Developing a Marketing Geographic Segmentation System Using SAS® Software

Kellie M. Poulin and Allison N. Freeman
Marsh Affinity Group Services, Ft. Washington, PA

ABSTRACT

Today's marketers are challenged by decreasing response rates and increasing competition. In order to continue to thrive in the industry, the successful marketer must move away from traditional mass marketing and toward a one-to-one customer-based approach (Peppers and Rogers, 1997). In order to make this change, marketers must be able to find and effectively use low-cost information to improve their level of customer insight. One popular approach is to implement an intelligent marketing segmentation system. This paper presents a low-cost, efficient solution integrating publicly available data and the marketer's own customer database to create a segmentation system that will aid in identifying, locating, and reaching customers effectively. The example presented in this paper will highlight the process of creating a ZIP code segmentation system with Base SAS®, SAS/STAT® and the SAS® Bridge for ESRI. The process for creating a segmentation system can be implemented across all operating systems, while the display of the system using ESRI can be implemented in a Windows environment. The topics discussed in this paper are appropriate for the intermediate programmer or statistical analyst.

INTRODUCTION

The marketing industry has changed dramatically over the past decade. The advent of inexpensive information technology has made it possible for companies to move from mass marketing, or marketing the same product to all consumers in the same way, to one-to-one marketing. In one-to-one marketing, information about the customer is used to tailor products, services, and communications to each individual (Peppers and Rogers 1997). The new challenge for the marketer is finding affordable information that can be used effectively to differentiate their customers. This paper will discuss one very low-cost method that can be used as a first step in this process.

It has been shown that people with similar demographic characteristics and product usage patterns tend to live near one another. In other words, there are geographic areas where residents are more likely to respond to certain product offers due to their environment and demographics. Several companies have created groupings of similar areas, called geographic segmentation systems. These can be purchased and applied to any customer database based on where the customer lives (Shepard 1995). The Conclusion of this paper lists several vendors of geographic level segmentation systems and provides their web site addresses. While these systems are often helpful in identifying customers with similar buying patterns, they are often costly and are not designed for an individual company's specific product offerings. Custom designed clusters can be more powerful because information about a company's current customer base can be incorporated into the model. In our experience, we have found that the data we collect about our customers, such as past customer behavior and products purchased, are valuable in developing custom-built segmentation tools and response models, thereby making them much more predictive than purchased tools.

Custom geographic segmentation systems can be developed for little cost using data from the U.S. Census Bureau, other government sources, and customer information aggregated to a geographic level. Census data are available on state, county, census tract, census block group, and ZIP code levels. Thus, clusters could be developed on any of these levels.

Once these clusters are developed, how can they aid marketers and business analysts? Marketers and analysts can use geographic segmentation tools to track buying trends and other customer behavior. A customer who falls into a specific segment will often respond to product offers in the same way that other people in that segment will respond. Also, segmentation systems can help to determine where a current customer base is concentrated and where to focus new marketing efforts. Analysts can locate areas with low customer penetration that have similar geographic characteristics to areas with high customer penetration and can use this information in their marketing strategies. A geographic segmentation system also enables analysts to use environmental, demographic and behavioral information to create appropriate marketing offers.

In developing a custom segmentation system, you should begin by asking the following questions:

- What are the basic characteristics of my best customers?

  What is their level of income? What is their average age and education level?
- Where do my best customers live?

  Do they live in areas with high population densities? Do they live where a large portion of the population is retired?
- What geographic factors affect the use of my product?

  Are my prices more competitive in some areas of the country? Is my product used more in warmer climates than in colder climates?

Often these answers are found through statistical analysis and in brainstorming sessions with the marketing staff of your company. The answers to these and other related questions will lead you to decide which data elements you will want to consider when creating your custom segmentation system.

In the example discussed in this paper, we illustrate the steps involved in developing a ZIP code segmentation system using customer data from our marketing database along with geographic-level data from the Census. Although it is not covered in this paper, a geographic-level segmentation system can often be a building block to more sophisticated models. Geographic segments can be used as inputs into individual-level segmentation systems and other models that include individual and household-level customer information.

GATHERING PERTINENT DATA

A segmentation system, or clustering schema, is designed to assign people to groups based on related characteristics. The variables you choose should be data elements that describe your customer base rather than the variables that are "statistically significant contributors," as would be used in developing a predictive model (Shepard 1995).

When choosing your variables, it is important to carefully consider what available data elements are the most relevant to your purpose. Clustering depends entirely on the variables that are input into the procedure; the procedure itself cannot distinguish between relevant and irrelevant variables.

Because clustering means that you are dividing a population into a number of homogeneous groups, you will have no response variable. Instead, those who will be using the segmentation system should choose what characteristics should divide the population into groups. For our example, a group of marketers and analysts convened to decide which information about a geographic area would be helpful in differentiating one area from the next. We devised a list of possible geographic data elements that would be the most important to our marketing group, and then we focused on what data were available to us for this purpose.

Another consideration when choosing your variables is the effect of multicollinearity. If several of your variables are related to each other, it may impact your final solution. This is because correlated data elements will have more weight than other non-correlated data elements when used together in your clusters. (Hair, Anderson, Tatham, and Black 1998). The more data elements you incorporate into your clustering solution, the higher the chance that some of your data elements may be correlated. For example, an analyst wants to create a clustering schema based on four variables: median education level, average income, average home value from the Census and the penetration of customers in the ZIP code from the company's database. The first three variables are all correlated with affluence. Therefore, affluence will have much higher weight in the analyst's clustering solution than will the current customer penetration. If this is not the intention, the analyst must find a way to combat this problem.

There are two approaches that can be taken to ensure that all variables have a similar weight in the clustering process. The first approach is to eliminate correlated data elements from your list by keeping only one variable from every set of correlated variables. In our example, the analyst may remove median education level and average home value while keeping average income as the variable of interest. The second approach would be for the analyst to create an "affluence factor" that is a combination of the three variables of interest. This approach will be discussed in the Factor Analysis section of this paper.

Finally, when choosing your data elements for clustering, it is important to note that the procedure works best with continuous numeric data. For example, if you wanted to include gender in your model, it would be best to use the percentage of people living in the area that is male, rather than an indicator stating that a ZIP code is predominantly male. Similarly, penetrations and percentages out of a total number rather than raw counts will better allow you to compare areas to one another. For example, knowing that 13% of the population in an area is insured with a certain product is more intuitive than knowing that 200 people are insured with the product in that area.

U.S. CENSUS DATA

Census data are public-domain information and can be found on the U.S. Bureau of the Census web site: http://www.census.gov. Data from the 2000 U.S. Census is compiled into three summary files which are available at no cost from the Census web site. These summary files are a rich source of information and are useful in creating geographic clusters. They contain data describing age, gender, occupation, industry, and income distributions on a variety of geographic levels.

For our clustering example, total population from the census is used in calculating penetrations of customers in each ZIP code and population density (population per square mile). Our marketers also expressed that areas containing large percentages of group quarters population would be unattractive to our client when marketing certain products. Thus, we included the group quarters data from the Census to identify areas where military bases and college dormitories are located.

OTHER PUBLIC DATA

One of the products that our company markets is health insurance. Thus, knowing what ZIP codes contain a high or low concentration of hospitals could be helpful in identifying good areas for marketing health insurance. A count of hospitals per ZIP code was calculated from hospital listings found on the Centers for Medicare & Medicaid Services (CMS) web site: http://cms.hhs.gov.

