

Introduction to Clustering Algorithms

Jason Brinkley, Abt Global Inc.

ABSTRACT

Clustering algorithms have become very popular in recent years as a means to extract new information from data of all types. A clustering algorithm is data driven and partitions the observed data up into a non-overlapping set of clusters with clusters having similar values in the original data. Often cluster algorithms follow two paths: quantitative and qualitative. Quantitative clustering algorithms often use continuous data and statistical measurements of 'distance' to group similar items together to form clusters. Qualitative clustering looks to take data with only a finite and fixed number of levels and group data with similar levels together. This hands-on workshop will explain in detail both quantitative and qualitative clustering from both a theoretical and code driven perspective. Topics include Ward and centroid based clustering, hierarchical clustering, dendrograms and other multivariate visualizations that assist in developing or showcasing the results of a clustering algorithm.

INTRODUCTION

There is a growing need to study large volumes of data to create actionable insights. In the past many problems were not well understood and insights could be gained by isolating single aspects of complex problems for study one at a time. The modern era sees challenges that often involve understanding across multiple fronts simultaneously and a need to drive decision-making by considering multiple factors simultaneously. Multivariate statistics concerns itself with drawing understanding from large, complex, and often messy datasets.

Multivariate statistics is a subdiscipline of statistics that focuses on the study of data with multiple characteristics taken together to help provide general insights. The focus of multivariate work is to help consolidate information from data across many dimensions in order to make insights that could not have been easily made exploring individual variables one at a time. It is often the case that a single dimension does not tell the entire story of a set of data. In health studies, for example, we often find that aggregating information across biology, psychology, and social support systems is key in understanding root causes in patient disparities or understanding how the interplay of patient characteristics and environment contribute toward health outcomes.

Statistical clustering is a collection of techniques that form data-driven non-overlapping subgroups or 'clusters' within a set of data. These techniques are often driven by algorithms that group 'similar' data together by using one or more variables and to bring together data that have the same (or 'close' values). Such groups could be made by clinical or subject matter expertise; for example, we could group students in a classroom by scores on a test, scores across multiple tests, or groupings where test scores were consolidated into letter grades and groups made based on collections of students who obtained the same letter grade across one or more tests. Here we can see that the number of ways data could be grouped together could be near-infinite and without a reasonable justification as to why data are being grouped together, may seem arbitrary to some. Clustering algorithms work by creating such groupings using data in lieu of such subject matter expertise. These techniques are best used when there does not seem to be a single well-defined indicator that adequately breaks data up into subgroups for meaningful analysis.

CLUSTERING ALGORITHMS

There are many different types of clustering algorithms, but all have three general starting points:

1. What type of data is being clustered? Algorithms for qualitative (or categorical) data is often clustered differently than quantitative data. Within each type of data there is even more depth, there are algorithms specifically designed for geospatial data, open text data, functional data, and others.
2. How many clusters does one want to make? Clustering is a data driven process and as will be seen via example, requires some trial and error. Since these techniques do not require insights

into the mechanisms of data generation or subject matter expertise, it can be hypothesized but is rarely known how a set of data will break out into clusters.

3. How will data be grouped as 'similar'? For categorical data with few levels (e.g. biological sex, hair color, preferred language, etc.) then similarity can be as straightforward as grouping data with the same levels. But how to prioritize which categories to group? If two individuals are of the same biological sex but with different hair color and preferred language; is that grouping 'better' than grouping based first on preferred language? For continuous data, we often group similarity by some association or measure of distance between points. Pearson correlations and Euclidean distance are common mechanisms for determining similarity but can be biased for skewed data.

As this is a simple introduction to clustering, the focus here will be on a handful of commonly used algorithms that are available to most SAS® users.

CLUSTERING IN SAS

If you think about data in the SAS format with variables as columns and observations as rows, then a lot of multivariate statistics breaks down into two sets of ideas:

- How can we group variables (columns) together to form new variables that give meaning to the relationships within the data?
- How can we collect groups of observations (rows) so that those groups represent data that have commonalities not already measured in a single column variable?

