Finding the Treasure: Using Geospatial Data
for Better Results with SAS® Visual Analytics

Tricia Aanderud, Zencos Consulting

ABSTRACT

Traditional business intelligence systems have focused on answering the *who, what*, and *when* questions, but organizations need to know the *where* of data as well. SAS® Visual Analytics makes it easy to plot geospatial data, which can add a completely new element to your data visualizations and analysis. When looking at a tabular report, you notice multiple columns that represent customers, competitors, and demographic information. But when you place the same geocoded data on a map, new insights jump off the page! Perhaps you see where the better customers are, where they are in relation to competitors, and the regions that provide the most market potential based on underlying demographics. These finding are about the *where* of the data. It's like creating a treasure map.

In this session, you will learn how to use the SAS Visual Analytics 8.2 geospatial objects, how to create custom geographic data items to use with the maps and see many examples of how others have incorporated maps for snazzy data presentations.

INTRODUCTION

One of the highlights of SAS Visual Analytics is how seamless it is to start data analysis with features like maps or the fancier term - geospatial analysis. Geospatial analytics is the ability to gain insights from the location component of your data. In this paper, you will learn how to use the SAS Visual Analytics geomap object, how to create geographic data items, and then we will review multiple geo objects, such as the one in the following display, to gain insights and ideas about how you can use geospatial analytics.

Europe is an international business center and a leading tourist destination. If you wanted to tell a data story about how popular of a location it is, you could start by exploring the airport traffic. The Anna Aero website (*http://www.anna.aero*) contains data that details trends from most of the world's airports. You can use the airport codes and passenger counts to start a data story about the most and least popular cities.

In Display 1, the location of each airport is shown along with the passenger count and the difference from the previous year. In this example, location enhances the story. Instead of fancy calculations, the viewer can simply use their eyes to search out patterns.

There are multiple observations. You might notice the darker circles that
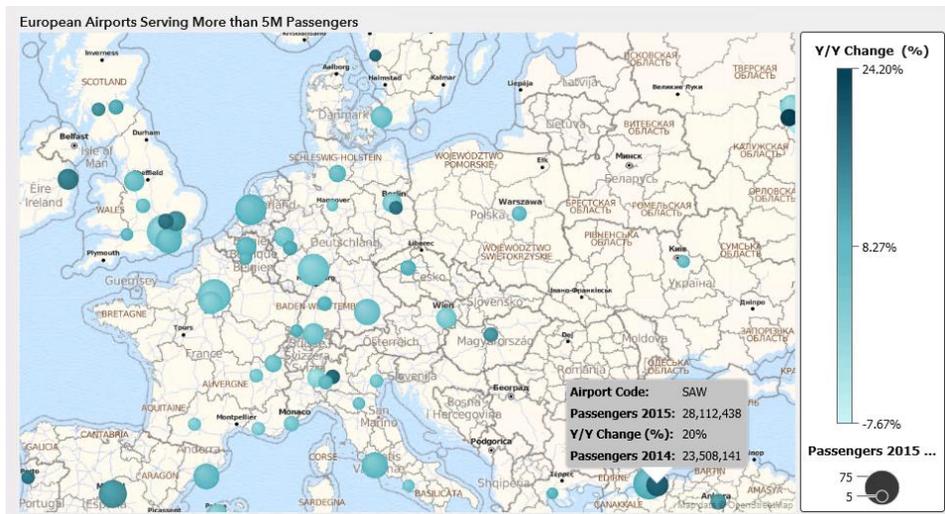
Display 1. Understanding European Airport Traffic

show increased passenger counts. Perhaps you notice where there are multiple airports within a close radius and you are curious why one has a higher passenger count. When used effectively, geospatial data can reveal previously unknown patterns or assist with confirming suspicions.

WHAT PUTS THE "GEO" IN THE DATA?