In addition, the number of elementary and secondary schools in an area can indicate the demographic make up of a ZIP code, the population density of an area, or even the affluence level of the ZIP code. The National Center for Education Statistics web site (http://nces.ed.gov/ccd/) provided a file containing a listing of each elementary and secondary school in the United States and detailed information about each. They call this "The Common Core of Data (CCD)." These data were aggregated to a ZIP code level and used in our model.

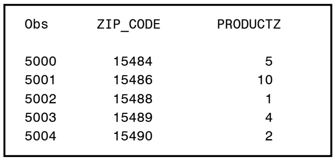
CUSTOMER COUNTS AND PENETRATIONS

We have found that aggregating counts of customers by product to a geographic level can be very helpful in developing models. If a large percentage of people have purchased certain products in an area, then that product must be attractive to those people for a reason. We would then continue to market that product in that area and look to find other similar areas to market the product.

Extracting counts from a customer base can be done in many different ways. One possible way to extract counts by product is by running the SQL procedure to count up the number of people in each geographic area, which is ZIP code in our case. An example of that code is as follows:

```
%macro counts(product,prod);
proc sql;
create table customer_counts_&prod. as
select
ZIP_code, count(*) as product&prod.

from datalib.customer_base
where product_type=&product.

group by ZIP_code
order by ZIP_code;

proc print data=customer_counts_&prod. (obs=20);
   title "ZIP Code Counts for Product &prod.";
run;
%mend;

***Call the macro for all product types;
%counts('Y',Y)
%counts('X',X)
%counts('Z',Z)
```

Below is an example of how each data set will look.



```
Obs      ZIP_CODE      PRODUCTZ

5000      15484               5
5001      15486              10
5002      15488               1
5003      15489               4
5004      15490               2
```

After extracting this data into a number of separate data sets, you could join or merge them together into one master data set containing a count of customers by product for every ZIP code. You can also make penetrations out of a "total in group" variable or append U.S. population data from the census to calculate population penetrations. An example of code to perform the merge and penetration calculations is below.

```
data all_customer_counts (drop = i j);
  merge customer_counts_X
```

```
        customer_counts_Y
        customer_counts_Z;
  by ZIP_code;
  array variable{3} productX productY productZ;
  do i=1 to 3;
    if variable{i}=. then variable{i}=0;
  end;
  total_insured=sum(productX, productY,
productZ);
  array penetration{3} productX_pen productY_pen
productZ_pen;
  do j=1 to 3;
    penetration{j}=variable{j}/total_customers;
  end;
```

Below is an example of how your data set should look.

```
                                                total_  product  product  product
 Obs   ZIP_CODE PRODUCTX PRODUCTY PRODUCTZ insured   X_pen    Y_pen    Z_pen

5000    14897         4        6        0       10  0.40000  0.60000  0.00000
5001    14898         9       14        0       23  0.39130  0.60870  0.00000
5002    14901       109      171        1      281  0.38790  0.60854  0.00356
5003    14902         3        5        0        8  0.37500  0.62500  0.00000
5004    14903        67      123        0      190  0.35263  0.64737  0.00000
```

PURCHASED DATA

Many vendors can provide data on the ZIP code or ZIP code plus 4 level. One web site that contains a list of geographic level data available for purchase is http://www.esri.com/. In the past, Experian/Choice Point has provided us with credit statistics on the ZIP+4 level, which we often use in modeling. Credit statistics can be very predictive data elements in response models for certain products, and we found that they were helpful in segmenting on the ZIP code level as well.

HOW MANY SEGMENTS DO WE WANT?

There are many things to consider when deciding how many groups are sufficient to describe your population. The first consideration is how many groups your modeling and data storage systems can handle. Many companies have space and access restrictions, which may limit the number of clusters that could be developed.

Second, how deeply would you like to (or can you) segment your database? A small database or a database containing many homogeneous people may not need to be clustered into a large number of different segments. You may find that you do not have sufficient data or data that varies enough to divide your sample into a large number of groups. It also may be difficult to manage many segments and to strategize based on the various groups. On the other hand, if you have a very diverse database, you may want a large number of segments.

Third, you need to consider the level of geography you are working with. The more detailed the level of geography, the more clusters you may be able to find. For example, there are over 32,000 ZIP codes in America, but over 200,000 Census block groups. Therefore, block group level data will probably allow you to find more distinct segments since each segment contains fewer households and is, hopefully, more homogeneous. Another approach is creating clusters within clusters. This will not be discussed in this paper; however, many segmentation systems available for purchase use this clustering approach. They may have only ten or fewer main segments, but within each of the main groups are smaller groups, resulting in a large total number of groups.

In ZIP code clustering, we wanted to be able to create around ten different segments. We felt that this would give us enough homogeneity within the clusters, but that it would also clearly show us patterns within our groups.

THE ART OF CLUSTERING

The main objective for clustering is to find the best natural grouping structure among a set of observations. Objects in the same cluster should be more similar to each other than they are to objects in other clusters. It is important to note that there is no unique clustering solution for every data set. The "best" clustering solution is not determined by statistical output alone, but is determined through profiling a number of solutions and determining the solution with the most practical importance. In this way, clustering is "purely an exploratory technique...[and] is more of an art than a science" (Hair, Anderson, Tatham, and Black 1998).

Knowing how you want to use the clusters and knowing what data you have available is all you need to begin the clustering process. Once these two questions have been answered, there are a few different paths that you can take in order to create your segmentation system.

There are two primary ways of performing cluster analyses: hierarchical grouping and nonhierarchical or direct grouping (Johnson 1998). We decided to use a nonhierarchical method. This method is much easier to use when attempting to cluster large data sets, like our ZIP code data set with over 32,000 observations. The SAS procedure associated with this type of clustering is FASTCLUS. This procedure is equipped to deal with a larger number of items, and works well if you have a good estimation of how many clusters would best describe your population (Hamer 1997).

REVIEWING YOUR DATA

It is very important that you start with a clean data set. Running diagnostics on your data to see the distribution of each variable, amount of missing data, minimum and maximum values, and outliers is a very important step in the preparation process. We suggest using PROC UNIVARIATE to check for outlier observations and missing data elements. It is necessary that you handle missing data appropriately. If a value for one or more data elements is missing, the entire ZIP code will not be included in the analysis. You will also want to look for data elements that are possibly incorrect. For example, you could find that the number of people owning a specific product in the ZIP code is larger than the number of people supposedly living in that ZIP code. This may happen if your database uses default ZIP codes for those people who do not have a ZIP code on file. You may choose to omit incorrect or outlier ZIP codes from your analysis, which would mean that those ZIP codes would not be clustered. Another option would be to set anomalous values of a certain variable to zero or to default them to the overall average.

FACTOR ANALYSIS

It is possible that some of the data elements you would like to use for clustering are correlated with each other. As we mentioned above, one option is to use only one of the correlated data elements in creating your clusters. You may decide to choose a single variable based on prior research or simply based on ease of explanation to the user.

We suggest that, when practical, the analyst run a factor analysis on a preliminary set of variables. This analysis will serve several purposes. First, it will help to identify the level of correlation between the selected data elements. Second, if the analyst wants to choose only the variable that is most representative of the group of variables to use in the cluster analysis, the factor loadings will help him or her pick the most appropriate variable. Finally, if the analyst wishes to combine the correlated data elements together into a single composite variable, this is easily accomplished using the output from a factor analysis. The SAS procedure associated with factor analysis is PROC FACTOR.

