

LINKING SAS® ANALYTICS, U.S. CENSUS GAZETTEER DATA AND ARCVIEW GEOCODING SOFTWARE FOR MEDICARE FRAUD SCREENING

Laurie Hamilton, U.S. Government Accountability Office, Washington D.C.¹

ABSTRACT

This paper describes a proof-of-concept study undertaken to determine the usefulness of provider-beneficiary mean geographic distances as a screening tool in Medicare fraud, waste and abuse detection activities. Techniques included geocoding provider and beneficiary addresses, calculating geographic distances between provider-beneficiary pairs, and creating summary statistics and automated screening procedures. We used ESRI's ArcView Desktop GIS software for street-level geocoding, Gazetteer files publicly available from the U.S. Census Bureau for zip code-level geocoding, and Base SAS for all data transfer activities, distance calculations, summary statistics and screening. SAS procedures included PROC DBF, PROC MEANS and PROC SUMMARY. The DATA Step was also used for calculations and screening. Outcomes suggest that larger-than-average mean provider-beneficiary geographic distances are a good indicator of potentially anomalous provider activity and that the methodology developed is for the most part time-efficient and cost-effective.

INTRODUCTION

As part of a data-driven Medicare fraud, waste and abuse detection program, we were asked to explore the potential of adding a Geographic Information System (GIS) software package to the existing analytical and data mining tools, which included Base SAS. The GIS system chosen was ESRI's ArcView Desktop Version 9.1. As we came to understand the inner-workings of a GIS, we realized that one underlying product – the geocoding tool – might prove useful in a unique approach to prescreening large volumes of claims data for anomalies.

ArcView's geocoding tool is an address locator called StreetMap USA. From an input file of U.S. addresses, StreetMap is able to return the exact geographic coordinates (latitude and longitude) of all addresses known at the time of the 2000 Census. For us, this translated to the potential to determine the geographic distances between Medicare providers and the beneficiaries they serve, because it is possible to calculate the distance between two points once the latitude and longitude are known. We were working from the hypothesis that unusually large distances between service providers and their beneficiaries frequently are anomalous and potentially indicative of abusive, wasteful or legally fraudulent activity.

Our hypotheses linking large provider-beneficiary geographic distances and anomalous provider activity was based on known patterns of fraudulent Medicare provider activity. Among those practices are claiming for services not actually rendered and using a purchased list of beneficiaries.

We developed a proof-of-concept methodology to test our provider-beneficiary distance hypotheses. At the outset, it included the use of the StreetMap USA geocoding tool and Base SAS analytics. We later expanded the geocoding efforts to include Gazetteer data publicly available from the U.S. Census Bureau. From this mix of U.S. Census Gazetteer data, StreetMap geocoding capabilities, and Base SAS analytics, we attempted to answer two questions:

- Could we demonstrate that large geographic distances between Medicare providers and the beneficiaries they serve mirrored the results of more traditional data mining pattern detection results?
- Was the production sequence needed to accumulate and transform the data necessary for geocoding and distance calculations time-efficient and cost-effective?

¹ This paper describes work done when Laurie Hamilton was an employee of SRA International, Inc., Fairfax, Virginia.

DISTANCE CALCULATION AND GEOCODING BASICS

In order to calculate the distance between two points on the earth's surface, it is necessary to find the geographic coordinates for the two locations. The process of identifying the latitude and longitude coordinates for a specific location is called geocoding. Once the coordinates are known, various algorithms can be used to calculate the physical distance between the points. For any address, the most accurate level of geocoding is street-level – the exact latitude and longitude for the physical position of the address. If the data are not accurate enough for street-level geocoding, it is also possible to identify the position less precisely based on zip code or a new U.S. Census geographic entity called the ZCTA.

