Paper BtB-06

Use SAS® to create equal sized geographical clusters of people

Josh Young, Research Analyst, The Carolinas Center for Medical Excellence

ABSTRACT

How do you determine where staff should be located to serve your population when that population is spread out across a geographic region? With PROC GEOCODE and the SAS® supplied ZIP code files the tools are available to determine where staff should be located to fit your population. One method has been developed for figuring out how the staff should be distributed. This paper will discuss this method in addition to how to summarize and display the results using SAS® maps.

INTRODUCTION

The Carolinas Center for Medical Excellence works with organizations and individuals across North and South Carolina. Like many businesses, our company employs a combination of on-site and remote workers (field staff). Field staff need to service customers in multiple locations across the states, including remote areas from the coast to the mountains. This raises the question of where we need to have field staff located to minimize driving distances and maximize our coverage. Our approach to answer these questions using SAS geocoding and a custom macro, will be detailed in this paper.

METHOD

The source data used in this example is an address file representing the population of interest, as well as the ZIP code file from SAS help and the county file from SAS maps, updated via a download from the SAS support website. The address file contains street addresses, city, state, and ZIP code fields. I also used the street-level geocoding data available from the SAS support website. I used these files to divide the population into groups based on their location.

The first step is to geocode the addresses for the population of interest. This is done using PROC GEOCODE and the street-level lookup data available from SAS. Using the method=street option returns the x and y coordinates for the address if found and for the ZIP code centroid if the address can't be found. The file usm is the file downloaded from the SAS support website for street-level lookup. Depending on the address file you are using you may want to do some address clean up before running the geocoding. Below is sample code:

```
libname lookup 'C:\SAS Streetlookup Geocode\201210\Data' ;
proc geocode data=popb
             out=pop2
             addressvar=addr1
             addresscityvar=city
             addressstatevar=state
             addresszipvar=zip
             method=street
             lookupstreet=lookup.usm
             ;
run ;
```

To create the population groups for evaluation, I needed a set of starting points spread across the state so that I could create and evaluate groupings. I decided to use the ZIP code centroids provided by SAS as these points, since they are readily available. The ZIP code centroids are the geographical center points for each ZIP code. The first step was to use the geocoded locations for the population and create a cross file with the ZIP code centroids. To help evaluate the groups, I included the distance from each geocoded coordinate to each ZIP code centroid in the file. I used a formula for straight line distance that takes into account the curvature of the earth's surface. From this file I created groupings on every ZIP code, with each evaluation group containing the people closest to the ZIP code centroid.

```
proc sql ;
   create table cross as
```

```
   select a.zip,
          a.x as x1,
          a.y as y1,
          /* create a new variable for identifying the zip code matches vs the
street level ones. */
          case when b._matched_ = 'ZIP match' then 'group'
               else 'point'
               end as type,
          b.mid as mid2,
          b.zip as zip2,
          b.x as x2,
          b.y as y2,
          /* calculate distance in miles for each person compared to each zip
code. */
          3949.99*arcos(sin(atan(1)/45 * y1)*sin(atan(1)/45 * Y2)
               + cos(atan(1)/45 * y1)*cos(atan(1)/45 * Y2)*cos(atan(1)/45
               * X1-atan(1)/45
             * X2)) as dist
   from nczip as a, pop2 as b
   ;
   quit ;
```

In this case, my goal was to create 60 final groups of the same size. To do this, I calculated how many people were needed in each group (&target), and then ran a data step to create evaluation groups based on each of the ZIP code centroids including the closest people. For example, if each group needed 700 people and I had 1000 zip codes, I would create 1000 evaluation groups of 700 people. Each of these 1000 evaluation groups would contain the 700 people that lived the closest to the centroid for the zip code.

```
/* create a group around every ZIP code in the state. */
proc sort data=cross out=cross_g ;
   by zip dist mid2 ;
run ;

data cross2 (keep=zip type g:) ;
   set cross_g ;
   by zip dist mid2 ;
   if _n_ = 1 then group = 0 ;
   if first.zip then do ;
      group + 1 ;
      cnt = 0 ; /* restart the counter for number of people in each group. */
   end ;
   if cnt <= &target then do ;
      /* keep the mid, x and y coordinates, and the distance from the ZIP code
the group is formed on. Rename them all. */
      g_mid = mid2 ;
      g_x = x2 ;
      g_y = y2 ;
      g_d = dist ;
      cnt + 1 ;
      output ;
   end ;
run ;
```