In order to run a factor analysis you need to make sure there are sufficient correlations in your data (Hair, Anderson, Tatham, and Black 1998). The MSA option in the PROC FACTOR statement runs a test called the Measure of Sample Adequacy that checks for the amount of intercorrelations among the variables. This statistic ranges from 0 to 1. If the resulting MSA is less than 0.5, there is not enough correlation between your data elements to proceed with a factor analysis and you can skip this step in the clustering process (Hair, Anderson, Tatham, and Black 1998).

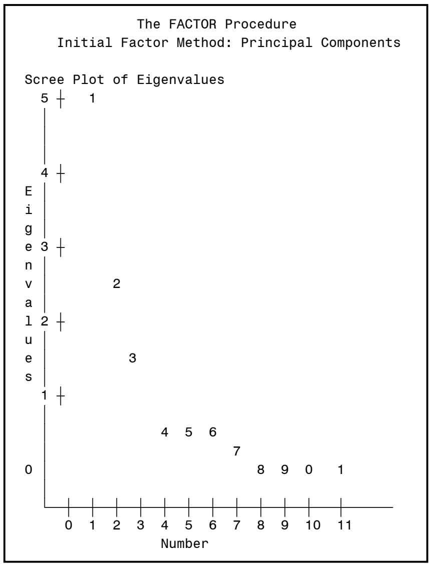
In order to determine the number of factors that would best describe the correlation within your candidate variables, you can run the procedure several times until you find a solution that meets your needs. In order to decide how many factors are appropriate for your solution, one of several different methods may be utilized. We have found that a solution that explains a good portion of the variance but does not contain extraneous factors is best. In the example below, we would probably choose three factors, which explain a cumulative total of 80 percent of the total variation, but we would run the procedure again selecting four and five factors to make sure that neither of the additional factors explained a unique component of the data.

```
                         The FACTOR Procedure
              Initial Factor Method: Principal Components

  Eigenvalues of the Correlation Matrix: Total = 11  Average  = 1

         Eigenvalue    Difference    Proportion    Cumulative

  1     4.93686572    2.57013865        0.4488        0.4488
  2     2.36672706    0.85610847        0.2152        0.6640
  3     1.51061860    0.98805212        0.1373        0.8013
  4     0.52256648    0.07020195        0.0475        0.8488
  5     0.45236453    0.01958271        0.0411        0.8899
  6     0.43278182    0.08853880        0.0393        0.9293
  7     0.34424302    0.13591426        0.0313        0.9606
  8     0.20832875    0.07621719        0.0189        0.9795
  9     0.13211156    0.04596481        0.0120        0.9915
  10    0.08614675    0.07890104        0.0078        0.9993
  11    0.00724571                      0.0007        1.0000

  3 factors will be retained by the MINEIGEN criterion.
```

Another way to determine the number of factors is by setting the MINEIGEN= option in the PROC FACTOR statement. This allows you to specify the minimum eigenvalue that you want to use for creating a new cluster. The default for the MINEIGEN= option is 1 because a single variable will have an eigenvalue of 1, so we want any factor to at least contain as much information as a single variable. In our example above, that would limit the number of factors to three.

A third way to determine the optimal number of factors of your variables is to examine a plot called a scree plot as shown in Figure 1. The SCREE option in the PROC FACTOR statement will produce a scree plot. The analyst can examine the plot to determine the point before the pattern flattens out. The analyst should set the number of factors to be the number indicated at that point. The plot on the left would suggest considering three factors.

Figure 1



```
              The FACTOR Procedure
    Initial Factor Method: Principal Components

Scree Plot of Eigenvalues
  5 ┼   1
    │
    │
    │
  4 ┼
E   │
i   │
g   │
e 3 ┼
n   │
v   │      2
a   │
l 2 ┼
u   │
e   │        3
s   │
  1 ┼
    │
    │            4  5  6
    │                     7
0   │                        8  9  0   1
    │
    └┼──┼──┼──┼──┼──┼──┼──┼──┼──┼──┼──┼
     0  1  2  3  4  5  6  7  8  9  10 11
                 Number
```

If you have run the procedure several times, have examined your Scree Plot, and know the number of factors you want, you can tell the procedure how many factors to produce using the NFACTORS option. In our example, we ran the procedure several times with two, three, four, and five factors. We chose our best solution based on additional output from each of those runs.

PROC FACTOR's many options allow the statistician to analyze a variety of solutions and diagnostic output. One option we always include is CORR. This will print the correlation matrix for your variables, which is valuable information even if you do not end up including factored variables in your cluster analysis. Below is an example of PROC FACTOR code.

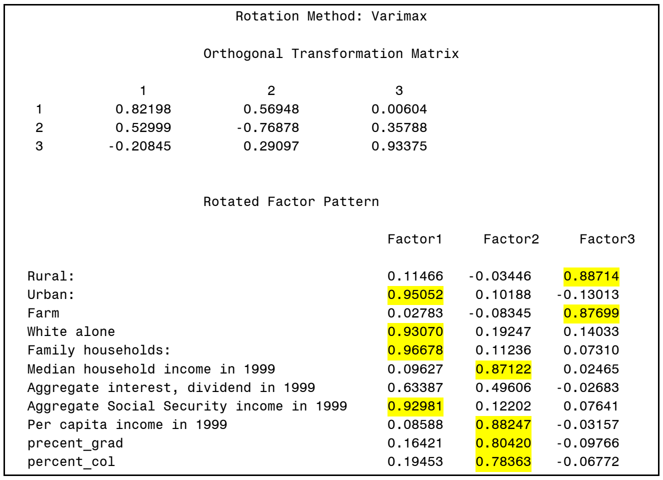
```
proc factor data=census.test corr msa scree
rotate=varimax;
var p005005 p005002 p005006 p006002 p013002;
run;
```

Another important option of PROC FACTOR that we have not yet discussed is the ROTATE= option. Initial factors will not be created where only a few variables have very high correlations; rather many variables will tend to have only somewhat high correlations. This will make the results fairly difficult to interpret, and it will be virtually impossible to choose only one variable to represent your factor. "Rotating" the factors will help to create higher correlations on certain variables within each factor. That makes it easier to select the main variables from each factor and comprehend the "definition" of each factor. Factor analysis will, by default, create orthogonal (or completely independent) factors. Several methods of rotating the factors will instead produce oblique (or dependent) factors. This may be helpful in interpreting your clusters, as it will usually create even more dramatic results. Nevertheless, if you are using the results of factor analysis to create new variables for your clustering procedure, we recommend that an orthogonal rotation method be used. The orthogonal rotation method we suggest is VARIMAX, as it creates factors that are the most easily interpreted.

