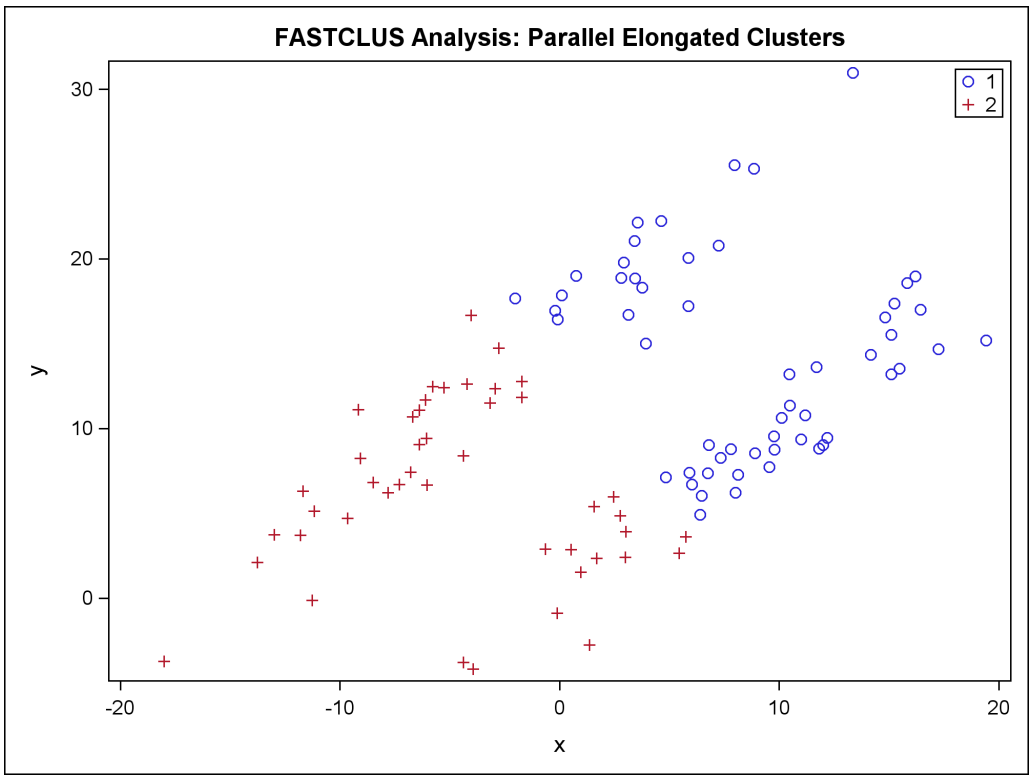
Elongated Multinormal Clusters

In this example, the data are sampled from two highly elongated multinormal distributions with equal covariance matrices. The following SAS statements produce Figure 11.18:

```
data elongate;
   keep x y;
   ma=8; mb=0; link generate;
   ma=6; mb=8; link generate;
   stop;
generate:
   do i=1 to 50;
      a=rannor(7)*6+ma;
      b=rannor(7)+mb;
      x=a-b;
      y=a+b;
      output;
   end;
   return;
run;

proc fastclus data=elongate out=out maxc=2 noprint;
run;

proc sgplot noautolegend;
   title 'FASTCLUS Analysis: Parallel Elongated Clusters';
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

Notice that PROC FASTCLUS found two clusters, as requested by the MAXC= option. However, it attempted to form spherical clusters, which are obviously inappropriate for these data.

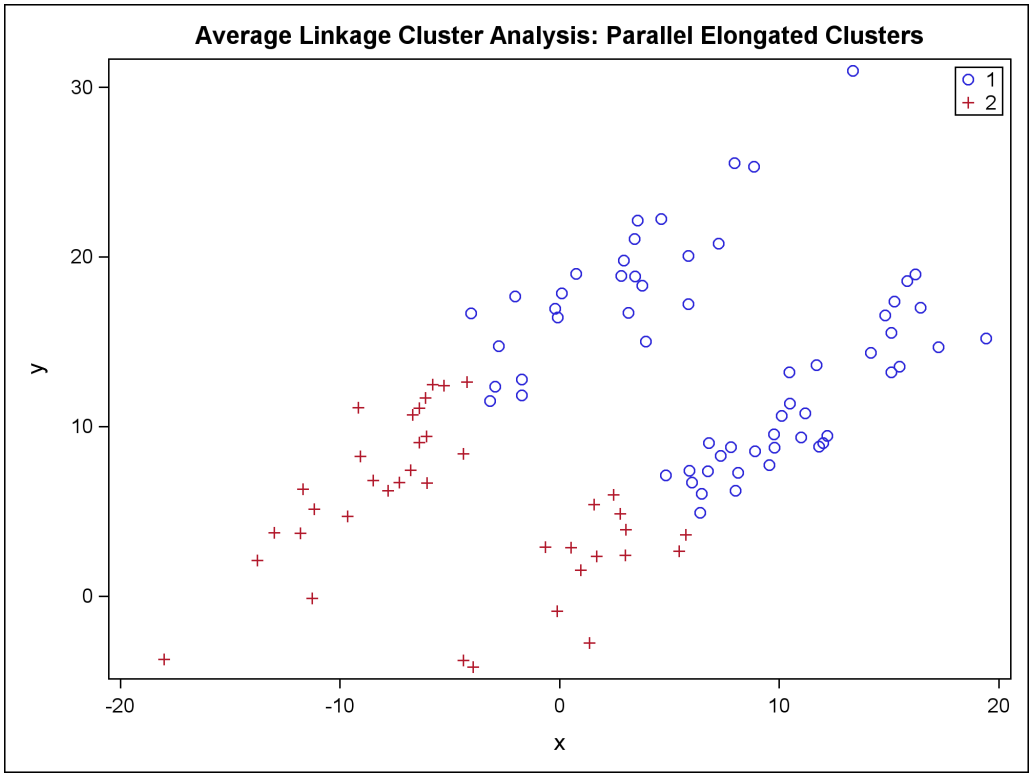
Figure 11.18 Parallel Elongated Clusters: PROC FASTCLUS



The following SAS statements produce Figure 11.19:

```
proc cluster data=elongate outtree=tree method=average noprint;
run;

proc tree noprint out=out n=2 dock=5;
   copy x y;
run;

proc sgplot noautolegend;
   title 'Average Linkage Cluster Analysis: '
         'Parallel Elongated Clusters';
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

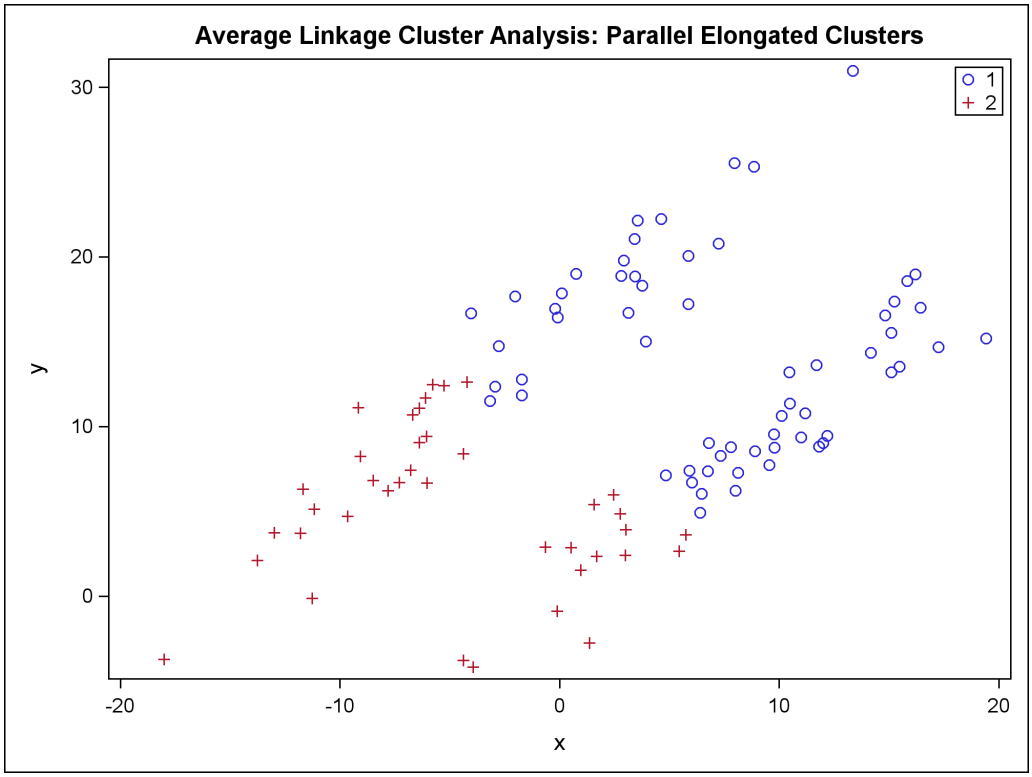
Figure 11.19 Parallel Elongated Clusters: PROC CLUSTER METHOD=AVERAGE



The following SAS statements produce Figure 11.20:

```
proc cluster data=elongate outtree=tree method=twostage k=10 noprint;
run;

proc tree noprint out=out n=2;
   copy x y;
run;

proc sgplot noautolegend;
   title 'Two-Stage Density Linkage Cluster Analysis: '
         'Parallel Elongated Clusters';
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

Figure 11.20 Parallel Elongated Clusters: PROC CLUSTER METHOD=TWOSTAGE



PROC FASTCLUS and average linkage fail miserably. Ward's method and the centroid method (not shown) produce almost the same results. Two-stage density linkage, however, recovers the correct clusters. Single linkage (not shown) finds the same clusters as two-stage density linkage except for some outliers.

In this example, the population clusters have equal covariance matrices. If the within-cluster covariances are known, the data can be transformed to make the clusters spherical so that any of the clustering methods can find the correct clusters. But when you are doing a cluster analysis, you do not know what the true clusters are, so you cannot calculate the within-cluster covariance matrix. Nevertheless, it is sometimes possible to estimate the within-cluster covariance matrix without knowing the cluster membership or even the number of clusters, using an approach invented by Art, Gnanadesikan, and Kettenring (1982). A method for obtaining such an estimate is available in the ACECLUS procedure.

In the following analysis, PROC ACECLUS transforms the variables X and Y into the canonical variables Can1 and Can2. The latter are plotted and then used in a cluster analysis by Ward's method. The clusters are then plotted with the original variables X and Y.

The following SAS statements produce Figure 11.21 and Figure 11.22:

```
proc aceclus data=elongate out=ace p=.1;
   var x y;
   title 'ACECLUS Analysis: Parallel Elongated Clusters';
run;