We used ArcView's StreetMap USA software to geocode addresses to street-level. The information underlying the StreetMap geocoding process is vintage 2000 Census. This means that the StreetMap software cannot geocode addresses which did not exist at the time of the 2000 Census. Since our proof-of-concept used data from 2002 to 2005, this was an initial source of missing coordinate data in our distance calculation datasets. In addition, all geocoding software is set up to recognize addresses in certain formats and will return missing values for badly formatted data, invalid addresses, incomplete addresses and PO Boxes. In order to impute coordinates for the addresses which could not be geocoded by StreetMap, we turned to several Gazetteer files publicly available from the U.S. Census Bureau. These included a 1999-vintage file of Zip Codes with coordinates and a 2000-vintage file linking the newly devised ZCTAs with coordinates. In both cases the coordinates are assigned based on a best estimate of the location of the geographic center, or centroid, of the area covered. The differences between the two Census files, and the relative usefulness of each for our proof-of-concept, will be discussed later in this paper.

The four-step process we developed to assign coordinates to provider and beneficiary addresses was:

- 1) Send all addresses to StreetMap to attempt street-level geocoding,
- 2) Geocode all addresses to zip code-level using the 1999 Census Zip Code Gazetteer File,
- 3) Geocode all addresses to ZCTA-level using the 2000 Census ZCTA Gazetteer File, and finally
- 4) Select the most precise coordinates available for each address as the input for the distance calculations

Algorithms to calculate distances along the surface of the earth are highly complex mathematically. There are many versions, all based on trigonometric functions which are beyond the scope of this paper. We chose to use the Great Circle Formula and adapted our methodology from a sample provided by SAS in their online Technical Documentation Samples section.

Street-Level Geocoding Techniques

ArcView accepts and processes data in both Dbase V (DB5) and Microsoft Access formats. We chose to transfer data between SAS and ArcView in the DB5 format using the DBF Procedure. The geocoding software parses the input addresses and attempts to match them to proprietary lists containing latitude and longitude coordinates. The output from StreetMap contains latitude and longitude, when available, and additional information identifying the level of certainty of the result.

Preparing the Transfer File

The first step in street-level geocoding is to prepare the file for transfer to the geocoding software. All geocoding software needs a minimum of four pieces of information: a street address, city, state and zip code. StreetMap will accept a file with ancillary information; it keys only on the four primary location fields for the geocoding activity.

The rules for creating the DBF file for StreetMap are straightforward. The key location fields must be identified by name: ADDRESS, CITY, STATE and ZIP. StreetMap will attempt to recognize names similar to these; however, SAS truncates all variable names of more than 10 characters when creating a DB5 file so it is important to name the fields carefully.

Below is the code we used to export our SAS data to a DBF file. In addition to the location fields, we also included a unique provider or beneficiary identifier and summary dollar values. The code illustrates an important point

regarding the use of PROC DBF to create a Dbase file. Numeric fields containing decimal values must be formatted before the procedure, otherwise the decimal portion of the numbers will be lost.

```

/* prepare the data for the DBF procedure. Attach a format to numeric values
   with decimal places. Rename variables to conform to ArcView requirements */

data fordbf (rename=(practloc_address1=address
                    practloc_city      =city
                    practloc_state     =state
                    practloc_zip      =zip      ));

   set prov_data (keep = prov_id
                  dollars
                  practloc_address1
                  practloc_city
                  practloc_state
                  practloc_zip      );

format dollars 12.2;
run;

filename dbout 'c:\provs_to_geocode.dbf';
proc dbf db5 = dbout
   data = fordbf;
run;

```

Note that in the code above we selected a field called `practloc_address1` to use in geocoding. Depending on the structure of the underlying data warehouse, Medicare transfer files may contain both a business address and a billing address for a given provider. It is important to select the business or practice location address for the purposes of calculating provider-beneficiary distances since other addresses may not reflect the location at which services were provided. Both provider and beneficiary address files also frequently contain more than one street-address field. It is necessary to pre-screen the fields to determine which has the most usable information for geocoding. In most cases, actual street addresses are in the first address field, the second contains information such as apartment numbers or snippets of non-address identifying information.

It may also be necessary to consider the number of records in the file being exported to ArcView for geocoding. Although ESRI documentation states that the limitation is several million records, and DB5 supports tables of this size, we encountered StreetMap errors when attempting to geocode a DBF file containing over 60,000 rows. We circumvented the problem by creating and separately geocoding several smaller DBF files, and recombining the results when importing back to SAS.