The concept of rotation can be difficult to grasp. The output below shows how rotation can be helpful. In our example, we chose the three-factor solution. Factor 1 has high factor loadings for virtually all variables, while Factor 2 has lower loadings for almost every variable, making it difficult for the analyst to determine what characteristic(s) either factor is describing.

```
                               The FACTOR Procedure
                    Initial Factor Method: Principal Components

                                  Factor Pattern

                                                   Factor1     Factor2    Factor3

Rural:                                             0.07999     0.40475    0.79444
Urban:                                             0.83854     0.37887    0.29000
Farm                                              -0.06510     0.36326    0.80041
White alone                                        0.87548     0.39552    0.00697
Family households:                                 0.85910     0.45217    0.10057
Median household income in 1999                    0.57542    -0.60993    0.25645
Aggregate interest, dividend income in 1999        0.80337    -0.05502    0.01285
Aggregate Social Security income in 1999           0.83424     0.42633    0.08697
Per capita income in 1999                          0.57294    -0.64421    0.20939
precent_grad                                       0.59236    -0.56618    0.10859
percent_col                                        0.60575    -0.52358    0.12423
```

However, once we rotate the factors, the patterns become clearer. The rotated factor patterns are displayed on the following page. In examining these patterns, the analyst may conclude that Factor 3 indicates rural ZIP codes with a presence of farms, Factor 2 describes affluence, and Factor 1 highlights areas with high population density, possibly cities.



```
                       Rotation Method: Varimax

                    Orthogonal Transformation Matrix

             1              2              3
  1        0.82198        0.56948        0.00604
  2        0.52999       -0.76878        0.35788
  3       -0.20845        0.29097        0.93375


                    Rotated Factor Pattern

                                             Factor1     Factor2     Factor3

 Rural:                                      0.11466    -0.03446     0.88714
 Urban:                                      0.95052     0.10188    -0.13013
 Farm                                        0.02783    -0.08345     0.87699
 White alone                                 0.93070     0.19247     0.14033
 Family households:                          0.96678     0.11236     0.07310
 Median household income in 1999             0.09627     0.87122     0.02465
 Aggregate interest, dividend in 1999        0.63387     0.49606    -0.02683
 Aggregate Social Security income in 1999    0.92981     0.12202     0.07641
 Per capita income in 1999                   0.08588     0.88247    -0.03157
 precent_grad                                0.16421     0.80420    -0.09766
 percent_col                                 0.19453     0.78363    -0.06772
```

Once you have determined the number of factors you would like to use to describe your overall set of candidate variables, you can rerun PROC FACTOR with an OUT= option. This will create an output data set in which each of your observations will be scored with your factors. You can use these scores as inputs to the clustering procedure. In the output dataset they will be called *factor1, factor2*, and so on. Please note that you must tell SAS the number of factors you would like to create in order for it to write an output dataset.

```
 proc factor data=census.test corr msa scree
rotate=varimax out=census.factors nfactor=3;
var p005005 p005002 p005006 p006002 p013002;
run;
```

We suggest that you rename these factors to more descriptive names such as *Affluence_Component* or *Presence_of_Farms* so that you will be able to accurately interpret their impact on your clusters.

STANDARDIZING YOUR DATA

Most likely, the data that you collected to use for clustering is in many different formats and encompasses a wide range of numbers. For example, one variable may be in percentage format, and is a number between 0.00 and 1.00. Another variable could be describing the average income of an area, and might range from $0 to $500,000. If you were to use the data as it is, the income variable would have much more weight than the percentage variable because the values are larger (Johnson 1998). If the income variable has the largest values on your data set, it could completely determine the breaks in your clusters.

To eliminate this problem, you can standardize your data using the STANDARD procedure. PROC STANDARD will replace missing values and standardize your data based on specified values.

To replace missing values with a number, you will use the REPLACE option. With no other options, it will replace missing values with the variable mean. If you use the MEAN= option with a specific numeric value, it will replace all missing values with that value. The PRINT option will print statistics for each variable that will be standardized. An example of the SAS code is below.

```
proc standard data=all_data out=new_data replace
print;
   var percent1 percent2 income1 credit1
penetration1 penetration2;
   run;
```

PROC STANDARD code without the REPLACE option will overwrite all original values of the variables percent1, percent2, penetration1, penetration2, income1 and credit1 with the standardized values of the variables. If you want to keep the original unstandardized variables on your data set to use for analysis after clustering, you should create a second copy of these variables on that data set before standardizing.

```
Data new_data2;
   set new_data;
   sd_percent1=percent1;
   sd_percent2=percent2;
   sd_penetration1=penetration1;
   sd_penetration2=penetration2;
   sd_income1=income1;
   sd_credit1=credit1;
```

To standardize variables around a specific mean with a specific standard deviation, you will not use the REPLACE option, but instead, you will include the MEAN= and STD= options. To scale every variable the same way, center all variables on the same mean and standard deviation. In most cases, it is best to convert each variable to a standard normal distribution. Some clustering methods, including the methods that we have chosen for this paper, Euclidean distances, assume that the data have a standard normal distribution. The code below tells SAS to subtract the mean of the variable from the observation and divide by the standard deviation, in effect centering the variable around a mean of 0 and a standard deviation of 1.

```
proc standard data=new_data2 out=final_data
mean=0 std=1;
   var sd_percent1 sd_percent2 sd_income1
sd_credit1 sd_penetration1 sd_penetration2;
```

On your normalized data set, values near zero vary closely with the mean of the variable before standardizing. Observations with negative values are below the mean of the variable and observations with positive values are above the mean of the variable.

Your data set should now be ready for clustering.

THE FASTCLUS PROCEDURE

PROC FASTCLUS not only finds a starting point for the clustering algorithm (called the cluster seeds), but it also uses a "standard iterative algorithm for minimizing the sum of squared distances from the cluster means" (Arnold and Stokes 1999). This means that it will try to find clusters where the distance between each observation and the cluster centroid is small but where the distances between the cluster centroids are the largest.

PROC FASTCLUS allows many ways of finding your cluster solution. Different combinations of procedure options can give you a variety of cluster solutions to choose from. First, you can tell PROC FASTCLUS how many clusters you would like in your solution by specifying a maximum number of clusters with the MAXCLUSTERS= option. You can also specify how far apart you want the initial cluster seeds to be by using the RADIUS= option. You can use a combination of both options as well. The default number of clusters you will attain if you do not use the MAXCLUSTERS= option is 150. If you specify both options, the FASTCLUS procedure will pick initial seeds with at least your specified radius between them.

Next, you can also tell FASTCLUS how it should choose your initial seeds with the REPLACE= option. The selection of the initial seeds will significantly impact your final clustering solution. Using the default, or REPLACE=FULL option, the order of the data will determine the seeds. For example, if you want 20 clusters, it will use the first 20 observations in the data set that satisfy all of your criteria as the initial seeds. Another method is the REPLACE=RANDOM option, which will tell PROC FASTCLUS to choose the initial seeds randomly from the data set. Every time you run the procedure with the REPLACE=RANDOM option, you will find a different solution because it uses different seeds. You can specify your random seed with the RANDOM= option as well, which is useful when attempting to duplicate a solution. In addition, you can tell PROC FASTCLUS how distant or different you want the clusters to be with the REPLACE=FULL option and a larger RADIUS. However, you cannot set the RADIUS option when using the REPLACE=RANDOM. There is no "correct" way of choosing initial seeds; rather, you should evaluate more than one result to determine the best solution for your situation.

There are advantages and disadvantages to using each option. If you have outliers in your data set, they can significantly impact the seed selection and therefore the entire cluster solution. If the outliers represent a valid component of the sample, they should be included in your data set. If they do not, you should delete them, and then you must rerun your procedure to compensate for the differences in your data set. In our ZIP code data set, we had outlier observations due to the use of group quarters information. ZIP codes with large military bases and large universities tended to fall into a couple of very small clusters. We suggest that you run the procedure several times, changing the procedure options each time, to find your optimal cluster solution.

