Mapping Participants to the Closest Medical Center

David Franklin, QuintilesIMS

ABSTRACT

"How far are patients from Clinics?" That was the question which was asked on a wintery afternoon. Recorded in the database was the ZIP codes of each participant, but how could we use that data to find the closest distance from the nearest clinic? "And would it not be nice if we could map it?"

This paper looks at calculating the distance from a participant ZIP code to a Medical Center using the GEODIST function, finding the smallest distance for each participant to attend, and finally color-coding the plots depending on the distance calculated using the GMAP procedure. Also helpful is producing reports, one of which will be the those participants who are farthest away and the distance.

INTRODUCTION

Mapping the location of data in your study can be fraught with complications of privacy and being able to identify participants. But sometimes it is necessary, particularly if you have clusters of participants who have to travel great distances where it may be necessary to set up clinics closer to them.

This paper looks at mapping locations and reporting useful data that is helpful to those whom have to make decisions on the conduct of a study.

OUR DATA

For the purposes of this paper, the following datastep has the ZIP Code of 210 participants for your study that will be used for the analysis:

```
data _pdat0;
 length zipcode $5;
 infile cards;
 input zipcode $ @@;
 cards;
39165 14602 92171 80420 22212 27833 73033 24894 25715 19019 75390 75326 53295 48333
15420 93232 63881 27619 06875 25360 68544 40621 07507 83729 05544 49320 65106 78768
23090 19898 98129 97388 29424 04744 94211 80448 54602 60290 07533 98259 22626 30359
95190 76369 45319 47853 46411 33163 62289 08868 94287 01613 86053 17607 38021 20429
39832 88102 90084 95194 38007 16016 94160 20081 59432 61826 71750 55582 39283 55713
31420 33733 48311 19733 49317 40602 53789 18449 95944 33871 16660 84016 71324 61204
44816 73022 92712 62540 15315 04123 53127 58126 74050 44309 75666 28699 25375 48046
06059 78658 76101 12257 44845 47420 15532 75682 38010 06140 19550 58707 92521 75688
29413 86023 83223 27429 29733 39569 85121 21609 28093 53537 33257 63022 25719 08212
25365 60681 44860 50981 51008 18602 32936 48061 40061 98062 97147 30623 32235 59717
56294 32886 15841 76439 74535 90510 14061 20075 14854 25911 91503 16546 31993 35249
32313 75372 73023 10156 96140 59936 18946 27593 94948 30237 90078 22161 58702 08526
30448 11773 50332 21041 20421 24463 36015 98614 35236 41549 70152 62071 19714 72812
43349 38187 77991 22160 85079 85320 73344 77882 05303 78381 01538 87302 36123 43007
46857 34133 38503 53792 55950 29333 95108 18540 27819 15265 15660 48410 75014 02652
;
run;
```

and the second has the zip codes of the clinical centers.

```
data _cdat0;
 length zipcode $5;
 infile cards;
 input zipcode $ @@;
cards;
10005 63102 90017
;
run;
```

It must be noted that this data is fictitious and does not reflect any data that may exist in actual study or similar databases.

CALCULATING THE DISTANCE

While there is a function called ZIPCITYDISTANCE that returns the geodetic distance in miles between two ZIP code locations, this could not be used in this work due to the function working on only the ZIPCODE dataset in the SASHELP directory, and unfortunately is not an up-to-date dataset due to new ZIP Codes being added on a continual basis, even faster than the updates to the ZIPCODE dataset allow. Another approach had to be adopted. The result was to use another dataset with ZIP Code, and longitude and latitude for each, plus some location information, and the GEODIST function that returns the geodetic distance in kilometers (default) or miles between two latitude and longitude coordinates, either as degrees (default) or radians. The syntax for the GEODIST function is:

GEODIST(*latitude-1, longitude-1, latitude-2, longitude-2 <,options>*)

Further details of the GEODIST function can be found online in relevant SAS documentation.

To find the longitude and latitude for each ZIP Code, there is the ZIPCODE dataset in the SASHELP library, as well as several other sources, free or otherwise, that provide this information to various levels of precision. With any of these, the three fields that are needed deal with:

- ZIP Code (usually to five digits, variable ZIPCODE)
- Latitude (numeric, usually degrees, variable Y)
- Longitude (numeric, usually degrees, variable X)
- State FIPS Code (numeric, needed for the GMAP procedure, variable FIPS)
- State Abbreviation (common two-character code, variable STATEAB)
- City (character, Name of City, variable, CITY)

Note that the variable names used here are in this dataset may be different from that of the ZIPCODE dataset.