The first idea (grouping columns) is the general work of data reduction techniques such as Principal Components Analysis and Factor Analysis. In contrast, Statistical Clustering is usually focused on implementing the second idea. There are a number of procedures in SAS that focus on clustering. Table 1 below has a listing a brief overview of their purpose.

SAS Procedure	Purpose
Proc Cluster	Traditional hierarchical cluster procedures for quantitative data.
Proc VarClus	Focuses on clustering that incorporates variable reduction (i.e. Principal Components Analysis).
Proc FastClus	Non-hierarchical based clustering that is computationally fast but sensitive to outliers
Proc HPClus	Clustering algorithms for big data that uses distributed computing.
Proc ACEClus	Computationally fast clustering algorithms designed for correlated multivariate normal data.
Proc ModelClus	Time-consuming clustering algorithms that are based on nonparametric density estimates.

Table 1. SAS Procedures for Statistical Clustering and General Purpose

From the options in Table 1 a few themes emerge; first, clustering can be computationally time consuming for large data. As clustering is an iterative process where similar data are joined together, data that has many rows and many columns will need to process a many different possible groupings based on the selected notion of distance or similarity. Next, there are algorithms that are built on a hierarchy (more on this below) versus those that are not; similarly, some algorithms make use of assumptions regarding statistical distributions of data to reduce computation time.

Here we will focus initially on Proc Cluster which is easy to implement and flexible. Note that unlike some other procedures, Proc Cluster does not immediately produce the cluster output and is often combined with other procedures for you to out and use the results of your cluster algorithm. Proc Cluster can be a very nice starting point to experiment with many different cluster algorithms to explore fit. There are statistics for determining the number of clusters needed and statistical tests, but clustering is often a 'fit for purpose' technique as we will see via example.

TOY EXAMPLE – BIG CLASS

The SAS Help data file 'Class' contains data on 19 students with their age, sex, height, and weight. Table 2 below shows all 19 observations and their multivariate data.

	Name	Sex	Age	Height	Weight
1	Joyce	F	11	51.3	50.5
2	Louise	F	12	56.3	77
3	Alice	F	13	56.5	84
4	James	M	12	57.3	83
5	Thomas	M	11	57.5	85
6	John	M	12	59	99.5
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84
10	Carol	F	14	62.8	102.5
11	Henry	M	14	63.5	102.5
12	Judy	F	14	64.3	90
13	Robert	M	12	64.8	128
14	Barbara	F	13	65.3	98
15	Mary	F	15	66.5	112
16	William	M	15	66.5	112
17	Ronald	M	15	67	133
18	Alfred	M	14	69	112.5
19	Philip	M	16	72	150

Table 2. SAS Help Class Data

Focusing on Age, Height, and Weight we can form a scatterplot of height by weight labeled by age which we see in Figure 1 below. Note that in addition to the linear association between height and weight we see that older students are generally taller and heavier.