The initial cluster seeds that PROC FASTCLUS chooses are most likely not the best cluster centroids. Using the MAXITER= option, you can specify the number of iterations through which you want the algorithm to run to be sure that it will settle on the best centroids. Choosing a large number of iterations (around 100-200) will ensure that it will run through enough iterations to converge on the best centriods. The algorithm will stop when the cluster seeds have stopped changing between iterations. To be sure that the algorithm converges and the final cluster seeds equal the cluster means, use the CONVERGE=0 option. In addition, to tell PROC FASTCLUS to recalibrate the cluster seeds with the new cluster means after each iteration, use the DRIFT option. Below is an example of the PROC FASTCLUS code.

```
%let varlist=sd_percent1 sd_percent2 sd_income1
sd_credit1 sd_penetration1 sd_penetration2;

proc fastclus data=final_data maxclusters=20
replace=random maxiter=100 converge=0 drift
distance
mean=cluster_means out=cluster_output;
       var &varlist;
run;
```

The MEAN= option allows you to create a data set with the final means of each cluster for every variable that was input into the procedure. The OUT= option creates an output data set containing the entire input data set, a variable called *distance* that is the distance between the observation and the cluster centroid, and a variable called *cluster* that gives the number of the cluster that the observation fell into. The DISTANCE option computes distances between each cluster mean and is a good way of comparing your clusters to each other in the output. There are many other options available in PROC FASTCLUS, but these are the options that we feel are necessary to create good geographic clusters. Please refer to the SAS documentation for information on other procedure options.

READING THE OUTPUT

The output from PROC FASTCLUS contains a wealth of information that can help you to decide if you have found the optimal solution for your problem. The output will contain:

- The values of each variable for the initial seeds used
- The minimum distance between initial seeds
- The Iteration History, which will show you the amount of relative change in the cluster seeds
- A Cluster Summary similar to that shown below

(Some column names have been abbreviated to fit our paper.)

```
                              Cluster Summary

                                   Max Dist
                     RMS Std      from Seed   Radius   Nearest  Distance Between
Cluster   Freq      Deviation       to Obs   Exceeded  Cluster  Cluster Centroids
--------------------------------------------------------------------------------
    1    10021        0.2827        6.5065                 17         1.0073
    2     4278        0.3550        6.5693                  9         1.9972
    3     7148        0.2399        6.4085                  1         1.2573
    4     7238        0.3144        6.4908                 20         1.1778
```

- Statistics for Variables, including those shown below

```
                         The FASTCLUS Procedure
     Replace=RANDOM  Drift  Radius=0  Maxclusters=20 Maxiter=100
                              Converge=0

                     Pseudo F Statistic = 12566.87
         Approximate Expected Over-All R-Squared =   0.55644
              Cubic Clustering Criterion =  409.997
```

- A Pseudo F-Statistic
- An Approximate Expected Overall R-squared
- The Cubic Clustering Criterion (or CCC); a good solution would have a CCC over 3.

```
                       Statistics for Variables

Variable      Total STD    Within STD    R-Square    RSQ/(1-RSQ)
------------------------------------------------------------------
sd_percent1     1.00407       0.28674    0.918460      11.263873
sd_percent2     0.99843       0.43516    0.810079       4.265333
sd_income1      0.99959       0.52285    0.726452       2.655664
sd_credit1      1.00070       0.47640    0.773399       3.413050
```

The top of each page of output from PROC FASTCLUS contains a header that lists all options that were used when the procedure was run. With these options, we obtained a Pseudo F Statistic that is extremely high. Our Cubic Clustering Criterion is well above 3, so we can say that this clustering solution is good. These statistics are primarily used to compare the cluster solutions generated when using different options in the procedure. As a general rule, you will want to consider solutions with the highest values for all three of these statistics.

- Cluster means and standard deviations for each variable

```
                           Cluster Means

Cluster   sd_percent1    sd_percent2    sd_income1     sd_credit1
-----------------------------------------------------------------
    1     -0.06499711    -0.58701679    -0.17950219    -0.80023634
    2     -0.06372894    -0.46876909     1.82396449    -0.41297967
    3     -0.06286305    -0.67608011    -0.41784308    -0.76027134
    4     -0.06292848     0.84408606    -0.16439056    -0.77268516


                     Cluster Standard Deviations

Cluster    sd_percent1   sd_percent2   sd_income1    sd_credit1
---------------------------------------------------------------
    1      0.139992542   0.347590817   0.518054810   0.366450045
    2      0.165958998   0.387469383   0.607362896   0.643014948
    3      0.181548856   0.333455614   0.495453272   0.368051443
    4      0.180417387   0.299385038   0.498704634   0.382692906
```

COMPARING CLUSTERS

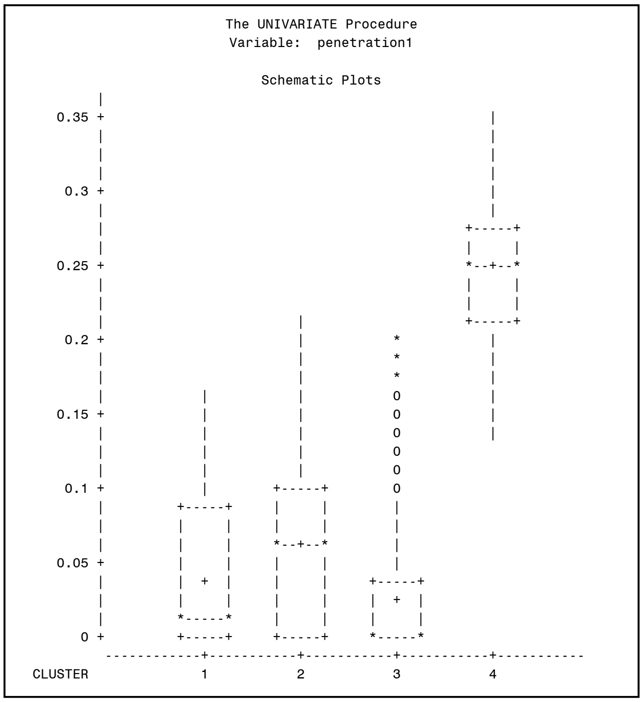
After you have settled on a cluster solution that seems optimal, you will want to compare your clusters to each other to determine if you have found clusters that are very different from each other. You can use PROC UNIVARIATE to compare the characteristics of the clusters to each other. The code below compares the values of variables *income1* and *penetration1* for clusters 1, 2, 3, and 4 to each other. The data must be sorted by the by-variable in order to use the **by** statement in PROC UNIVARIATE.

```
proc univariate data=cluster_output plots;
   by cluster;
   var penetration1 income1;
   where cluster in (1,2,3,4);
```

The PLOTS option will tell SAS to include a series of graphs in the output to help explain the data. With this option, in addition to the default statistics, you will see a histogram, a box-plot and a normal probability plot. At the end of the output, you will see a series of schematic plots, which include box plots like those in Figure 2 and Figure 3 that compare the clusters by variable.