The final groups needed to have every person included in only one group. Also, the goal was to have the total distance among all 60 of the final groups as low as possible. To do this, I first removed all ZIP code centroids that had no recipients within 10 miles of them. Doing this eliminated most of the crescent shaped regions I was getting. I also sorted the file so that the ZIP code grouping that had to go the furthest to reach the final person for the group was the first group in the file. In the evaluation file I kept the ZIP code that it was centered on, an identifier for each person, a number for the group, and the distance for the person farthest from the ZIP code centroid in the group.

```
proc sql ;
   create table group_anal as
   select a.zip,
          a.g_mid,
          a.group,
```

```
          c.max_dist /* Distance for the person farthest from the ZIP code in
the group. */
   from cross2 as a left join (select group,
                                      min(g_d) as min_dist,
                                      max(g_d) as max_dist
                               from cross2
                               group by group
                               ) as c
         on a.group=c.group
         where c.min_dist < 10
       /* Only keeping groups where at least one person is within 10 miles of
       the ZIP code. This helps eliminate crescent shaped groups that could
       surround another group. */
   order by max_dist desc
/* I want to select the group that has the person farthest from the ZIP code.
This selects the people that are hardest to place first, and makes the groups
smaller in total distance as we move forward. */
   ;
   quit ;
```

Assignment to the final groups was prioritized based on the farthest distance to the last person in the evaluation group. This allowed me to assign the people who were more remote first and leave the more densely packed areas for the end. I used a macro to loop through and remove people that had been assigned to a final group and removed ZIP codes I had used as starting points (I didn't want to use the same starting point twice and end up with a donut shaped region). The macro then created a new batch of evaluation groups to consider from the remaining people and ZIP code centroids. The results of each of the selected groups were written to a file so that I would have the assignments for every person in my original population.

```
%macro crt_grp ;
   * 60 groups mean 60 loops. ;
   %do i = 1 %to 60 ;
      /* selecting the group I want to keep. This will be one of the final
       groups. */
      proc sql ;
         create table g_w&i as
         select d.*
         from group_anal (obs=1) as a left join cross2 as d
            on a.group=d.group
         ;
         quit ;

      /* Create a file with all the people placed into a final grouping */
      %if &i = 1 %then %do ;
         data final ;
            set g_w&i (drop=group) ;
            gr = &i ;
         run ;
      %end ;
      %else %do ;
         data final ;
            set final g_w&i (drop=group) ;
            if gr = . then gr = &i ;
         run ;
      %end ;

/* Remove all the people who have been placed in final groups from the cross
file.  Also remove the ZIP codes I have selected as center points as I only
want each one used once. */
      proc sql ;
         create table cross_g as
         select a.*
         from cross as a left join final as b
            on a.mid2=b.g_mid left join final as c
               on a.zip=c.zip
         where b.gr is null
            and c.gr is null
```

```
         ;
         quit ;

/* create a new list for each remaining ZIP code with the people who are
closest to the ZIP code. */
      proc sort data=cross_g ; by zip dist mid2 ; run ;

      data cross2 (keep= zip type g:) ;
         set cross_g ;
         by zip dist mid2 ;
         if _n_ = 1 then group = 0 ;
         if first.zip then do ;
            group + 1 ;
            cnt = 1 ;
         end ;
         if cnt <= &target then do ;
            g_mid = mid2 ;
            g_x = x2 ;
            g_y = y2 ;
            g_d = dist ;
            cnt + 1 ;
            output ;
         end ;
      run ;

      /* Find the new evaluation group with the person farthest from a ZIP
       code. */
      proc sql ;
         create table group_anal as
         select a.zip,
                a.g_mid,
              a.group,
                c.max_dist
         from cross2 as a left join (select group,
                                           min(g_d) as min_dist,
                                           max(g_d) as max_dist
                                    from cross2
                                    group by group
                                    ) as c
               on a.group=c.group
         where c.min_dist < 10
         order by max_dist desc
         ;
         quit ;

   %end ;
%mend ;

%crt_grp ;
```