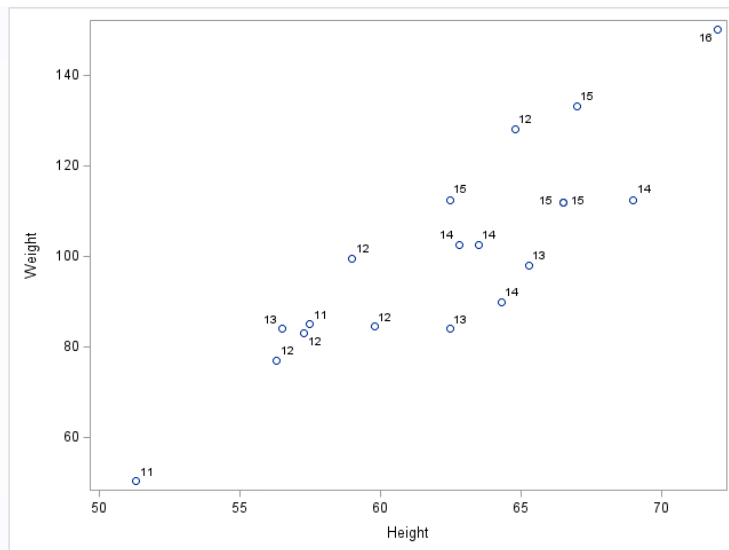


Figure 1. Scatterplot of Class Data for Height by Weight labeled by Age

Suppose I want to cluster this data into three distinct clusters; first note that three clusters might not be what is best for this data but we will use it as a starting point. So let's implement a traditional cluster algorithm in Proc Cluster with the following code:

```
ods graphics on;
proc cluster data=SASHelp.Class method=average outtree=Output;
    var height weight age;
    id name;
run;
ods graphics off;
```

We turn ods graphics on in order to obtain visuals that are key for exploring the results of clustering. We will use the 'average' based clustering algorithm with inputs on height, weight, and age. The ID variable helps us track which individuals are clustered together at which iteration and the outtree option creates a dataset to be used later in assigning individuals to clusters.

The output comes in three parts, we will skip the measures based output at this time so we can focus on exactly what happens in a standard cluster algorithm. From this code we obtain the output seen in Figure 2 below; the clustering algorithm starts with 19 clusters (one cluster for each data point) and then works to group individual observations together based on the distance measure (here focused on minimizing average Euclidean distance between observations). The algorithm 'chose' to group Mary and William together as it's first cluster which we can see from Table 1 as having the exact same height, weight, and age (so the distance between such points is zero). Next, the algorithm grouped Carol and Henry together as they have the same age and weight but slightly different heights. The process continues until we get to 15 clusters at which the cluster formed at 16 clusters (Alice and James) are grouped with Thomas to form a cluster with 3 individuals. Note that traditional clustering does not seek balanced cluster sizes as the metric is centered on minimizing the average distance inside a cluster. Finally, note that at the last iteration of clustering, Joyce is grouped with the rest of the data. From Table 1, we see that Joyce is the youngest and shortest but also has a smaller weight than the rest of the data. This showcases the impact that outliers can have on clustering and one reason why there are alternative clustering algorithm procedures in SAS as some are less sensitive to outliers.

Cluster History					
Number of Clusters	Clusters Joined		Freq	Norm RMS Distance	Tie
18	Mary	William	2	0	
17	Carol	Henry	2	0.0212	
16	Alice	James	2	0.0491	
15	CL16	Thomas	3	0.071	
14	Alfred	CL18	3	0.0828	
13	Jane	Jeffrey	2	0.0883	
12	CL15	CL13	5	0.1364	
11	CL14	Janet	4	0.1519	
10	Barbara	CL17	3	0.1541	
9	CL10	John	4	0.1777	
8	Robert	Ronald	2	0.1884	
7	CL12	Louise	6	0.2382	
6	CL7	Judy	7	0.3053	
5	CL11	CL9	8	0.3882	
4	Philip	CL8	3	0.6293	
3	CL5	CL6	15	0.7424	
2	CL3	CL4	18	1.3491	
1	CL2	Joyce	19	1.7288	

Figure 2. Cluster History

The second major output from Proc Cluster is a dendrogram, which is a visual representation of the clustering algorithm based on the distance measure and the results of the clustering algorithm itself. The dendrogram is shown below in Figure 3.