The first plot, shown in Figure 2, illustrates that the average value of the variable *penetration1* (the number of people with product1 divided by the total population) is much higher for cluster 4 than for the other three clusters. High values of *penetration1* indicate that ZIP codes included in cluster 4 may be excellent ZIP codes to continue to target with both advertising and direct mail efforts for product1. Extensive profiling can help to determine how to strategize based on your findings.

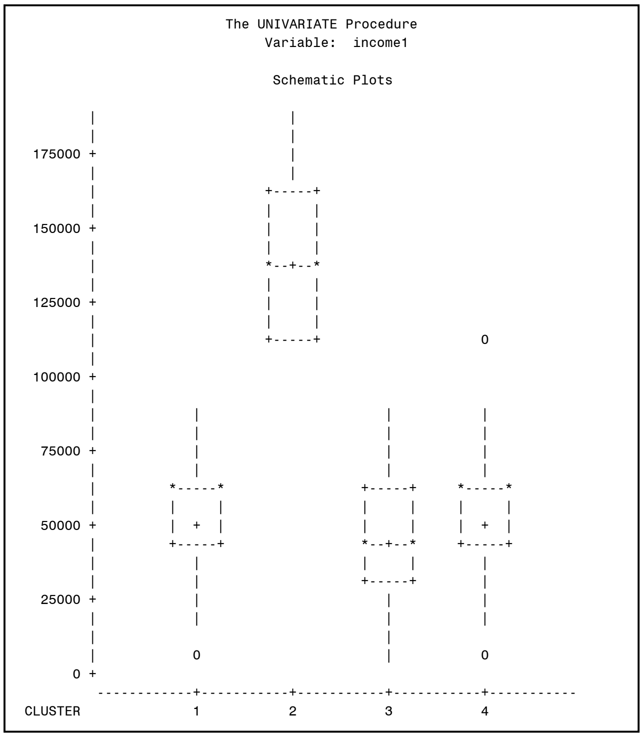
FIGURE 2:



```
                        The UNIVARIATE Procedure
                         Variable:  penetration1

                             Schematic Plots
         |
    0.35 +                                                |
         |                                                |
         |                                                |
         |                                                |
     0.3 +                                                |
         |                                                |
         |                                             +-----+
         |                                             |     |
    0.25 +                                             *--+--*
         |                                             |     |
         |                        |                    |     |
         |                        |                    +-----+
     0.2 +                        |           *           |
         |                        |           *           |
         |                        |           *           |
         |            |           |           0           |
    0.15 +            |           |           0           |
         |            |           |           0           |
         |            |           |           0
         |            |           |           0
     0.1 +            |        +-----+        0
         |         +-----+     |     |        |
         |         |     |     |     |        |
         |         |     |     *--+--*        |
    0.05 +         |     |     |     |        |
         |         |  +  |     |     |     +-----+
         |         |     |     |     |     |  +  |
         |         *-----*     |     |     |     |
       0 +         +-----+     +-----+     *-----*
          ------------+-----------+-----------+-----------+-----------
 CLUSTER              1           2           3           4
```

The box-plot in Figure 3 shows that cluster two contains ZIP codes with higher average income. Separating the ZIP codes with higher average income from those with low-to-mid income ranges can be important when creating marketing strategies. Areas where people are wealthier may be areas where people are likely to be more educated and working full time. This may make them prime prospects for some product offers and poor prospects for others. Reviewing the product penetrations by cluster is one way to tell which clusters will tend to respond to each product offer. Once again, extensive profiling of the characteristics of these clusters should be completed in order to determine the best way to address each cluster.

FIGURE 3:



```
                        The UNIVARIATE Procedure
                           Variable:  income1

                             Schematic Plots

         |                        |
         |                        |
  175000 +                        |
         |                        |
         |                     +-----+
         |                     |     |
  150000 +                     |     |
         |                     |     |
         |                     *--+--*
         |                     |     |
  125000 +                     |     |
         |                     |     |
         |                     +-----+                    0
         |
  100000 +
         |
         |            |                       |           |
         |            |                       |           |
   75000 +            |                       |           |
         |            |                       |           |
         |         *-----*                 +-----+     *-----*
         |         |     |                 |     |     |     |
   50000 +         |  +  |                 |     |     |  +  |
         |         +-----+                 *--+--*     +-----+
         |            |                    |     |        |
         |            |                    +-----+        |
   25000 +            |                       |           |
         |            |                       |           |
         |                                    |
         |            0                       |           0
       0 +
          ------------+-----------+-----------+-----------+-----------
 CLUSTER              1           2           3           4
```

HOW TO HANDLE OUTLIER CLUSTERS

Depending on which options you used in your FASTCLUS procedure, you may notice that you have a number of clusters with only a few observations. These outlier clusters will most likely contain observations with "extreme" values for one or more of the variables, and thus, they do not fit into any of the larger clusters. We found working with these clusters to be a challenge, so we have detailed a few approaches below that you can use to manage outlier clusters.

- Try using the REPLACE=RANDOM option to obtain a few different solutions. However, it may be difficult for the procedure to converge if the seeds that were chosen were not near the outliers. You could also separate the outliers from the main data set, creating an "Extreme/Outlier" cluster for them, and then rerun the procedure with the REPLACE=RANDOM option to get a new solution.
- You could try to run the procedure with the REPLACE=FULL option and a smaller number of MAXCLUSTERs to see if the outlier clusters would combine with any of the larger clusters. Often, this will only join the larger clusters together, and you will end up with only a few extreme clusters. The distances between the larger clusters and the outlier clusters are often so large that they will not ever cluster together.
- Outlier clusters with only a few observations could be added to the nearest cluster. Most likely, if the sample sizes of your other clusters are large enough, the outliers will not skew the data in the larger cluster.
- Create an "Extreme/Outlier" cluster, in which you will combine all clusters that have only a few observations. This could enable you to identify areas that may have unusual characteristics. This was the option that we chose for our ZIP code clusters.

PROFILING AND VALIDATING CLUSTERS

After you have created your clusters, performed preliminary comparisons, and chosen an optimal cluster solution, you should profile them more thoroughly with additional data you have about each area. You already know the distribution of the variables you used in creating your clusters, but you should also identify other significant characteristics of the clusters. You will also want to determine the practical significance of the clusters to your application. You can validate the clusters using campaign response information and examine how your current customer base and new responders are distributed within the ZIP code clusters. We also suggest mapping your clusters using SAS/GIS, the SAS Bridge to ESRI or another mapping package. Mapping with the SAS Bridge to ESRI for this purpose will be explained later in this paper in the Geographically Displaying your Clusters section.

In completing this analysis, some examples of questions you may seek to answer are the following:

- Do people living in certain ZIP code clusters respond better to direct mail campaigns or to advertisements?
- Do people in cluster X prefer certain products over others?
- Do people in cluster Y need products based on the environment or their lifestyle?

Finally, if you find that your clusters do not discriminate well between your customers after you have finished profiling, you may want to modify your clusters. You can do this by adding or subtracting variables or changing the number of clusters. A larger number of clusters will have more distinct centroids, but you may find that they are difficult to manage and use in designing product offers.