Geocoding Output

For the purposes of this paper, we will not be discussing the actual street-level geocoding process performed by StreetMap. We are concerned only with understanding the geocoding output file and transferring it back into SAS to be used in geographic distance calculations.

In preparation for geocoding, StreetMap parses the input address and attempts to match it to a proprietary underlying file containing latitude and longitude. The address locator provided with the basic ESRI ArcView license is based on the 2000 Census; street addresses created after that time therefore will not be successfully geocoded. In addition, PO Box-based addresses cannot be geocoded by the StreetMap address locator.

The StreetMap geocoding process results in five output files, with the extensions `.dbf`, `.prj`, `.shp`, `.shp.xml`, and `.shx`. The `.dbf` file is a mirror image of the input records with additional fields containing the geocoding results; the remaining files are used for mapping. The first step in interpreting the street-level geocoding results is to import the `.dbf` file into SAS. This is accomplished by reversing the structure of the PROC DBF statements, specifying that the input is a DBF file and the output is a permanent or temporary SAS dataset. In this case, we imported into a temporary SAS dataset and ran an additional dataset to add descriptive labels to the data elements before creating a permanent file; only the import code is shown below.

```

/* import db5 file */
filename dbfin 'c:\provs_geocoded.dbf';
proc dbf db5 = dbfin
   out = provs_imported
;
run;

```

The geocoded dataset contains 19 variables; six from the original input dataset and an additional 13 created by the StreetMap geocoding process. The variables created by StreetMap are:

Variable	Type	Len	Format	Label
ARC_City	Char	11	\$11	ArcView City (same as input City)
ARC_State	Char	2	\$2	ArcView State (same as input State)
ARC_Street	Char	25	\$25	ArcView Street Address (same as input Address)
ARC_Zip	Char	5	\$5	ArcView Zip Code (same as input Zip)
Pct_along	Num	8	16.4	ArcView Percent Along
Ref_ID	Num	8	9	ArcView Reference ID
Score	Num	8	16	ArcView Geocoding Success Score
Side	Char	1	\$1	ArcView Side of Street for Address
Stan_addr	Char	52	\$52	ArcView Parsed Street Address
Status	Char	1	\$1	ArcView Address Match Flag (M/U)
X	Num	8	16.10	ArcView Longitude, Degrees
Y	Num	8	16.10	ArcView Latitude, Degrees

Interpreting Street-Level Geocoding Results

The StreetMap variables of interest to us were X, Y and SCORE. X and Y are the street address coordinates in degrees longitude and degrees latitude, respectively. SCORE is a number from 0 to 100 which indicates how ArcView rates the accuracy of the geocoding result. Addresses which cannot be geocoded will have SCORE, X and Y set to 0. Addresses which are successfully geocoded will have SCORE greater than 0, with 100 being complete certainty based on the underlying address locator file. STATUS can be used to quickly summarize overall geocoding success; it will equal "M" for a matched address and "U" for an unmatched address.

Zip Code-Level Geocoding Techniques

Not all address data can be successfully geocoded to street-level accuracy. Addresses may be missing, invalid, or formatted in such a way that they are not recognized by the address locator. They may be too new to be included in the locator, too general (PO Box and c/o, for example), or simply incomplete. If zip codes are available, these addresses can often be geocoded based on the assigned zip code using publicly available data from the U.S. Census Bureau. The latitudes and longitudes assigned to them will be less precise than their exact street-level coordinates, but can serve as a reasonable imputation for the purposes of average distance calculations. This imputation technique works in most cases because the areas covered by zip codes for all but the most rural areas are relatively small. The coordinates of the center of the area covered by the zip code, also called the zip code centroid, are therefore reasonably close to the actual address being geocoded.