proc sgplot noautolegend;
   title 'Data Containing Parallel Elongated Clusters';
   title2 'After Transformation by PROC ACECLUS';
   scatter y=can2 x=can1;
   xaxis label='Canonical Variable 1';
   yaxis label='Canonical Variable 2';
run;
```

Figure 11.21 Parallel Elongated Clusters: PROC ACECLUS

ACECLUS Analysis: Parallel Elongated Clusters

The ACECLUS Procedure

Approximate Covariance Estimation for Cluster Analysis

Observations	100	**Proportion**	0.1000
Variables	2	**Converge**	0.00100

Means and Standard Deviations

Variable	Mean	Standard Deviation
x	2.6406	8.3494
y	10.6488	6.8420

COV: Total Sample Covariances

	x	y
x	69.71314819	24.24268934
y	24.24268934	46.81324861

Initial Within-Cluster Covariance Estimate = Full Covariance Matrix

Threshold =	0.328478

Figure 11.21 *continued*

Iteration History

Iteration	RMS Distance	Distance Cutoff	Pairs Within Cutoff	Convergence Measure
1	2.000	0.657	672.0	0.673685
2	9.382	3.082	716.0	0.006963
3	9.339	3.068	760.0	0.008362
4	9.437	3.100	824.0	0.009656
5	9.359	3.074	889.0	0.010269
6	9.267	3.044	955.0	0.011276
7	9.208	3.025	999.0	0.009230
8	9.230	3.032	1052.0	0.011394
9	9.226	3.030	1091.0	0.007924
10	9.173	3.013	1121.0	0.007993

WARNING: Iteration limit exceeded.

ACE: Approximate Covariance Estimate Within Clusters

	x	y
x	9.299329632	8.215362614
y	8.215362614	8.937753936

Eigenvalues of Inv(ACE)*(COV-ACE)

	Eigenvalue	Difference	Proportion	Cumulative
1	36.7091	33.1672	0.9120	0.9120
2	3.5420		0.0880	1.0000

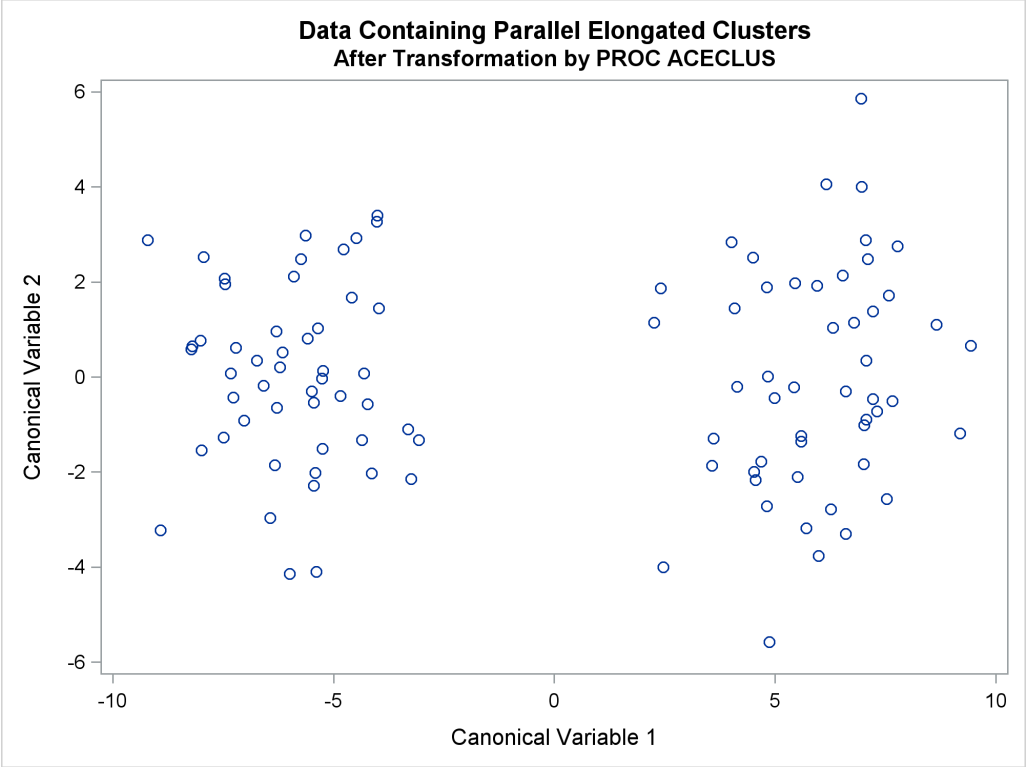
Eigenvectors (Raw Canonical Coefficients)

	Can1	Can2
x	-.748392	0.109547
y	0.736349	0.230272

Standardized Canonical Coefficients

	Can1	Can2
x	-6.24866	0.91466
y	5.03812	1.57553

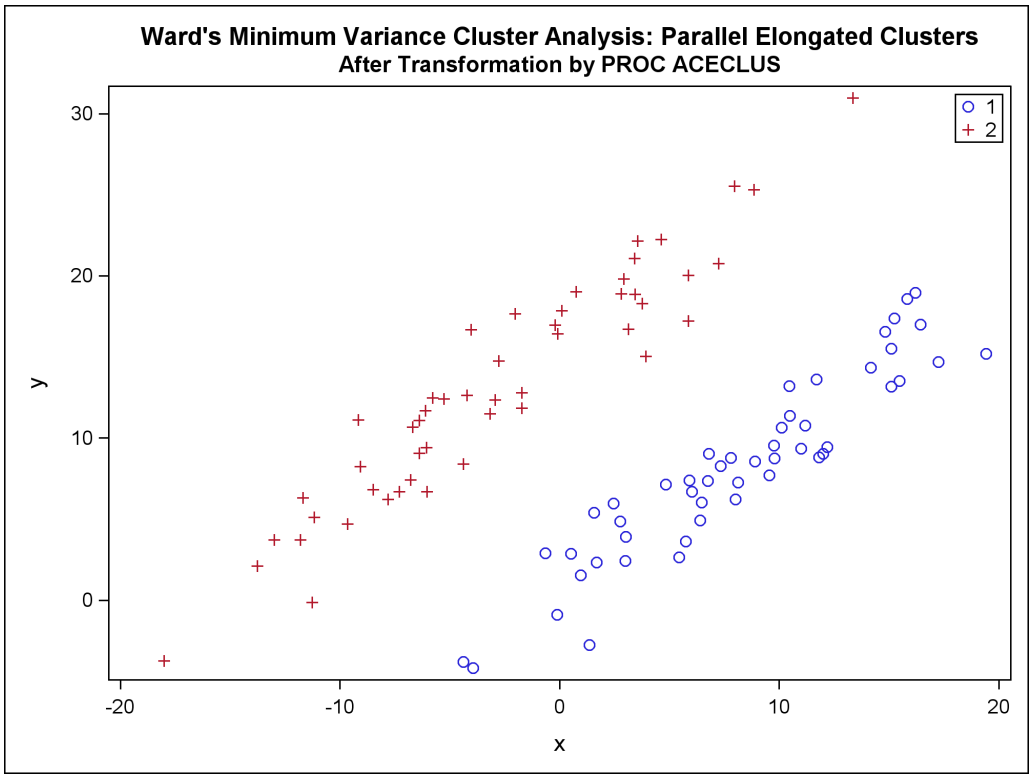
Figure 11.22 Parallel Elongated Clusters after Transformation by PROC ACECLUS



The following SAS statements produce Figure 11.23:

```
proc cluster data=ace outtree=tree method=ward noprint;
   var can1 can2;
   copy x y;
run;

proc tree noprint out=out n=2;
   copy x y;
run;

proc sgplot noautolegend;
   title 'Ward''s Minimum Variance Cluster Analysis: '
         'Parallel Elongated Clusters';
   title2 'After Transformation by PROC ACECLUS';
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

Figure 11.23 Transformed Data Containing Parallel Elongated Clusters: PROC CLUSTER METHOD=WARD

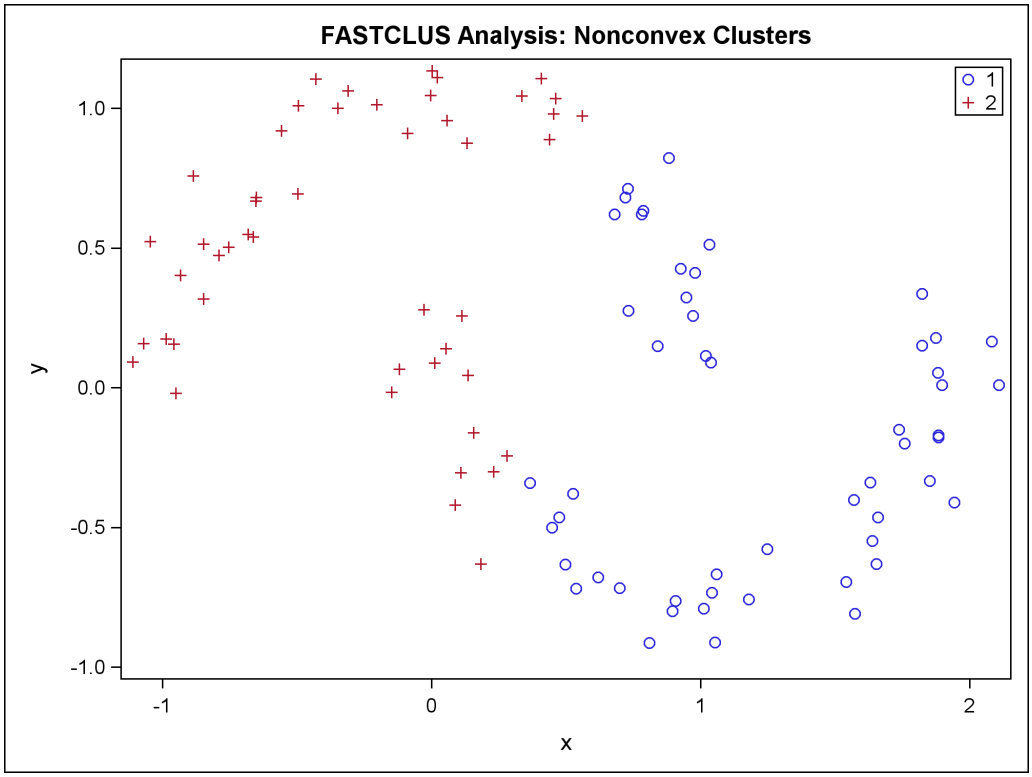


Nonconvex Clusters

If the population clusters have very different covariance matrices, using PROC ACECLUS is of no avail. Although methods exist for estimating multinormal clusters with unequal covariance matrices (Wolfe 1970; Symons 1981; Everitt and Hand 1981; Titterington, Smith, and Makov 1985; McLachlan and Basford 1988), these methods tend to have serious problems with initialization and might converge to degenerate solutions. For unequal covariance matrices or radically nonnormal distributions, the best approach to cluster analysis is through nonparametric density estimation, as in density linkage. The next example illustrates population clusters with nonconvex density contours. The following SAS statements produce Figure 11.24:

```
data noncon;
   keep x y;
   do i=1 to 100;
      a=i*.0628319;
      x=cos(a)+(i>50)+rannor(7)*.1;
      y=sin(a)+(i>50)*.3+rannor(7)*.1;
      output;
   end;
run;

proc fastclus data=noncon out=out maxc=2 noprint;
run;

proc sgplot noautolegend;
   title 'FASTCLUS Analysis: Nonconvex Clusters';
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

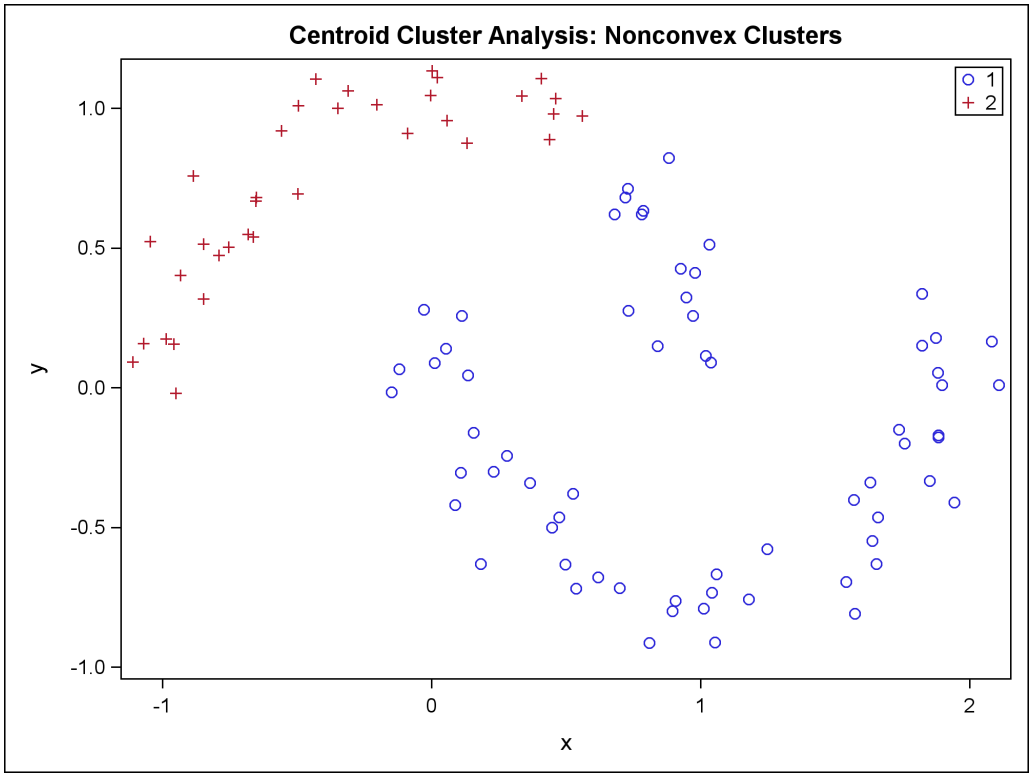
Figure 11.24 Nonconvex Clusters: PROC FASTCLUS



The following SAS statements produce Figure 11.25:

```
proc cluster data=noncon outtree=tree method=centroid noprint;
run;

proc tree noprint out=out n=2 dock=5;
   copy x y;
run;

proc sgplot noautolegend;
   title 'Centroid Cluster Analysis: Nonconvex Clusters';
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

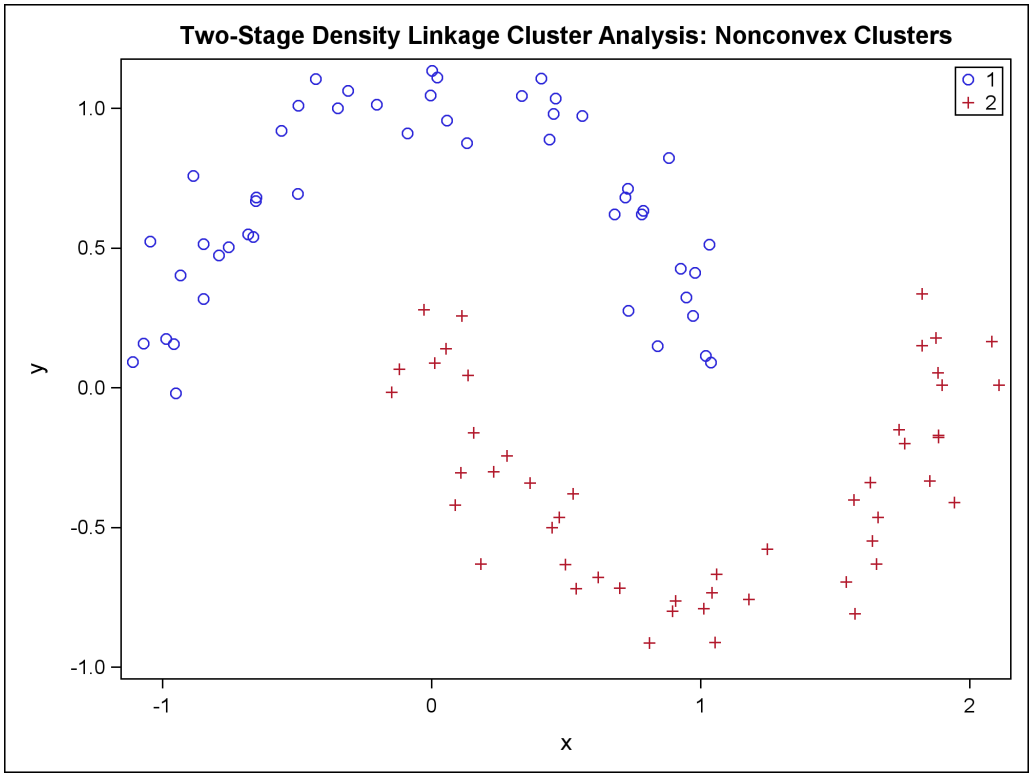
Figure 11.25 Nonconvex Clusters: PROC CLUSTER METHOD=CENTROID



The following SAS statements produce Figure 11.26:

```
proc cluster data=noncon outtree=tree method=twostage k=10 noprint;
run;

proc tree noprint out=out n=2;
   copy x y;
run;

proc sgplot noautolegend;
   title 'Two-Stage Density Linkage Cluster Analysis: Nonconvex Clusters';
   scatter y=y x=x / group=cluster;
   keylegend / location=inside position=topright sortorder=ascending
               across=1 noopaque title='';
run;
```

Figure 11.26 Nonconvex Clusters: PROC CLUSTER METHOD=TWOSTAGE



Ward's method and average linkage (not shown) do better than PROC FASTCLUS but not as well as the centroid method. Two-stage density linkage recovers the correct clusters, as does single linkage (not shown).

The preceding examples are intended merely to illustrate some of the properties of clustering methods in common use. If you intend to perform a cluster analysis, you should consult more systematic and rigorous studies of the properties of clustering methods, such as Milligan (1980).

The Number of Clusters

There are no completely satisfactory methods that can be used for determining the number of population clusters for any type of cluster analysis (Everitt 1979; Hartigan 1985a; Bock 1985).

If your purpose in clustering is dissection—that is, to summarize the data without trying to uncover real clusters—it might suffice to look at R square for each variable and pooled over all variables. Plots of R square against the number of clusters are useful.