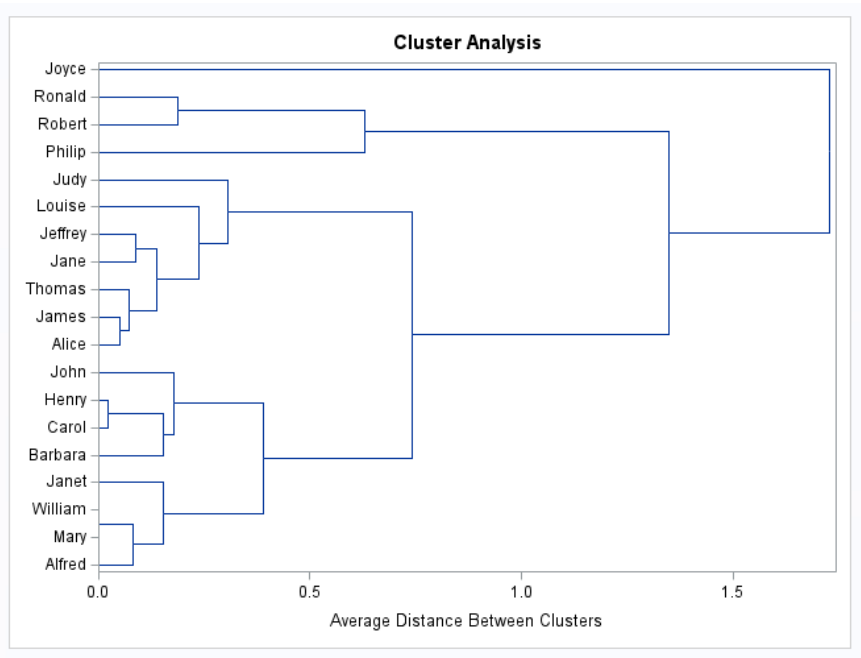


Figure 3 – Dendrogram of Proc Cluster Results

Dendrograms showcase the Cluster History and should be read from right to left. Lines that are connected form common groupings. As such, a three cluster solution from this data would have the following groups:

1. Joyce in a cluster all alone
2. Ronald, Robert, and Phillip together as a cluster
3. Everyone else as a cluster

Again, clusters follow the principle of the algorithm chosen and the distance measure; all of which can be varied for different results. Proc cluster does not natively select a specific number of clusters for the data as the output is designed to explore the potential number of clusters needed for analysis. To automate output of this three-cluster solution, we need to use the Output dataset created by the outtree statement. We do that using a secondary procedure called Proc Tree:

```
proc tree noprint ncl=3 out=clusterOut;
run;
```

This procedure produces a new dataset called clusterOut that has each observation ID and which cluster they belong to. This dataset will have to be merged with the original data. Note that other SAS procedures do output the clusters directly in the data (more on this later).

So having seen the basics of clustering, this particular solution might seem unsatisfactory. Note that statistical clustering is not immediately driven by statistical inference. It is an exploratory method that can be used many times to help generate insights or create groupings that are not immediately available in the data. We started with Euclidean distance based on averages and found that an outlier drove the construction of our cluster groups. But Figure 1 shows that the data falls into a predictable linear pattern and as such Joyce is less of an outlier than one might initially suspect. Consider a different algorithm for clustering as proposed below:

```
ods graphics on;
proc cluster data=SASHelp.Class method=Ward outtree=Output;
var height weight age;
```

```

        id name;
run;
ods graphics off;

```

Ward's clustering is based not on minimizing averages but in minimizing variance, or squared Euclidean distance. As such the results yield different clusters and have a different measure of performance. Consider the dendrogram from this run of code:

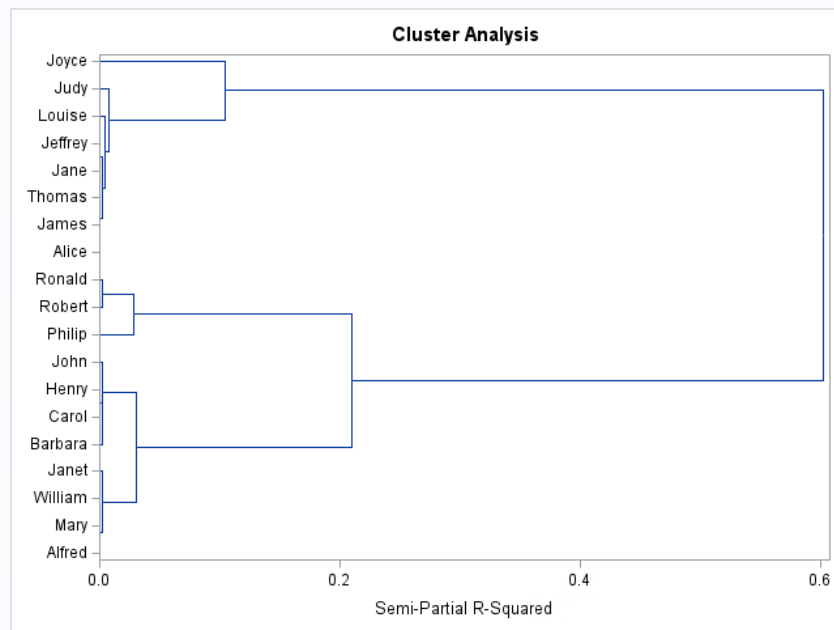


Figure 4 – Dendrogram of Proc Cluster with Ward's Method

Note that we still have a cluster with Ronald, Robert, and Philip; but the remaining data are now more evenly split with Joyce being grouped with other short and young individuals. It is critical to note that there is no right answer for clustering algorithms and both solutions are potentially useful depending on application. Ward's method breaks the data up into three groups of size 3, 8, and 8 while the average based method made three groups of size 1, 3, and 15. Here, Ward may have created clusters that are more usable in that they have allocated data in a more balanced fashion.

EXAMPLE – K-MEANS IN FASTCLUS

The algorithms in Proc Cluster focus mostly on hierarchical clustering where individual data points are grouped together iteratively using some sort of criteria for alignment. Not all clustering algorithms are hierarchical in nature. Among the most popular forms of clustering is k-means clustering, which is based on a series of algorithms that find cluster groups based on creating a set number of groups whose means are most different from one another. Termed 'k-means' the k is intended to be pre-specified and is a centroid based methodology. These algorithms work by defining a center and a radius in multiple dimensions and every data value that falls in that radius belongs to that cluster. Consider the following example created from synthetic data:

```

Data Example;
  call streaminit(8675309);
  do n = 1 to 1000;
    if n < 501 then x = rand("Normal",100,2);
    else x = rand("Normal", 20,1);
    if n < 501 then y = rand("Normal",50,5);
    else y = rand("Normal", 10,1);
    output;
  end;

```

```

end;

run;

proc sgplot data=Example;
    scatter x=X y=Y;
run;

```

This code creates data that can be showcased in the scatterplot shown below in Figure 5. Note the data have two well defined regions of data in the corners of the graph. This is the perfect type of data for k-means clustering as there are fairly specific groupings of data that are unambiguous but there currently exists no third variable that puts each data point into it's respective group.

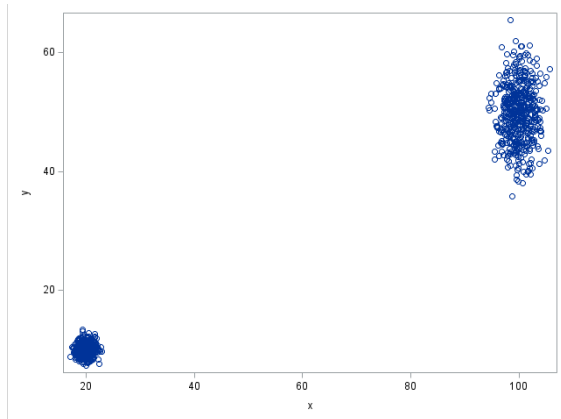


Figure 5. Scatterplot of Synthetic Data

We can run k-means using the Fastclus procedure:

```

proc fastclus data=Example out=Output
    maxclusters = 2 summary;
    var X Y;
run;

proc sgplot data=Output;
    scatter x=X y=Y/group=Cluster;
run;

```

The maxclusters command tells SAS how many clusters to produce, unlike Proc Cluster, we can immediately obtain cluster output which we can once again make a scatterplot which highlights the results of our clustering. As can be seen in Figure 6, the data were broken up into clusters that align with visual inspection.