Once all of the final groups had been created, I wanted to find a center point for each group. I did this by averaging the x and y coordinates, and finding the ZIP code centroid that was closest to that center point. This was done by creating a cross file between the averaged x and y coordinates and the ZIP code centroid file and calculating the distances between the two of them. I used this to create a list of “ideal” ZIP codes for the field staff.

```
proc sql ;
   create table gr_loc as
   select a.*,
          b.x as x2,
          b.y as y2,
          b.zip
   from (select gr,
                mean(g_x) as x,
                mean(g_y) as y
         from final
         group by gr
```

```
        ) as a, nczip as b
    ;
    quit ;

/* Calculate the distance from the final group center points to the ZIP code
centroids */
data gr_dist ;
    set gr_loc ;
    stLat = atan(1)/45 * y ;
    stLon = atan(1)/45 * X ;
    endLat = atan(1)/45 * Y2 ;
    endLon = atan(1)/45 * X2 ;
    arc = sin(stLat)*sin(endLat) + cos(stLat)*cos(endLat)*cos(stLon-endLon) ;
    distance = 3949.99*arcos(arc) ; *Distance in miles ;
    format distance 8.1 ;
run ;

/* Keep only the closest ZIP code centroid for each final group center */
proc sort data=gr_dist ; by gr distance ; run ;

data gr_final (keep=gr x y zip) ;
    set gr_dist ;
    by gr distance ;
    if first.gr then output ;
run ;
```

RESULTS

I put the final results into a series of tables and maps. Here are some of examples of the output I created:

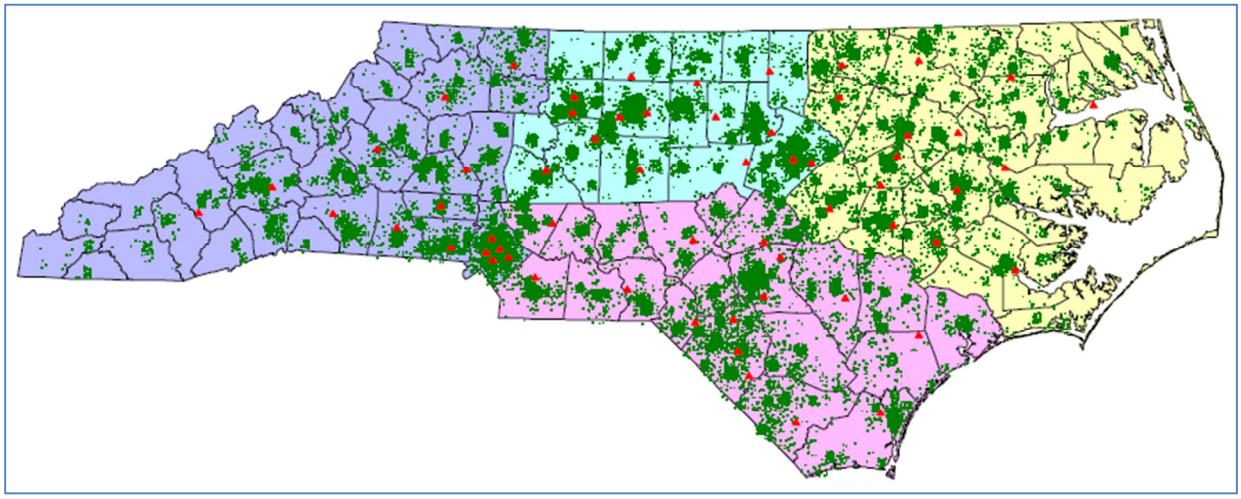

Figure 1. Display of ideal field staff (red) location and the location of the original population (green)