It is always a good idea to look at your data graphically. If you have only two or three variables, use PROC SGPLOT to make scatter plots identifying the clusters. With more variables, use PROC CANDISC to compute canonical variables for plotting.

Ordinary significance tests, such as analysis of variance F tests, are not valid for testing differences between clusters. Since clustering methods attempt to maximize the separation between clusters, the assumptions of the usual significance tests, parametric or nonparametric, are drastically violated. For example, if you take a sample of 100 observations from a single univariate normal distribution, have PROC FASTCLUS divide it into two clusters, and run a t test between the clusters, you usually obtain a p-value of less than 0.0001. For the same reason, methods that purport to test for clusters against the null hypothesis that objects are assigned randomly to clusters (such as McClain and Rao 1975 and Klastorin 1983) are useless.

Most valid tests for clusters either have intractable sampling distributions or involve null hypotheses for which rejection is uninformative. For clustering methods based on distance matrices, a popular null hypothesis is that all permutations of the values in the distance matrix are equally likely (Ling 1973; Hubert 1974). Using this null hypothesis, you can do a permutation test or a rank test. The trouble with the permutation hypothesis is that, with any real data, the null hypothesis is implausible even if the data do not contain clusters. Rejecting the null hypothesis does not provide any useful information (Hubert and Baker 1977).

Another common null hypothesis is that the data are a random sample from a multivariate normal distribution (Wolfe 1970, 1978; Duda and Hart 1973; Lee 1979). The multivariate normal null hypothesis arises naturally in normal mixture models (Titterington, Smith, and Makov 1985; McLachlan and Basford 1988). Unfortunately, the likelihood ratio test statistic does not have the usual asymptotic χ^2 distribution because the regularity conditions do not hold. Approximations to the asymptotic distribution of the likelihood ratio have been suggested Wolfe (1978), but the adequacy of these approximations is debatable (Everitt 1981; Thode, Mendell, and Finch 1988). For small samples, bootstrapping seems preferable (McLachlan and Basford 1988). Bayesian inference provides a promising alternative to likelihood ratio tests for the number of mixture components for both normal mixtures and other types of distributions (Binder 1978, 1981; Banfield and Raftery 1993; Bensmail et al. 1997).

The multivariate normal null hypothesis is better than the permutation null hypothesis, but it is not satisfactory because there is typically a high probability of rejection if the data are sampled from a distribution with lower kurtosis than a normal distribution, such as a uniform distribution. The tables in Englemann and Hartigan (1969), for example, generally lead to rejection of the null hypothesis when the data are sampled from a uniform distribution. Hawkins, Muller, and ten Krooden (1982, pp. 337–340) discuss a highly conservative Bonferroni method for the use of hypothesis testing. The conservativeness of this approach might compensate to some extent for the liberalness exhibited by tests based on normal distributions when the population is uniform.

Perhaps a better null hypothesis is that the data are sampled from a uniform distribution (Hartigan 1978; Arnold 1979; Sarle 1983) The uniform null hypothesis leads to conservative error rates when the data are sampled from a strongly unimodal distribution such as the normal. However, in two or more dimensions and depending on the test statistic, the results can be very sensitive to the shape of the region of support of the uniform distribution. Sarle (1983) suggests using a hyperbox with sides proportional in length to the singular values of the centered coordinate matrix.

Given that the uniform distribution provides an appropriate null hypothesis, there are still serious difficulties in obtaining sampling distributions. Some asymptotic results are available (Hartigan 1978, 1985a; Pollard 1981; Bock 1985) for the within-cluster sum of squares, the criterion that PROC FASTCLUS and Ward's minimum variance method attempt to optimize. No distributional theory for finite sample sizes has yet appeared. Currently, the only practical way to obtain sampling distributions for realistic sample sizes is by computer simulation.

Arnold (1979) used simulation to derive tables of the distribution of a criterion based on the determinant of the within-cluster sum of squares matrix $|\mathbf{W}|$. Both normal and uniform null distributions were used. Having obtained clusters with either PROC FASTCLUS or PROC CLUSTER, you can compute Arnold's criterion with the ANOVA or CANDISC procedure. Arnold's tables provide a conservative test because PROC FASTCLUS and PROC CLUSTER attempt to minimize the trace of $\mathbf{W}$ rather than the determinant. Marriott (1971, 1975) also provides useful information about $|\mathbf{W}|$ as a criterion for the number of clusters.

Sarle (1983) used extensive simulations to develop the cubic clustering criterion (CCC), which can be used for crude hypothesis testing and estimating the number of population clusters. The CCC is based on the assumption that a uniform distribution on a hyperrectangle will be divided into clusters shaped roughly like hypercubes. In large samples that can be divided into the appropriate number of hypercubes, this assumption gives very accurate results. In other cases the approximation is generally conservative. For details about the interpretation of the CCC, consult Sarle (1983).

Milligan and Cooper (1985) and Cooper and Milligan (1988) compared 30 methods of estimating the number of population clusters by using four hierarchical clustering methods. The three criteria that performed best in these simulation studies with a high degree of error in the data were a pseudo F statistic developed by Caliński and Harabasz (1974), a statistic referred to as $J_e(2)/J_e(1)$ by Duda and Hart (1973) that can be transformed into a pseudo t^2 statistic, and the cubic clustering criterion. The pseudo F statistic and the CCC are displayed by PROC FASTCLUS; these two statistics and the pseudo t^2 statistic, which can be applied only to hierarchical methods, are displayed by PROC CLUSTER. It might be advisable to look for consensus among the three statistics—that is, local peaks of the CCC and pseudo F statistic combined with a small value of the pseudo t^2 statistic and a larger pseudo t^2 for the next cluster fusion. It must be emphasized that these criteria are appropriate only for compact or slightly elongated clusters, preferably clusters that are roughly multivariate normal.

