Distances: Let SAS Do the Heavy Lifting
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.

RESULTS (CLICK TO EDIT)

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.
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 */
    ADDRESSTYPEVAR=state /* State variable in input data set */
    ADDRESSECVAR=city /* City variable in input data set */
    ADDRESSZIPVAR=zip_cd /* Zip Code variable in input data set */
;
RUN;
```
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;
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?