West	28805	Buncombe	5
West	28641	Burke	9
West	28609	Catawba	31
West	28150	Cleveland	13
West	28054	Gaston	35
West	28738	Haywood	1
West	28092	Lincoln	32
West	28297	Mecklenburg	54
West	28220	Mecklenburg	58
West	28218	Mecklenburg	59
West	28212	Mecklenburg	60
West	28208	Mecklenburg	57
West	28167	Rutherford	6
West	27017	Surry	7
West	28697	Wilkes	8

Table 1. Ideal staff for Western Region of the state

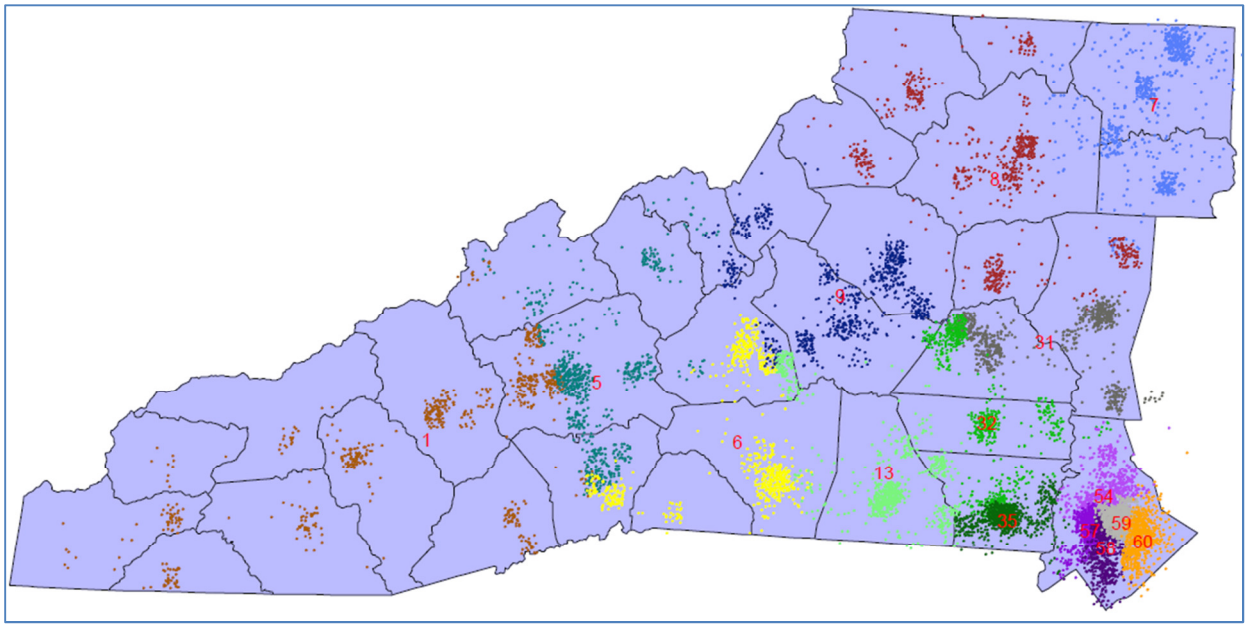

Figure 2. West Region Map of Groupings

Table 1 is the list for the western part of the state. It shows an “ideal” ZIP code location for each field staff, and the county that the ZIP code is associated with. The groups with background color correspond to the map in Figure 2**.** Figure 2 is a blowup of the western part of North Carolina with the population colored for each group and the ideal staff locations numbered. The numbers on the map correspond to the order they were assigned across the state. The colors and numbers match up with Table 1.

The code for generating the maps and tables shared here is available at:

https://drive.google.com/folderview?id=0ByEmUxWaHSyHNE5wTll4eGU3eXc&usp=sharing

CONCLUSION

This paper shows one way to create groups using geographical coordinates while forcing the groups to be the same size. SAS provides many different tools to tackle data problems. With a bit of creativity and some graphical procedures you can create groups based on location. Hopefully some of the ideas presented here can help you develop your own geographical programs.

CONTACT INFORMATION

Josh Young
The Carolinas Center for Medical Excellence
100 Regency Forest Drive, Suite 200
Cary, NC 27518
jyoung@thecarolinascenter.org