SAS® GLOBAL FORUM 2017

April 2 – 5 | Orlando, FL

Distances: Let SAS Do the Heavy Lifting

USERS PROGRAM

Distances: Let SAS Do the Heavy Lifting

Jason O'Day, MBA

US Bank

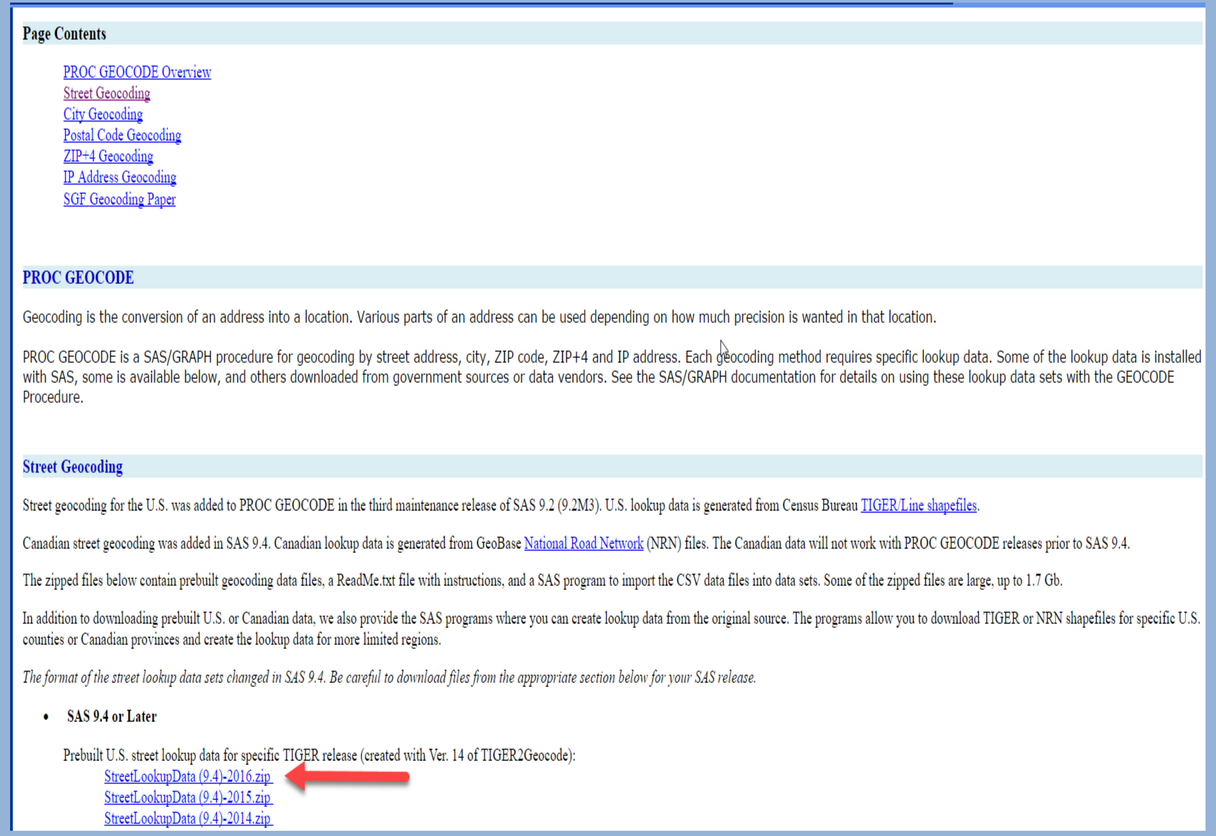
ABSTRACT

SAS® has a very efficient and powerful way to get distances between an event and a customer. Using the tables and code located at http://support.sas.com/rnd/datavisualization/mapsonline/html/geocode.html#street [(1)], you can load latitude and longitude to addresses that you have for your events and customers. Once you have the tables downloaded from SAS, and you have run the code to get them into SAS data sets, this paper helps guide you through the rest using PROC GEOCODE and the GEODIST function. This can help you determine to whom to market an event. And, you can see how far a client is from one of your facilities.