U.S. Census Bureau Gazetteer Files- Zip Code 1999

The Census data we originally used for zip code-level geocoding contains the latitude and longitude coordinates for all zip codes active as of November, 1999. This file is available in DBase format on the Census website. Below are three sample records from the file; for our purposes we kept only ZIP_CODE, LATITUDE and LONGITUDE:

ZIP_CODE	LATITUDE	LONGITUDE	CLASS	PONAME	STATE	COUNTY
	(Degrees)	(Degrees)	P=PO Box	Post Office Name	FIPS State Code	FIPS County Code
32007	+29.799631	-81.627324	P	BOSTWICK	12	107
32008	+30.101927	-82.908004		BRANFORD	12	121
32009	+30.521109	-81.906051		BRYCEVILLE	12	089

Our technique for zip code-level geocoding using this file was a two-step process. First we imported the Census Gazetteer file into SAS using PROC DBF. Then we used a DATA Step to merge the imported Gazetteer file containing zip code, latitude and longitude with the provider and beneficiary address files which had already been geocoded using StreetMap. The files were merged on the zip code fields and only the records in the address files were output.

The Census Bureau states that it will not be updating the Zip Code Gazetteer file beyond the 1999 version currently available. Instead, they have switched to Gazetteer files based on a new geographic entity called the ZCTA (Zip Code Tabulation Area). Detailed information about Census's philosophy behind the development of the ZCTA, and the relationship of ZCTAs to zip codes, is available on the Census website. What was important for us was to test how closely the distances calculated using zip code coordinates approximated those calculated using ZCTA coordinates. We could attempt this because ZCTAs are frequently equivalent to zip codes, only the centroid latitude and longitude coordinates differ because the areas covered differ somewhat in size.

U.S. Census Bureau Gazetteer Files - ZCTA 2000

The Census files containing ZCTA coordinates are also available on the Census website. We selected the file containing coordinates based on the 2000 Census as most closely approximating the zip code coordinates from November, 1999. This file is in ASCII text format. It contains 10 fields which can be read into SAS using column delimited format based on the file layout provided by Census. Three sample records from the file are below; for our purposes we kept the fields for ZCTA, LATITUDE AND LONGITUDE:

State Code + ZCTA	Area Name	Population (2000)	Housing Units (2000)	Land Area (sq mi)	Water Area (sq mi)	Land Area (sq meters)	Water Area (sq meters)	Latitude	Longitude
AL35004	5-Digit	ZCTA	6998	2815	49387881	259146	19.06877	0.100057	33.606379
AL35005	5-Digit	ZCTA	8985	3690	92158183	14126	35.58248	0.005454	33.592585
AL35006	5-Digit	ZCTA	3109	1488	339241043	1012342	130.9817	0.390867	33.451714

Our technique for zip code-level geocoding using the ZCTA Gazetteer was similar to that used for the zip code file. We assumed that the ZCTA field closely approximated the zip codes in our address files and merged the Gazetteer and provider and beneficiary files by renaming the ZCTA field to ZIP.

OVERALL GEOCODING SUCCESS

Overall, we were able to geocode close to 99% of both the provider and beneficiary addresses. StreetMap identified 72% of the provider street addresses and 67% of the beneficiary street addresses. ESRI confirms that street-level geocoding success rates near 65% are the norm, primarily because the underlying StreetMap data are vintage Census 2000 and many recent address files contain some addresses which did not exist at the time of the 2000 Census. In addition, we made no effort to standardize or clean the addresses we obtained from the Medicare data warehouse.

An additional 27% of providers and 33% of beneficiaries were geocoded using zip code-level coordinates. All zip codes which had a match in the November 1999 Zip Code Gazetteer file also had a match in the 2000 ZCTA Gazetteer file.

Our proof-of-concept file contained nearly half a million provider-beneficiary pairs. The highest level of specificity, street-level coordinates for both members, was known for 53% of the pairs. An additional 38% were matched with street-level coordinates for one member and zip code-level coordinates for the other member. Only 9% of the pairs fell into the least specific geocoding group – both members with zip code level coordinates.

Provider Success	Beneficiary Success	Percent of Pairs
Street Address	Street Address	52.7
	Zip/ZCTA Centroid	23.7
Zip/ZCTA Centroid	Street Address	14.3
	Zip/ZCTA Centroid	9.0
Other		0.3

CALCULATING DISTANCES

Creating a Master Provider-Beneficiary File With Geographic Coordinates

Once provider and beneficiary addresses were geocoded, we created a master file of all provide-beneficiary pairs actually occurring in the claims data and attached the geocoding information to each pairs' record. This resulted in a file with provider and beneficiary location coordinates combined on one record. Below is the structure of our final master provider-beneficiary file:

Variable Name	Label
Prov_iD	Provider ID
Prov_x	Provider-Street Level-Longitude (Degrees)
Prov_y	Provider-Street Level-Latitude (Degrees)
Prov_cenzip_x	Provider-Zip Code or ZCTA Code Level-Longitude (Degrees)
Prov_cenzip_y	Provider-Zip Code or ZCTA Code Level-Latitude (Degrees)
Bene_iD	Beneficiary ID
Bene_x	Beneficiary-Street Level-Longitude (Degrees)
Bene_y	Beneficiary-Street Level-Latitude (Degrees)
Bene_cenzip_x	Beneficiary-Zip Code or ZCTA Level-Longitude (Degrees)
Bene_cenzip_y	Beneficiary-Zip Code or ZCTA Level-Latitude (Degrees)

Distance Calculations Using the Great Circle Formula

We were now ready to calculate distances. We used the best geographic information available for each member of a pair. We did not calculate distances for pairs in which at least one member could not be geocoded to the street- or zip code-level. We ran the distance calculations twice, once using the zip code-based coordinates and again using the ZCTA-based coordinates.

The algorithm we chose for distance calculations was the Great Circle Formula, one of several methodologies for determining straight line distances between two points on a sphere. Our code, shown below, was adapted from sample code available on the SAS Users website:

```

/* decimal degrees must be converted to radians, another geographic unit of location */

/* exact coordinates known, based on the StreetMap report of a longitude value > 0*/
if bene_x ne 0 then do;
  bene_long_rad = -1 * atan(1)/45 * bene_x;
  bene_lat_rad = atan(1)/45 * bene_y;
end;
/* zip or zcta only coordinates known */;
else
if bene_cenzip_x ne . then do;
  bene_long_rad = -1 * atan(1)/45 * bene_cenzip_x;
  bene_lat_rad = atan(1)/45 * bene_cenzip_y;
end;

/* repeat the process for the provider coordinates */

/* calculate distance in miles using the great circle formula */
Dist_miles = 3949.99 *
  arcos
  (
    sin( prov_lat_rad ) *
    sin( bene_lat_rad ) + cos( prov_lat_rad ) *
    cos( bene_lat_rad ) *
    cos( prov_long_rad - bene_long_rad )
  )
;
run;

```

CALCULATING DISTANCE STATISTICS

Zip Code vs. ZCTA

Before we proceeded to analyze our distance calculations, we compared the accuracy of the distances obtained using zip-code based coordinates to those obtained using ZCTA-based coordinates. The results were virtually identical:

Provider Success	Beneficiary Success	Distance (Miles)					
		1999 Zip Code Coordinates			2000 ZCTA Coordinates		
		Mean	STD	Median	Mean	STD	Median
Street Address	Street Address	22.6	49.6	7.9	22.6	49.6	7.9
	Zip/ZCTA Centroid	40.0	62.8	18.2	40.0	61.7	19.1
Zip/ZCTA Centroid	Street Address	38.7	71.1	11.3	38.2	71.8	10.0
	Zip/ZCTA Centroid	42.8	69.6	17.1	42.7	68.5	18.3
All	All	30.5	58.5	10.4	30.9	59.0	10.5

This finding is important since it suggests that using ZCTA coordinates, which are the only zip code-like coordinates Census will be compiling in the future, do not significantly inflate the distances between points which can only be located by zip code.

Distance Means, Standard Deviations and Z-Scores

We were now ready to see if the distance calculations yielded information which might be useful in quickly screening Medicare claims data for potential anomalous provider activity. Our first step was to calculate several basic descriptive statistics for each of the providers in our proof-of-concept pool. We first screened the pool to include only providers with more than \$100,000 of allowed claims during the period under study.

We used the MEANS and SUMMARY Procedures to calculate means and standard deviations and the DATA step to use these statistics to calculate Z-Score statistics. The Z-Score is a parametric statistic which assumes that the underlying population is normally distributed. Z-Scores with an absolute value greater than 2 indicate that the mean of interest is more than 2 standard deviations removed from the population mean.