After the dataset with ZIP Code and longitude and latitude is available, it is necessary to apply the latitude and longitude coordinates to each record in the two datasets, with Y1 and X1 being the latitude and longitude for those of the PARTICIPANT dataset, and with Y2 and X2 being the latitude and longitude for those of dataset MEDICAL_CENTER -- the reason for the pairs Y1/X1 and Y2/X2 is that the two points are needed for the calculation of distance. The following example is the first ten records from _PDAT with the longitude (X1) and latitude (Y1):

```
   zipcode          x1           y1    fips    city_state

   06120        -72.67        41.78      9     Hartford, CT
   54824        -92.55        45.45     55     Centuria, WI
   61747        -89.42        40.42     17     Hopedale, IL
   80818       -103.50        39.28      8     Genoa, CO
   50165        -93.27        40.86     19     Millerton, IA
   50320        -93.58        41.54     19     Des Moines, IA
   49756        -84.27        44.81     26     Lewiston, MI
   52079        -90.72        42.31     19     Zwingle, IA
   57065        -96.61        43.89     46     Trent, SD
   93240       -118.44        35.63      6     Lake Isabella, CA
```

and the medical center dataset having the following data:

```
   zipcode          x2           y2    fips     city_state

   10005        -74.01        40.71     36     New York, NY
   63102        -90.19        38.64     29     Saint Louis, MO
   90017       -118.27        34.05      6     Los Angeles, CA
```

The next phase of the calculation of distance was to find the closest distance of each PARTICIPANT record to the MEDICAL_CENTER record -- this was done by doing a one to many merge and then finding the the closest distance with the matching ZIP code, as shown below:

```
%macro tmp;
   *Get the location of centers;
   data _null_;
      if 0 then set _cdat1 nobs=nobs;
      call symput('num',trim(left(put(nobs,8.))));
      Stop;
   run;

   *Process for each center;
   %do i=1 %to &num;

      *Get center zip, longitude/latitude and merge, calculating distance (miles);
      data _dat1;
         set _cdat1;
         if _n_=&i;
         rename zipcode=cntrzip citystate=cntrcity;
      data _pdat1;
         if _n_=1 then set _dat1;
         set _pdat1;
         length _dist&i _zip&i _xx&i _yy&i 8 cntrcity $100;
         _dist&i=geodist(y1,x1,y2,x2,'M');
         _zip&i=cntrzip;
         _city&i=cntrcity;
         _yy&i=y2;
         _xx&i=x2;
         drop y2 x2 cntrzip cntrcity;
      run;
   %end;
```

```
    *Find closest distance to each of the centers and corresponding zip code;
    data _pdat2;
       set _pdat1;
       array zaz{%eval(&num.+0)} _dist: ;
       array zbz{%eval(&num.+0)} _zip: ;
       array zcz{%eval(&num.+0)} _city: ;
       array zdz{%eval(&num.+0)} _xx: ;
       array zez{%eval(&num.+0)} _yy: ;
       do i=1 to &num;
          if _cdist>zaz{i} or _cdist=. then do;
             _cdist=zaz{i};
             _czip=zbz{i};
             _ccity=zcz{i};
             _xx=zdz{i};
             _yy=zez{i};
          end;
       end;
    run;

  %mend tmp;
  %tmp;
  run;
```

Once the dataset has the closest zipcode and distance from the same, the next stage is to plot the graph.

The first part was to use bring together the PARTICIPANT data, with distance information, and MEDICAL_CENTER data, for longitude and latitude information and noting the source.

```
data longlat;
  set _pdat2 (in=a)
      _cdat1 (in=b rename=(x2=x1 y2=y1));
  x=atan(1)/45*x1;
  y=atan(1)/45*y1;
  x=-x;
  if a then _src=1;
  else if b then _src=2;
  keep zipcode x y _src _cdist;
run;
```

The next phase is to create the annotated dataset with the longitude and latitude information, but setting the size and the color of the 'dot' depending on the distance:

```
data anno;
  length color $10;
  set longlat;
  retain xsys ysys '2' /*function 'label' size .75 */ flag 1 when 'a';
  function='symbol';
  text='dot';
  if _src=1 then do;
    if closedist>=750 then color='RED';
    else if closedist>=250 then color='ORANGE';
    else if closedist>. then color='BROWN';
    size=1.25;
  end;
  else if _src=2 then do;
    color='green';
    size=2.50;
  end;
  output;
run;
```