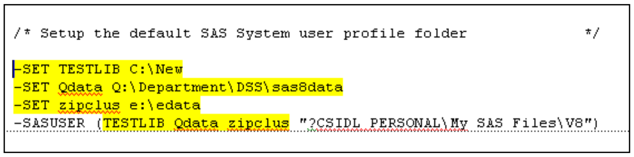
GEOGRAPHICALLY DISPLAYING YOUR CLUSTERS

Once you have decided on your final set of clusters, you will want to display them graphically in various ways to assist in interpretation and to create a comfort level with the marketers who will be using them. Mapping your clusters, both on a National level and in target areas of interest, is a great to do this. Mapping in SAS/GIS can be cumbersome and inflexible. We have found that ESRI's ArcGIS software can be more user-friendly for displaying geographic information and will create more visually pleasing maps. In the past, we needed to convert our SAS data sets into dBase format for mapping. Recently, SAS and ESRI built a bridge between the two systems to facilitate the mapping of SAS datasets.

To begin, merge any data from SAS data sets that you may want to display on your maps to your output dataset from PROC FASTCLUS. (If, after starting your map, you think of an additional data element you would like to add, you can join data to your main data set in ArcGIS as well.) Once your dataset is ready, you need to be sure that the bridge between the systems will know where to find the data set. The bridge default will only give you access to SAS data sets in SASHELP, MAPS, and SASUSER directories in SAS. If it is not convenient for you to add your mapping data sets to one of the three default areas that the Bridge to ESRI allows you to use, you will need to add directories to the SASV8.CFG file. You can reconfigure the SASV8.CFG file to tell SAS to include other folders under the SASUSER "umbrella." This file is usually located on your computer at

C:\Program Files\Sas Intitiute\SAS\v8\sasv8.cfg.

Below is an excerpt from this file. This example shows three directories where we can put SAS data for mapping: on our hard drive C:\ in TESTLIB, on a CDROM in drive E:\ in ZIPCLUS, and on network drive Q:\ in QDATA. We strongly urge that you make a backup copy of the SASV8.CFG file before editing it!



```
/* Setup the default SAS System user profile folder          */

-SET TESTLIB C:\New
-SET Qdata Q:\Department\DSS\sas8data
-SET zipclus e:\edata
-SASUSER (TESTLIB Qdata zipclus "?CSIDL PERSONAL\My SAS Files\V8")
```

Once the libraries are assigned in your configuration file, ArcGIS will be able to find them.

Next, since SAS and ESRI share data using an OLE protocol, you will need to add each SAS library as an OLE database. You will do this in ArcCatalog. Below are summarized instructions.

- Open ArcCatalog.
- Double-click **Add OLE DB Connection**.
- Select **SAS IOM Data Provider 9.0** from the Provider tab of the Data Link Properties window.
- Click the Connections tab and type **_local_** in the Data Source field. Click OK.
- Right-click **OLE DB Connection** and select Connect from the pop-up menu. Click the **Add Data** icon.

Next you will want to add the data to a blank map. Open ArcMap and click on "New Empty Map." Add your SAS data set to your map by following these instructions:

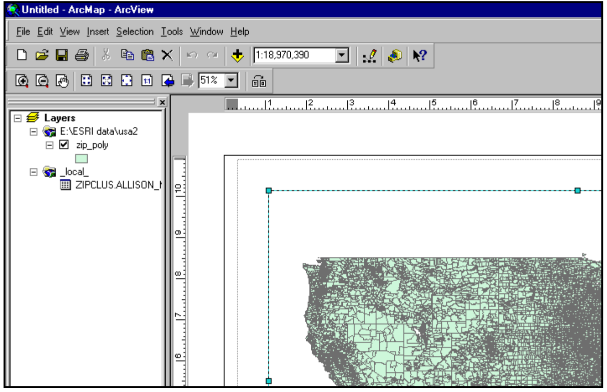
- Click the **Add Data** icon.
- Select **OLE DB Connection.odc** from the drop-down list of sources.
- Select the SAS data set to be added.
- Click **Add**.

A full instruction set for making the connection between SAS and ESRI using the OLE protocol are available on the SAS web site at the following URL:
http://support.sas.com/rnd/datavisualization/BridgeForESRI/V1/

You are now ready to create your map. Maps in ArcGIS start with a shape file as the base. A shape file contains coordinate information about geographic areas so that ArcMap can draw the map. It is any file that ends with the extension .SHP. Your shape and layer files should all be located together within one or two directories on your computer. You can also download shape and layer files from the Internet if the specific ones you are looking for are not already on your computer.

From ArcCatalog, drag the shape file onto either the blank map in ArcMap or under "Layer" in the Layers window of ArcMap. A map will appear in the larger mapping window. For ZIP code mapping, we use the ZIP code shape file called zip_poly.shp. An example of a new map including the ZIP code shape file and the SAS data that can be displayed on the map is shown in Figure 4.

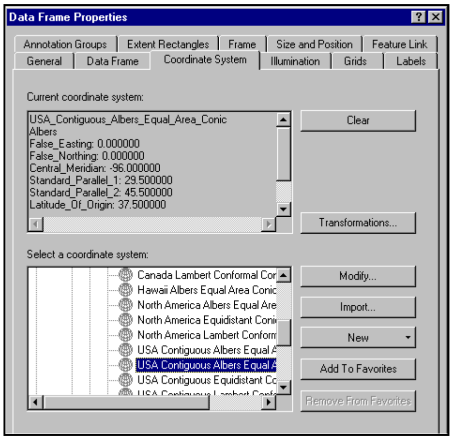
Figure 4



When you add the shape file to the empty map, you will probably find that the map of the United States looks sort of flat and long. This is because you have not yet chosen a projection for the map. To change the projection, right click on top of the map and then click Properties. Go to the Coordinate System tab. Our favorite projections are found by clicking Predefined, then

Projected Coordinate Systems, Continental, and North America. Then, finally, choose one of the USA Contiguous Albers Equal Area projections. This is shown below in Figure 5.

Figure 5



Data Frame Properties

Current coordinate system:

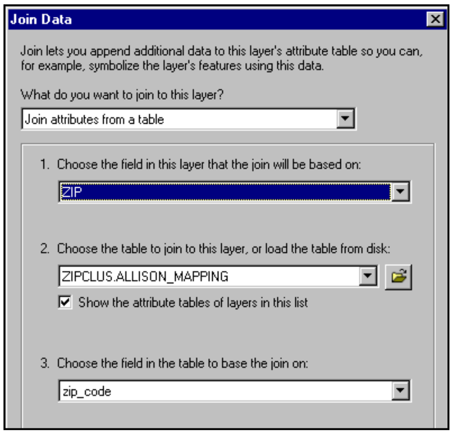
USA_Contiguous_Albers_Equal_Area_Conic
Albers
False_Easting: 0.000000
False_Northing: 0.000000
Central_Meridian: -96.000000
Standard_Parallel_1: 29.500000
Standard_Parallel_2: 45.500000
Latitude_Of_Origin: 37.500000

Select a coordinate system:

Canada Lambert Conformal Cor
Hawaii Albers Equal Area Conic
North America Albers Equal Are
North America Equidistant Coni
North America Lambert Conform
USA Contiguous Albers Equal A
USA Contiguous Albers Equal A
USA Contiguous Equidistant Cc

Now you must join your SAS data to the shape data in ArcMap. The easiest way to do that is by right clicking on the shape file's name in the Layers window and then by clicking "Joins and Relates." Then click "Joins," which will open another window. The default in this window should be "Join attributes from a table," which is what you want to do for a map of this type. In the "Choose the field in this layer that the join will be based on" box, choose the field from the shape file that you would like to use to join your data to the map, *ZIP* in our case. In the next box, choose the SAS data set to join to, and then finally, in the final box, choose the field that you want to join the map to on the SAS data set, *zip_code* in our example. The fields that you are using to join should match in type, although their names may not. The dialog box for joining your data to the shape file is shown in Figure 6.