Recent research has tended to deemphasize mixture models in favor of nonparametric models in which clusters correspond to modes in the probability density function. Hartigan and Hartigan (1985) and Hartigan (1985b) developed a test of unimodality versus bimodality in the univariate case.

Nonparametric tests for the number of clusters can also be based on nonparametric density estimates. This approach requires much weaker assumptions than mixture models, namely, that the observations are sampled independently and that the distribution can be estimated nonparametrically. Silverman (1986) describes a bootstrap test for the number of modes using a Gaussian kernel density estimate, but problems have been reported with this method under the uniform null distribution. Further developments in nonparametric methods are given by Müller and Sawitzki (1991); Minnotte (1992); Polonik (1993). All of these methods suffer from heavy computational requirements.

One useful descriptive approach to the number-of-clusters problem is provided by Wong and Schaack (1982) based on a *k*th-nearest-neighbor density estimate. The *k*th-nearest-neighbor clustering method developed by Wong and Lane (1983) is applied with varying values of *k*. Each value of *k* yields an estimate of the number of modal clusters. If the estimated number of modal clusters is constant for a wide range of *k* values, there is strong evidence of at least that many modes in the population. A plot of the estimated number of modes against *k* can be highly informative. Attempts to derive a formal hypothesis test from this diagnostic plot have met with difficulties, but a simulation approach similar to Silverman (1986) does seem to work Girman (1994). The simulation, of course, requires considerable computer time.

PROC MODECLUS uses a less expensive approximate nonparametric test for the number of clusters. This test sacrifices statistical efficiency for computational efficiency. The method for conducting significance tests is described in the chapter on the MODECLUS procedure. This method has the following useful features:

- No distributional assumptions are required.
- The choice of smoothing parameter is not critical since you can try any number of different values.
- The data can be coordinates or distances.
- Time and space requirements for the significance tests are no worse than those for obtaining the clusters.
- The power is high enough to be useful for practical purposes.

The method for computing the *p*-values is based on a series of plausible approximations. There are as yet no rigorous proofs that the method is infallible. Neither are there any asymptotic results. However, simulations for sample sizes ranging from 20 to 2000 indicate that the *p*-values are almost always conservative. The only case discovered so far in which the *p*-values are liberal is a uniform distribution in one dimension for which the simulated error rates exceed the nominal significance level only slightly for a limited range of sample sizes.

References

Anderberg, M. R. (1973). *Cluster Analysis for Applications*. New York: Academic Press.

Arnold, S. J. (1979). "A Test for Clusters." *Journal of Marketing Research* 16:545–551.

Art, D., Gnanadesikan, R., and Kettenring, J. R. (1982). "Data-Based Metrics for Cluster Analysis." *Utilitas Mathematica* 75–99.

Banfield, J. D., and Raftery, A. E. (1993). "Model-Based Gaussian and Non-Gaussian Clustering." *Biometrics* 49:803–821.

Bensmail, H., Celeux, G., Raftery, A. E., and Robert, C. P. (1997). "Inference in Model-Based Cluster Analysis." *Statistics and Computing* 7:1–10.

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum.

Bezdek, J. C., and Pal, S. K. (1992). *Fuzzy Models for Pattern Recognition*. New York: IEEE Press.

Binder, D. A. (1978). "Bayesian Cluster Analysis." *Biometrika* 65:31–38.

Binder, D. A. (1981). "Approximations to Bayesian Clustering Rules." *Biometrika* 68:275–285.

Blashfield, R. K., and Aldenderfer, M. S. (1978). "The Literature on Cluster Analysis." *Multivariate Behavioral Research* 13:271–295.

Bock, H. H. (1985). "On Some Significance Tests in Cluster Analysis." *Journal of Classification* 2:77–108.

Caliński, T., and Harabasz, J. (1974). "A Dendrite Method for Cluster Analysis." *Communications in Statistics—Theory and Methods* 3:1–27.

Cooper, M. C., and Milligan, G. W. (1988). "The Effect of Error on Determining the Number of Clusters." In *Proceedings of the International Workshop on Data Analysis, Decision Support, and Expert Knowledge Representation in Marketing and Related Areas of Research*, 319–328. Berlin: Springer-Verlag.

Duda, R. O., and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. New York: John Wiley & Sons.

Duran, B. S., and Odell, P. L. (1974). *Cluster Analysis*. New York: Springer-Verlag.

Englemann, L., and Hartigan, J. A. (1969). "Percentage Points of a Test for Clusters." *Journal of the American Statistical Association* 64:1647–1648.

Everitt, B. S. (1979). "Unresolved Problems in Cluster Analysis." *Biometrics* 35:169–181.

Everitt, B. S. (1980). *Cluster Analysis*. 2nd ed. London: Heineman Educational Books.

Everitt, B. S. (1981). "A Monte Carlo Investigation of the Likelihood Ratio Test for the Number of Components in a Mixture of Normal Distributions." *Multivariate Behavioral Research* 16:171–80.

Everitt, B. S., and Hand, D. J. (1981). *Finite Mixture Distributions*. London: Chapman & Hall.

Gersho, A., and Gray, R. M. (1992). *Vector Quantization and Signal Compression*. Boston: Kluwer Academic.

Girman, C. J. (1994). "Cluster Analysis and Classification Tree Methodology as an Aid to Improve Understanding of Benign Prostatic Hyperplasia." Ph.D. diss., Department of Biostatistics, University of North Carolina at Chapel Hill.

Good, I. J. (1977). *The Botryology of Botryology, in Classification and Clustering*. Edited by J. Van Ryzin. New York: Academic Press.

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

Hartigan, J. A. (1977). "Distribution Problems in Clustering." In *Classification and Clustering*, edited by J. V. Ryzin, 45–72. New York: Academic Press.

Hartigan, J. A. (1978). "Asymptotic Distributions for Clustering Criteria." *Annals of Statistics* 6:117–131.

Hartigan, J. A. (1981). "Consistency of Single Linkage for High-Density Clusters." *Journal of the American Statistical Association* 76:388–394.