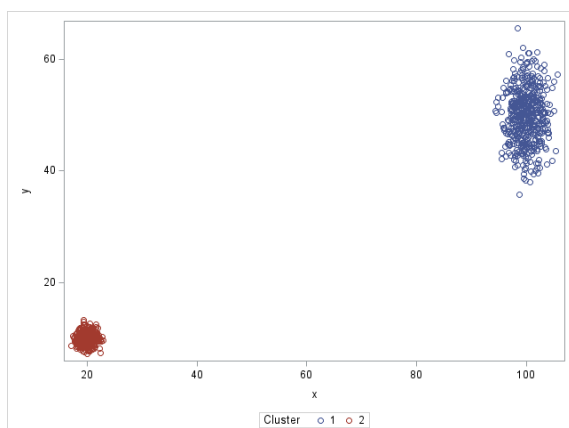


Figure 6. Scatterplot of Synthetic Data with K-Means Clustering (K=2)

Visual alignment is not always guaranteed. Consider using the same algorithm on the SAS Help dataset 'Zip Code' which shows map data for the US and territories. Suppose we run virtually the same k-means cluster algorithm:

```
proc fastclus data=sashelp.zipcode out=Output
    maxclusters = 5
    summary;
    var X Y;
run;
```

We can obtain a similar scatterplot output by using the same code as before but the results look very different. Figure 7 shows the results of the k-means clustering. It is clear that there are 5 distinct groupings in the scatterplot, with the continental United States making up the bulk of data. Given the proximity of Puerto Rico to other portions of the United States, the k-means algorithm instead grouped that data with a large portion of the eastern United States. While popular and fast to compute, k-means clustering is popular but not always the best fit. More on general and best practices in clustering in the discussion section below.