PROBLEM COMPLEX

There was a standing process that had been used in the past but the analysts that were running it wanted the SAS Developers to take it over and to update it for other portions of the business. After meeting with them I found that they were getting latitude and longitude from some 'mysterious file' and the distances were being calculated using a formula that was found online by the team. In taking this over I wanted to rewrite most of the process and look for ways to improve on what has been done. I also wanted to get to the bottom of this file that was used to obtain the coordinates. Below are the sample codes and the description of how it works in gathering distances from one location to another as the bird flies.

RESULTS (CLICK TO EDIT)



Page Contents

PROC GEOCODE Overview
Street Geocoding
City Geocoding
Postal Code Geocoding
ZIP+4 Geocoding
IP Address Geocoding
SGF Geocoding Paper

PROC GEOCODE

Geocoding is the conversion of an address into a location. Various parts of an address can be used depending on how much precision is wanted in that location.

PROC GEOCODE is a SAS/GRAPH procedure for geocoding by street address, city, ZIP code, ZIP+4 and IP address. Each geocoding method requires specific lookup data. Some of the lookup data is installed with SAS, some is available below, and others downloaded from government sources or data vendors. See the SAS/GRAPH documentation for details on using these lookup data sets with the GEOCODE Procedure.

Street Geocoding

Street geocoding for the U.S. was added to PROC GEOCODE in the third maintenance release of SAS 9.2 (9.2M3). U.S. lookup data is generated from Census Bureau TIGER/Line shapefiles.

Canadian street geocoding was added in SAS 9.4. Canadian lookup data is generated from GeoBase National Road Network (NRN) files. The Canadian data will not work with PROC GEOCODE releases prior to SAS 9.4.

The zipped files below contain prebuilt geocoding data files, a ReadMe.txt file with instructions, and a SAS program to import the CSV data files into data sets. Some of the zipped files are large, up to 1.7 Gb.

In addition to downloading prebuilt U.S. or Canadian data, we also provide the SAS programs where you can create lookup data from the original source. The programs allow you to download TIGER or NRN shapefiles for specific U.S. counties or Canadian provinces and create the lookup data for more limited regions.

The format of the street lookup data sets changed in SAS 9.4. Be careful to download files from the appropriate section below for your SAS release.

- **SAS 9.4 or Later**

Prebuilt U.S. street lookup data for specific TIGER release (created with Ver. 14 of TIGER2Geocode):

StreetLookupData (9.4)-2016.zip
StreetLookupData (9.4)-2015.zip
StreetLookupData (9.4)-2014.zip

Distances: Let SAS Do the Heavy Lifting

Jason O'Day, MBA

US Bank

RESULTS CONTINUED (CLICK TO EDIT)

```
| Summary  : 1) Download Census Bureau TIGER/Line files for the desired US
|               counties and states from www.census.gov/geo/www/tiger.
|            2) Unzip all downloaded TIGER files into a single directory.
|            3) Set required TIGERPATH macro var to that location.
|            4) Set required DATASETPATH macro var to directory where the
|               final PROC GEOCODE lookup data sets are to be written.
|            5) Specify desired optional macro vars from above list.
|            6) Submit TIGER2GEOCODE.sas file to compile macro programs.
|            7) Invoke the TIGER2GEOCODE macro program.
|            8) Check log for errors or warnings.
|            9) Review geocoding lookup data sets.
|            10) Test lookup data using PROC GEOCODE street method.
|                See PROC GEOCODE documentation.
|            11) Create backup of PROC GEOCODE lookup data sets.
|            12) If no longer needed, delete temporary files and data sets:
|                   a) Downloaded TIGER/Line zip files
|                   b) Unzipped TIGER files in TIGERPATH
|                   c) Individual county data sets in WORK
|                   d) Interim files and data sets in WORK
|            12) Lookup data sets are ready for street level geocoding.
|                See LOOKUPSTREET= option in PROC GEOCODE doc for instructions.
```

PROC GEOCODE

In the code below you can see that we need to assign the METHOD and the variables that follow in the PROC GEOCODE process.