First we calculated the overall mean and standard deviation of the provider-beneficiary distances for the providers of interest.

```
proc means data = here.distance_use noprint;
  where prov_totallow > 100000;
  var dist_miles;
  output out = distall (drop = _type_)
    mean = totmean
    std = totstd
;
run;
```

Next we did the same calculations by provider. Note that we have mixed the use of the MEANS and SUMMARY procedures in this example to demonstrate that either can be used for calculations such as these:

```
proc summary data = here.distance_use2 nway missing;
  class prov_id;
  where prov_totallow > 10000;
  var dist_miles;
  output out = distprov (drop = _type_)
    mean = prov_mean
    std = prov_std
;
run;
```

Finally, we used the DATA Step to calculate Z-Scores:

```
data dist_stats;
  merge distall
        distprov
  ;
  /* the merge will attach the overall mean and standard deviation to the first record in
  the provider-level file. By using the retain statement and an assignment statement, we
  attach this information to each record in the provider file */
  retain mean
         std
  ;
  if _n_ = 1 then do;
    all_mean = totmean;
    all_std  = totstd;
  end;

  zscore = (prov_mean - all_mean)/all_std;

  label all_mean  = 'Overall Mean Distance, Miles'
        all_std   = 'Overall STD, Miles'
        prov_mean = 'Mean Distance, This Provider, Miles'
        prov_std  = 'STD, This Provider, Miles'
        zscore    = 'Z-Statistic'
  ;
run;
```

RESULTS

Screening the Distance Statistics

We now had mean and standard deviation statistics for each provider, along with a measure of how much each provider-beneficiary pool average distance deviated from the overall mean (our Z-Scores). We decided to rank the providers three times, once by each of our three statistics. We then selected the top 5% of providers in the mean and standard deviation ranking, and all providers with a Z-Score > 2. We then merged providers in the three rankings to create our final target group:

```
/* create three separate files, each sorted in descending order by the statistic of interest */
proc sort data = dist_stats out = by_mean;
  by prov_id descending prov_mean;
run;
proc sort data = dist_stats out = by_std;
  by prov_id descending prov_std;
run;
proc sort data = dist_stats out = by_zscore;
  by prov_id descending zscore;
run;

/* combine the statistics, by provider */
data forscreen;
  merge by_mean   (obs = 15)
        by_std    (obs = 15)
        by_zscore (obs = 15)
  ;
  by prov_id;
run;
```

This resulted in a list of 19 providers, each of whom was in the top 5% of provider-beneficiary distances based on either mean distance, the standard deviation of that mean, or calculated Z-score. Compared to the overall provider-beneficiary average distance of 30.5 miles (30.9 miles for those with total allowed dollars greater than \$100,000), the maximum mean distance in our list was 315.3 miles. Interestingly, only three providers had a mean distance more than 2 standard deviations above the group mean.

To assess the success of this strategy, we compared the 19 providers on our list to the list of providers from the same dataset identified as anomalous by more traditional data mining and analytical techniques. The list identified by non-geographic approaches contained 9 providers who had either been referred for internal investigation or to law enforcement for investigation. Our list contained 5 of the 9:

Provider	Anomalous by Non-Geographic Analyses	Screening			Distance Statistics			Allowed Amount (\$000)
		Z-Score (>2)	Mean (Top 5%)	Std (Top 5%)	Z-Score	Mean (miles)	Std (miles)	
E	***	*	*	*	4.79072	315.3	116.1	170
O		*	*	*	3.03443	211.1	97.2	370
R		*	*		2.02431	151.2	71.4	600
A	***		*	*	1.86459	141.7	99.6	1,880
I			*		1.54798	122.9	52.8	440
S			*	*	1.52398	121.5	103.3	560
D			*		1.37578	112.7	77.1	700
L			*	*	1.16205	100.0	115.9	420
K			*	*	0.95362	87.6	100.8	1,070
M	***		*	*	0.95284	87.6	124.4	1,720
P			*		0.92374	85.8	64.6	970
G	***		*		0.90183	84.5	54.0	1,800
Q			*	*	0.89631	84.2	91.6	1,300
N	***		*	*	0.81121	79.2	78.1	4,200
H			*	*	0.78226	77.5	90.5	590
C				*	0.35162	51.9	92.4	200
B				*	0.13907	39.3	78.5	104
J				*	0.10855	37.5	80.3	350
F				*	-0.11293	24.3	94.5	190
All						30.9	59.3	295,000

Our technique was remarkably successful at identifying providers worthy of investigation. As with any fraud, waste and abuse detection activity, simply being flagged by an anomaly-detection pattern is not in itself indicative of abusive, wasteful or fraudulent behavior. For example, we know that one of the flagged providers maintains multiple offices covering a large geographic area. Another legitimately provides mobile services.