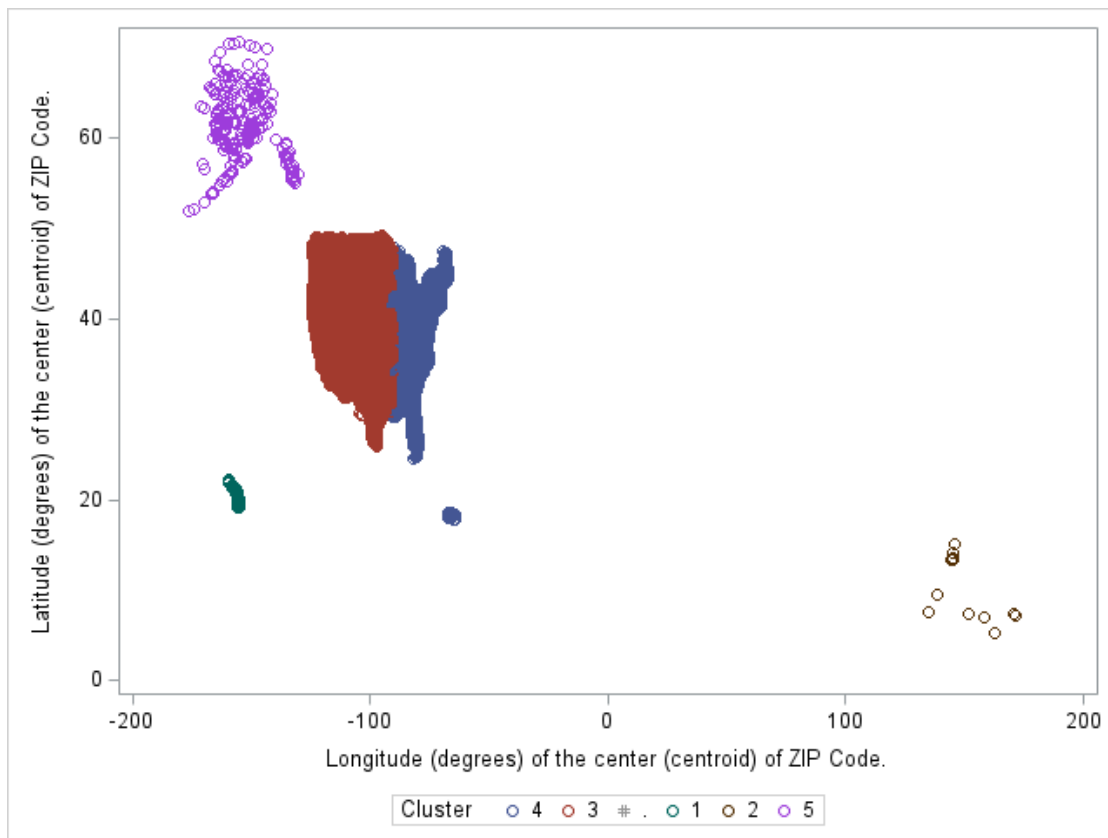


Figure 7. Scatterplot of US Zip Code Data with K-Means Clustering (K=5)

CLUSTERING CATEGORICAL/MIXED DATA

While there are a multitude of clustering algorithms built specifically for categorical data, many are not available in the traditional SAS/STAT® platform. There are options in SAS Viya® and with the High Performance computing add-ons (e.g. HPClus) but few options using more routine packages. So in lieu of focusing on clustering just categorical data, here we will revisit the toy scenario and add a categorical input. Recall from Table 2 that the SAS Help dataset Class contains not only height, weight, and age but also sex. Suppose I want to incorporate one or more categorical variables into a cluster analysis. Note that when forming groups of multiple categorical variables that this can also be explored manually by

looking at intersections of categorical variables and their relative frequencies. For example, if there are three variables with 2 levels each then there is a maximum of $2 \times 2 \times 2 = 8$ intersecting groups which would be the starting point for any clustering algorithm. That is to say that the maximum possible number of clusters in all categorical data is limited to the maximum number of intersecting groups, and while there are algorithms to help consolidate those intersecting groups into clusters based on some criteria, more often such groupings are made via clinical insight or subject matter expertise.

In looking at mixed type data, we can use the same approach as before by prespecifying the number of desired clusters; however, to use Proc Cluster on mixed type data we need a common measure of similarity or distance between data points. For many of the examples discussed so far, Euclidean distance has been the measure of interest. Gower (1971) created a general similarity metric which can be applied to both quantitative and categorical data to measure similarity. Others have shown that Gower Similarity can be reduced to other measures of similarity or distance depending on circumstance but used on its own can be a powerful tool to help add categorical data to the discussion.

Returning to the Class dataset we start with using Proc Distance to create a distance matrix for input into Proc Cluster in lieu of raw data:

```
Proc distance data=SASHelp.Class method=dgower out=distances;
    id name;
    var nominal (sex);
    var interval (height weight age);
run;

ods graphics on;
Proc Cluster data=distances method=Ward outtree=Output;
    id name;
run;
ods graphics off;
```

Note that we are again applying Ward's method but instead of feeding the raw data, one can use a distance matrix from Proc Distance instead. This can be useful when one has alternative distance measures to be used in lieu of a traditional metric like Euclidean distance. More on distance measures in the next section. Moving on to the cluster algorithm we see a very different Dendrogram when adding sex to the variable mix, as is shown in Figure 8 below. We see that all female students and all male students are initially grouped together, and smaller cluster solutions have breakouts of just males and just females.