The fields: ADDRESS, STATE, CITY and ZIP_CD are from the dataset UNQ_ENVT_LOCATION, which is the input table that we need to match with the USM table in order to get the latitude and longitude variables.

```
PROC GEOCODE DATA=WORK.unq_envt_location /* Input table that needs Coordinates table */
            OUT=WORK.event_geo           /* Output table that needs Coordinates table */
            METHOD=street                /* METHOD Type (street in our case) */
            LOOKUPSTREET=sasdata.usm     /* SAS Address lookup table */
            ADDRESSVAR=address           /* Address variable in input data set */
            ADDRESSSTATEVAR=state        /* State variable in input data set */
            ADDRESSCITYVAR=city          /* City variable in input data set */
            ADDRESSZIPVAR=zip_cd         /* Zip Code variable in input data set */
            ;
  RUN;
```

Distances: Let SAS Do the Heavy Lifting

Jason O'Day, MBA

US Bank

PROC GEOCODE (CONT)

You can see that field 'Y' is latitude and 'X' is longitude. Other fields that we can use are '_MATCHED_' and '_SCORE_'. _MATCHED_ will tell you what field it was matched on whether it was ADDRESS, ZIP or CITY. '_SCORE_' gives a numeric value of how well it matched from USM to your dataset, the higher the better.

```
PROC SQL;
        CREATE TABLE WORK.event_geo1 AS
        SELECT DISTINCT
                y                       AS lat
                ,x                      AS lon
                ,_matched_
                ,_score_
                ,submarket
                ,address
                ,city
                ,state
                ,zip_cd
        FROM WORK.event_geo
        WHERE UPCASE(_matched_) ^= 'NONE' /*Keeping only where there is a Match*/
        ORDER BY submarket
 ;QUIT;
```

GEODIST

GEODIST(latitude-1, longitude-1, latitude-2, longitude-2 <,options>) [3]

The options are either 'K' for kilometers, 'M' for miles, 'D' for degrees or 'R' for radians.

In the code below we are joining the tables and calculating the distances in miles and are only keeping the data were the distance is less than 10 miles and it is not null.

```
PROC SQL;
        CREATE TABLE WORK.mbr_evnt_geocoding AS
        SELECT
                A.*
                ,E.submarket
                ,E.lat                  AS event_lat
                ,E.lon                  AS event_lon
                ,E.state
                ,GEODIST(A.lat, A.lon, E.lat, E.lon, 'M')    AS distancetoevent
        FROM WORK.mbrs_geo1 AS a
                LEFT JOIN WORK.event_geo1 AS e ON A.state = E.state
        WHERE CALCULATED distancetoevent <=10
                AND CALCULATED distancetoevent ^=.
        ORDER BY A.person_id
 ;QUIT;
```

Distances: Let SAS Do the Heavy Lifting

Jason O'Day, MBA

US Bank

CONCLUSIONS

In summary, this paper can really be broken down to a few components that will make your life easier as far as gathering the coordinates for locations and then calculating those differences using SAS. It is made easy because SAS first gives you the latitude and longitude files as well as the code to load them to your environment. Second, SAS gives you the power of PROC GEOCODE which we have barely scratched the surface on in this paper. If you want more information on this procedure I suggest reading a few of the papers by Darrell Massengill and Ed Odom. Finally, SAS gives the ease of calculating the distances using the GEODIST function. You can try gathering the coordinates yourself and even using complicated calculations to get the distances, but wouldn't you prefer to have SAS do the heavy lifting?

SAS® GLOBAL FORUM 2017

April 2 – 5 | Orlando, FL