Further studies will be necessary to determine if this result applies to other provider-beneficiary pools. We suspect that it will, because as analysts working with claims data frequently observe, "When providers is doing one thing wrong, they are doing several things wrong."

The final step in our proof-of-concept was to look back and assess whether the technique was cost effective.

Production "Costs"

Our proof-of-concept data contained 1.5 million claims records and 500,000 addresses. We processed on a desktop Pentium 4 with 1 gigabyte of memory, optimized for running ArcView Desktop GIS.

We assessed the overall costs for implementing our strategy in general categories, based on the time involved in each. Accessing and formatting Census data is straightforward. The files are available on the Bureau's website, along with format statements. Street-level geocoding becomes time-intensive when large volumes of addresses need to be processed. Our computing capabilities required over 48 clock hours to street-level geocode slightly more than one-half million addresses. Zip code-level geocoding, distance calculations, analytical statistics and screening were all automated using Base SAS. The longest run-times we encountered during the SAS processes were for the distance calculation algorithms. The program required 10 minutes to calculate distances for slightly more than 500 thousand provider-beneficiary combinations.

The largest cost associated with implementing geographic distance calculations in an analytical or data mining environment is therefore the time involved in street-level geocoding. This is clock time only, no manual intervention is required as the StreetMap software processes a DBF file. Otherwise, a simple set of Base SAS programs can complete the work in under an hour for a dataset similar in size to the one we used.

CONCLUSION

With the StreetMap USA geocoding tool, U.S. Census Bureau Gazetteer files and Base SAS at our disposal, we were able to successfully demonstrate that larger than average distances between providers and their beneficiaries is a quick and inexpensive method to screen for providers worthy of further investigation. It was not necessary that the magnitude of the mean distance differences be statistically significant, simply that they be in the top 5% of the provider-beneficiary rankings in terms of the absolute size of the mean distances or their standard deviations. Overall the technique was not resource intensive, with the exception of the clock time involved in street-level geocoding of nearly 500,000 addresses.

REFERENCES

ESRI Technical Support. *Personal communication regarding the expected success rate for street-level geocoding using files with addresses newer than 2000.* October, 2005.

U.S. Census Bureau. *Zip Code Gazetteer File, November 1999.* Available from the Census website (www.census.gov) by following the links: Geography>Tiger>U.S. Postal Service Zip Codes or by going directly to <http://www.census.gov/geo/www/tiger/zip1999.html>.

U.S. Census Bureau. *ZCTA Gazetteer File, 2000.* Available from the Census website (www.census.gov) by following the links: Gazetteer>Census 2000 Gazetteer Files> ZCTAs (Zip Code Tabulation Areas or by going directly to <http://www.census.gov/geo/www/gazetteer/places2k.html>.

U.S. Census Bureau. Information on ZCTAs is available on the Census website (www.census.gov) by following the relevant links in the Geography section or going directly to <http://www.census.gov/geo/ZCTA/zcta.html>

SAS Technical Support. *Sample: Distances Between Selected Cities, 2002.* Available in the Technical Support>Samples section of the SAS website (www.sas.com) or by going directly to <http://ftp.sas.com/techsup/download/sample/graph/gmap-distance.html>.

ACKNOWLEDGMENTS

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

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

The author wishes to thank the Data Miners of the Business Intelligence Center at SRA International, Inc. for the opportunity to work with them.

The opinions expressed in this paper are those of the author and not those of the U.S. Government Accountability Office.

CONTACT INFORMATION

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

Laurie Hamilton
U.S. Government Accountability Office
441 G Street NW
Washington DC 20548
202-512-5317
hamiltonls@gao.gov