Figure 6



Join Data

Join lets you append additional data to this layer's attribute table so you can, for example, symbolize the layer's features using this data.

What do you want to join to this layer?

Join attributes from a table

1. Choose the field in this layer that the join will be based on:

   ZIP

2. Choose the table to join to this layer, or load the table from disk:

   ZIPCLUS.ALLISON_MAPPING

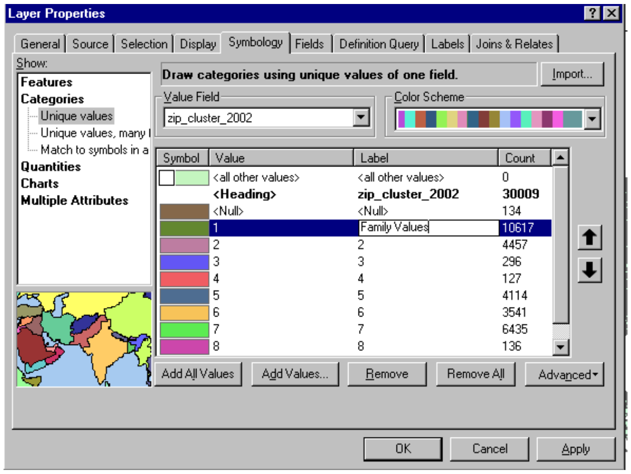
   ☑ Show the attribute tables of layers in this list

3. Choose the field in the table to base the join on:

   zip_code

Finally, in order to see the data on your map, you need to specify different colors or symbols for different categories or quantities. We wanted to examine how the ZIP code clusters map out across the United States, so we made each cluster a unique color. To do this, right click again on the shape file name in the Layers window and click "Properties" at the bottom. Go to the Symbology tab. If you are mapping categorical data, such as we are, click "Categories" on the left and then choose the field that you want to display in the map from the Value Field box, for example, Zip_Cluster_2002. You will need to Add Values (Add All Values if you so wish) to retrieve the possible values for Zip_Cluster_2002. You can then choose a color scheme and format labels for the clusters in the same window. If you do not like the colors that ArcMap offers, you can change each color by clicking on the color and then selecting "Properties for Selected Symbols." You can also change the labels of your clusters to reflect more descriptive names for the marketers. We also suggest removing the symbol outlines if creating a zip code map, because when the outlines are present, they can obscure the coloring of small zip codes on your map.

Figure 7



Layer Properties

Draw categories using unique values of one field.

Value Field: zip_cluster_2002

Color Scheme

Symbol	Value	Label	Count
	<all other values>	<all other values>	0
	<Heading>	**zip_cluster_2002**	**30009**
	<Null>	<Null>	134
	1	Family Values	10617
	2	2	4457
	3	3	296
	4	4	127
	5	5	4114
	6	6	3541
	7	7	6435
	8	8	136

There are many options available for customization of your maps. For more tips on making maps that meet your needs, visit http://www.esri.com.

Figures 8 and 9 show examples of the maps that can be created using this software.

Figure 8

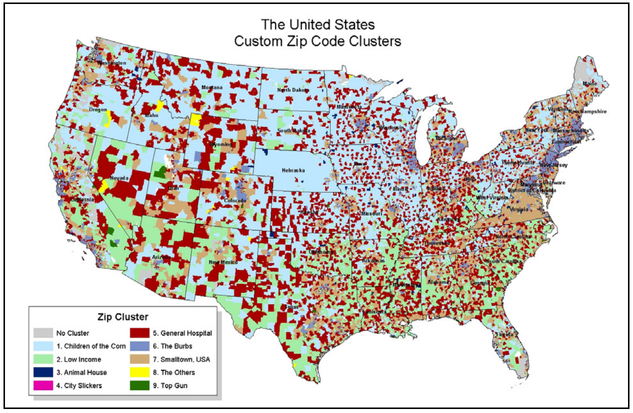
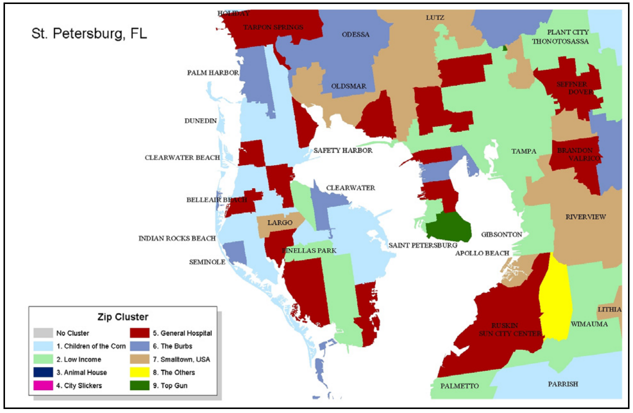


Figure 9



CONCLUSION

Creating a custom geographic segmentation system can be useful for direct marketing companies yet inexpensive to develop. The procedures explained in this paper describe only one method of clustering, and you should examine all possibilities for creating clusters before deciding on the best method for segmenting your data. There is no single "best" way of clustering your data set, so you should take time to explore different options before deciding on the best solution for your application.

You may also choose to purchase a segmentation system instead of creating one, which would save time and may be sufficient for your clientele. Understanding the inputs to these segmentation systems is key to being able to properly use them. Some vendors who create and sell segmentation systems are:

Claritas' Prizm and MicroVision Codes:
http://www.claritas.com/index.html

ESRI Business Information Systems' ACORN Codes:
http://www.esribis.com/

Experian's MOSAIC System:
http://www.experian.com/products/segmentation_systems.html

REFERENCES

Arnold, T. and Stokes, M. (1999*). SAS/STAT® User's Guide, Version 8.* Cary, NC: SAS Institute Inc.

Hair, J., Anderson, R., Tathem, R. and Black, W. (1998). *Multivariate Data Analysis: Fifth Edition.* New Jersey: Prentice Hall, Inc.

Hamer, Robert M. (1997). *Multivariate Statistical Methods: Practical Applications.* Cary, NC: SAS Institute Inc.

Johnson, D. (1998). *Applied Multivariate Methods for Data Analysis*. Pacific Grove: Duxbury Press.

Peppers, D. and Rogers, M. (1997). *The One to One Future: Building Relationships One Customer at a Time*. New York: Double Day.

Shepard, D. (1995). *The New Direct Marketing. How to Implement a Profit-Driven Database Marketing Strategy*. New York: McGraw-Hill.

ACKNOWLEDGMENTS

Thanks to Jen Warner, Mike Morgan, and Shawn Yoder for their assistance in topic development and contributions. Thanks to Dr. Craig Bach, Dr. Tom Short and Paula Marolewski for proofreading.

CONTACT INFORMATION

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

Allison Freeman or Kellie Poulin
Marsh Affinity Group Services
602 Office Center Drive, Suite 320
Ft. Washington, PA 19034

Phone: 215-653-8000
Fax: 215-653-5490

Email: Allison.N.Freeman@marshpm.com,
Kellie.M.Poulin@marshpm.com