Hartigan, J. A. (1985a). "Statistical Theory in Clustering." *Journal of Classification* 2:63–76.

Hartigan, J. A., and Hartigan, P. M. (1985). "The Dip Test of Unimodality." *Annals of Statistics* 13:70–84.

Hartigan, P. M. (1985b). "Algorithm AS 217: Computation of the Dip Statistic to Test for Unimodality." *Journal of the Royal Statistical Society, Series C* 34:320–325.

Hawkins, D. M., Muller, M. W., and ten Krooden, J. A. (1982). "Cluster Analysis." In *Topics in Applied Multivariate Analysis*, edited by D. M. Hawkins, 303–356. Cambridge: Cambridge University Press.

Hubert, L. J. (1974). "Approximate Evaluation Techniques for the Single-Link and Complete-Link Hierarchical Clustering Procedures." *Journal of the American Statistical Association* 69:698–704.

Hubert, L. J., and Baker, F. B. (1977). "An Empirical Comparison of Baseline Models for Goodness-of-Fit in *r*-Diameter Hierarchical Clustering." In *Classification and Clustering*, edited by J. Van Ryzin, 131–153. New York: Academic Press.

Kaufman, L., and Rousseeuw, P. J. (1990). *Finding Groups in Data*. New York: John Wiley & Sons.

Klastorin, T. D. (1983). "Assessing Cluster Analysis Results." *Journal of Marketing Research* 20:92–98.

Lee, K. L. (1979). "Multivariate Tests for Clusters." *Journal of the American Statistical Association* 74:708–714.

Ling, R. F. (1973). "A Probability Theory of Cluster Analysis." *Journal of the American Statistical Association* 68:159–169.

MacQueen, J. B. (1967). "Some Methods for Classification and Analysis of Multivariate Observations." *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* 1:281–297.

Marriott, F. H. C. (1971). "Practical Problems in a Method of Cluster Analysis." *Biometrics* 27:501–514.

Marriott, F. H. C. (1975). "Separating Mixtures of Normal Distributions." *Biometrics* 31:767–769.

Massart, D. L., and Kaufman, L. (1983). *The Interpretation of Analytical Chemical Data by the Use of Cluster Analysis*. New York: John Wiley & Sons.

McClain, J. O., and Rao, V. R. (1975). "CLUSTISZ: A Program to Test for the Quality of Clustering of a Set of Objects." *Journal of Marketing Research* 12:456–460.

McLachlan, G. J., and Basford, K. E. (1988). *Mixture Models*. New York: Marcel Dekker.

Mezzich, J. E., and Solomon, H. (1980). *Taxonomy and Behavioral Science*. New York: Academic Press.

Milligan, G. W. (1980). "An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms." *Psychometrika* 45:325–342.

Milligan, G. W. (1981). "A Review of Monte Carlo Tests of Cluster Analysis." *Multivariate Behavioral Research* 16:379–407.

Milligan, G. W., and Cooper, M. C. (1985). "An Examination of Procedures for Determining the Number of Clusters in a Data Set." *Psychometrika* 50:159–179.

Minnotte, M. C. (1992). "A Test of Mode Existence with Applications to Multimodality." Ph.D. diss., Department of Statistics, Rice University.

Müller, D. W., and Sawitzki, G. (1991). "Excess Mass Estimates and Tests for Multimodality." *Journal of the American Statistical Association* 86:738–746.

Pollard, D. (1981). "Strong Consistency of *k*-Means Clustering." *Annals of Statistics* 9:135–140.

Polonik, W. (1993). *Measuring Mass Concentrations and Estimating Density Contour Clusters—an Excess Mass Approach*. Technical Report 7, Beiträge zur Statistik, Universität Heidelberg.

Sarle, W. S. (1982). "Cluster Analysis by Least Squares." In *Proceedings of the Seventh Annual SAS Users Group International Conference*, 651–653. Cary, NC: SAS Institute Inc.

Sarle, W. S. (1983). *Cubic Clustering Criterion*. Technical Report A-108, SAS Institute Inc., Cary, NC.

Scott, A. J., and Symons, M. J. (1971). "Clustering Methods Based on Likelihood Ratio Criteria." *Biometrics* 27:387–397.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. New York: Chapman & Hall.

Sneath, P. H. A., and Sokal, R. R. (1973). *Numerical Taxonomy*. San Francisco: W. H. Freeman.

Spath, H. (1980). *Cluster Analysis Algorithms*. Chichester, UK: Ellis Horwood.

Symons, M. J. (1981). "Clustering Criteria and Multivariate Normal Mixtures." *Biometrics* 37:35–43.

Thode, H. C., Jr., Mendell, N. R., and Finch, S. J. (1988). "Simulated Percentage Points for the Null Distribution of the Likelihood Ratio Test for a Mixture of Two Normals." *Biometrics* 44:1195–1201.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*. New York: John Wiley & Sons.

Ward, J. H. (1963). "Hierarchical Grouping to Optimize an Objective Function." *Journal of the American Statistical Association* 58:236–244.

Wolfe, J. H. (1970). "Pattern Clustering by Multivariate Mixture Analysis." *Multivariate Behavioral Research* 5:329–350.

Wolfe, J. H. (1978). "Comparative Cluster Analysis of Patterns of Vocational Interest." *Multivariate Behavioral Research* 13:33–44.

Wong, M. A. (1982). "A Hybrid Clustering Method for Identifying High-Density Clusters." *Journal of the American Statistical Association* 77:841–847.

Wong, M. A., and Lane, T. (1983). "A *k*th Nearest Neighbor Clustering Procedure." *Journal of the Royal Statistical Society, Series B* 45:362–368.

Wong, M. A., and Schaack, C. (1982). "Using the *k*th Nearest Neighbor Clustering Procedure to Determine the Number of Subpopulations." *Proceedings of the Statistical Computing Section, American Statistical Association* 40–48.