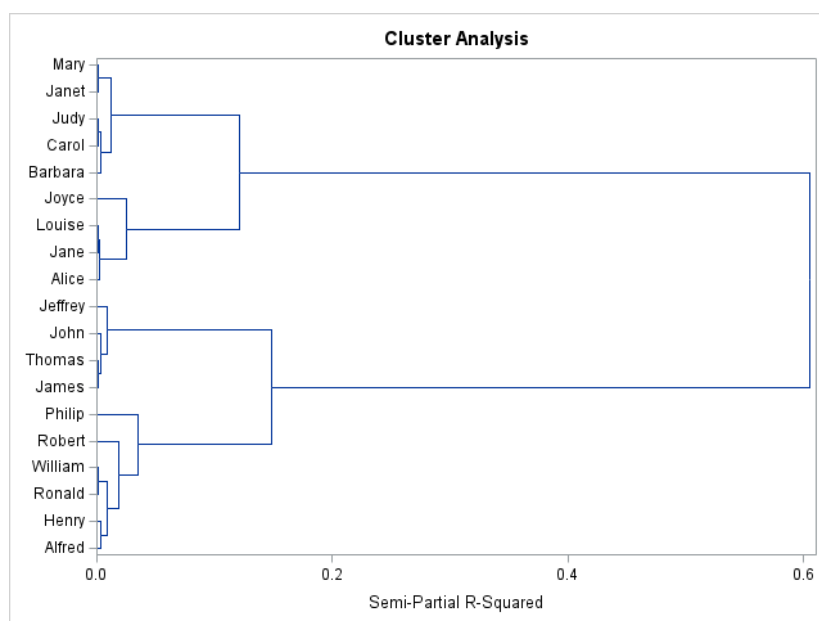


Figure 8. Dendrogram of Class Data with Mixed Variable Type Input

DISCUSSION

This work is designed to be a starting point in cluster analysis and is not designed to be comprehensive. Through example we have seen the power and utility of clustering as a way to gain new insights from data.

We have seen through example that there are many varieties of clustering and the reader should note that each cluster algorithm is different and there is no one direct way of clustering that is most effective for all types of data. Methods such as Ward's algorithm and k-means are very popular and are often quite effective in helping make sense of complex data. Many of the details and options discussed here are dependent on having advanced statistical expertise. One can get 'better' clusters in the same ways that one typically gets a 'better' statistical model of any type: if one can understand and classify the statistical distribution the data comes from or have a sense of need from a cluster solution (for example, whether one should cluster take linear associations between variables into account).

Finally, the measure of similarity or distance is especially key. Many rely on traditional Euclidean distance as the go to measure but there are others that can be useful. Some options include:

- Euclidean – Traditional metric for the distance between two points on a Cartesian plane.
- Manhattan – A more generalized form of distance that uses norms (absolute values) in settings where squared distance makes less sense.
- Mahalanobis – Distances based on the deviation from a known statistical distribution.
- Relative Frequency – Common for clustering all categorical data.
- Correlation – While Pearson Correlation based clustering often provides the same answer as Euclidean distance since they are built on harmonious principles, alternative correlations such as Spearman Rank, Cramer's V, and Kappa can all be used to cluster in the right data driven scenario.
- Other – There are distance functions specific to geospatial data, genomic data, and even correlations can be used as distance functions in some situations.

Clustering is first and foremost an exploratory technique. Often clustering is best used when it is suspected that one or many variables are acting together and there are unique subgroups of individuals who perform differently. Clustering is often data set specific and can sometimes be hard to reproduce in new or alternate data which makes validating clusters difficult. Limitations aside, clustering can be a powerful tool to organize initial summaries of data and to help draw insights in ways many other tools can't. It is flexible and easy to implement in SAS and the analyst often loses nothing but time in such an initial exploration.

REFERENCES

SAS Institute Inc., SAS 9.1.3 Help and Documentation, Cary, NC: SAS Institute Inc., 2002-2004.

Gower, J. C. "A General Coefficient of Similarity and Some of Its Properties." *Biometrics*, vol. 27, no. 4, 1971, pp. 857–71. JSTOR, <https://doi.org/10.2307/2528823>. Accessed 5 Sept. 2024.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Jason Brinkley
Enterprise: Abt Global Inc
Address: 5001 South Miami Blvd
City, State ZIP: Durham, NC 27703
E-mail: Jason_Brinkley@abtglobal.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.