The last stage is to produce the graph

```
*Set graphic pattern;
pattern1 v=me c=grp r=50;

*Titles and Footnotes;
title1 'Distance from Participant to Nearest Medical Center';
footnote1 color=black font=swiss "Legend: "
         color=green font=marker "W " font=swiss "Medical Center "
         color=brown font=marker "W " font=swiss "0-249 miles "
         color=orange font=marker "W " font=swiss "250-749 miles "
         color=red font=marker "W " font=math "M" font=swiss "750 miles";

*Generate map;
proc gmap data=maps.states map=maps.states(where=(state not in(2 15 72)));
  id state;
  choro state / anno=anno nolegend;
  quit;
run;
```
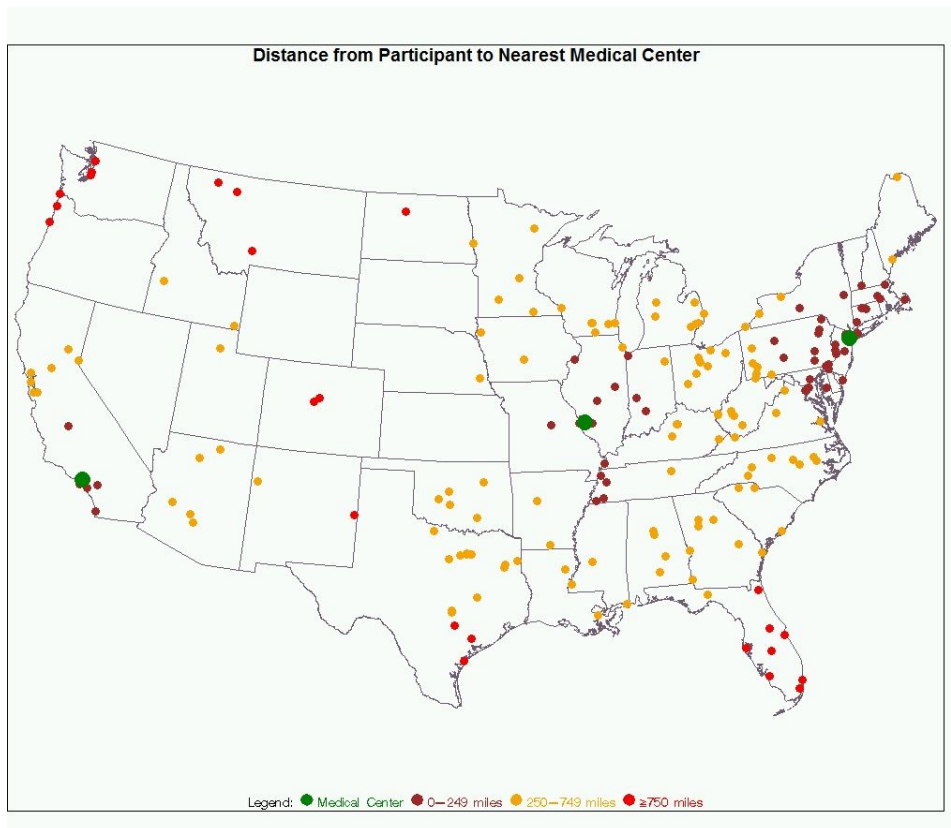
and the result is in Figure 1.

Figure 1: Distance from Participant to Nearest Medical Center

While the map is certainly interesting, there are produce reports that are of interest. In the case of the data above, it may be of interest to actually list those patients who are 750 miles or more from the closest medical center.

```
proc sort data=_pdat2 out=_pat3;
   by _ccity descending _cdist;
   where _cdist>750;
title1 'Distance from Participant to Nearest Medical Center';
title2 'Where Distance >750 Miles';
proc print data=_pdat3 noobs label;
   by _ccity;
   id _ccity;
   var city_state zipcode _cdist;
   format _cdist 8.;
   label _ccity='Medical Center'
         city_state='City'
         zipcode='ZIP Code'
         _cdist='Distance from Medical Center (miles)';
run;
```

which produces the output in Output 1.

```
         Distance from Participant to Nearest Medical Center
                    Where Distance >750 Miles


                                               Distance
                                             from Medical
                                        ZIP     Center
Medical Center     City                 Code    (miles)

Los Angeles, CA    West Glacier, MT     59936      1021
                   Dupuyer, MT          59432      1019
                   North Lakewood, WA   98259       996
                   Seattle, WA          98129       960
                   Seahurst, WA         98062       950
                   Chinook, WA          98614       895
                   Bozeman, MT          59717       887
                   Wheeler, OR          97147       856
                   Gleneden Beach, OR   97388       808
                   Grant, CO            80448       788
                   Alma, CO             80420       767


Saint Louis, MO    Miami, FL            33257      1065
                   Miami, FL            33163      1051
                   Bonita Springs, FL   34133       978
                   Sebring, FL          33871       919
                   Melbourne, FL        32936       909
                   Saint Petersburg, FL 33733       865
                   Minot, ND            58702       865
                   Minot, ND            58707       865
                   Orlando, FL          32886       861
                   Rockport, TX         78381       833
                   Clovis, NM           88102       781
                   Vanderbilt, TX       77991       761
                   Jacksonville, FL     32235       754
                   Ottine, TX           78658       753
```

Output 1: Distance from Participant to Nearest Medical Center, Where Distance >750 Miles

SO LET'S ADD SOME LINES

While looking at this challenge, an additional challenge was added to draw lines from the closest Medical Center to the closest participant so it is possible to easily see the link between both.

From what was already present, their had to be one more step added in the creation of the LONGLAT dataset which transformed the coordinates of the medical center, as shown in red below:

```
data longlat;
  set _pdat2 (in=a)
      _cdat1 (in=b rename=(x2=x1 y2=y1));
  x=atan(1)/45*x1;
  y=atan(1)/45*y1;
  x=-x;
  if a then do;
     xx=atan(1)/45*_xx;
     yy=atan(1)/45*_yy;
     xx=-xx;
     _src=1;
  end;
  else if b then _src=2;
  keep zipcode x y xx yy _src _cdist _czip;
run;
```

and then creating a second annotate dataset with line information:

```
data anno2;
  length color $10;
  set longlat;
  where _src=1;
  retain xsys ysys '2' /*function 'label' size .75 */ flag 1 when 'a';
  if _czip=10005 then color='pink';
  else if _czip=63102 then color='lime';
  else if _czip=90017 then color='aqua';
  function='move'; output;
  function='draw'; x=xx; y=yy; output;
run;
```

then combining the two annotate datasets to make certain the 'dots' overlay any lines

```
data anno;
   length function $8;
   set anno2 anno;
   flag=1;
run;
```

After the code is rerun with these changes, the graphic in Figure 2 was generated:
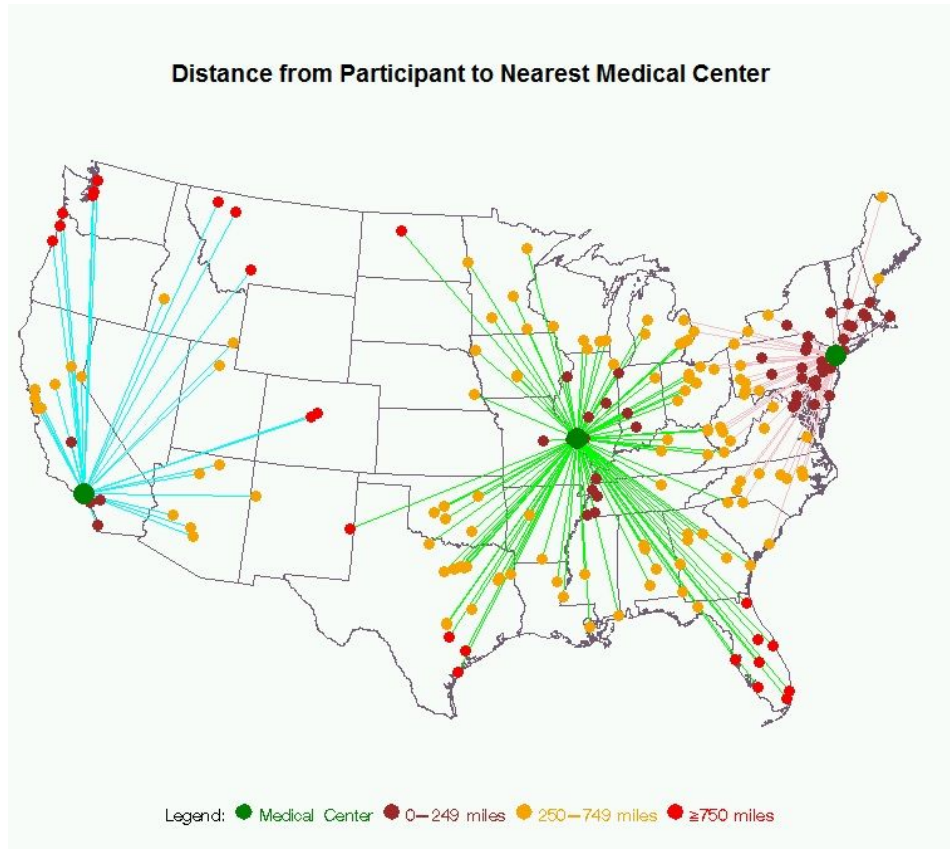
Figure 2: Distance from Participant to Nearest Medical Center, with lines

CONCLUSION

Finding the distance of participants from the nearest center, and mapping it, is possible through converting ZIP Codes to Longitude and Latitude, and then using the GEODIST function for calculation of the distance. Once this is done, with a little bit of annotation it is possible to use PROC GMAP to plot the location of participants and centers, and do analysis of distance.

CONTACT INFORMATION

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

David Franklin
QuintilesIMS
david.franklin1@quintilesims.com
quintilesims.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.