Your data may already contain geo-information. In the following display, the data set contains several fields: customerNumber and several geodata types; street, city, state, postal codes and country values. Any of these values can be used to plot values on a map.

customerNumber	addressLine1	city	state	postalCode	country
181	2678 Kingston Rd.	New York	New York	10022	United States
198	16780 Pompton St.	Brockton	Massachusetts	58339	United States
201	12, Berkeley ...ens Blvd	Liv...	Live...	WX1...	Unit...gdom
202	1900 Oak St...	Va...er	Vict...	V3F...	Can...
204	7635 Spinnaker Dr.	Brockton	Massachusetts	58339	United States
205	78934 Hillside Dr.	Pasadena	California	90003	United States

Display 2. Data with geospatial components

In Display *3*, a subset of the data set is plotted by street address along with some data about the customer spending habits. Using location, you can understand them better as a group.

Perhaps you want to understand where the customers live in relation to the store (shown with red arrow). In the bottom left, the grouping of buildings suggests a townhouse community. How does seeing that pattern assist you with analysis? You might not see that same pattern if you were just looking a bar chart. The location analysis reveals previously unseen patterns.
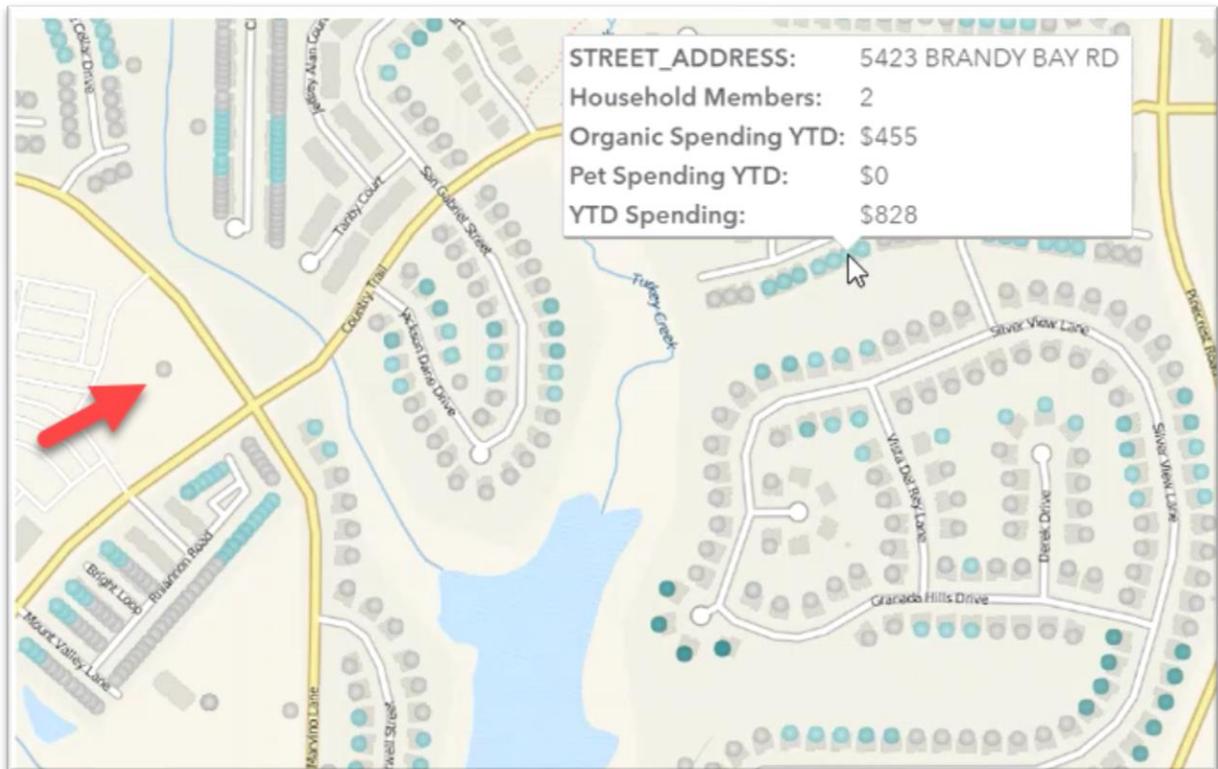
Display 3. Customer locations in relation to your store

CREATING GEOSPATIAL DATA ITEMS

Before creating a geomap visualization, you must have a geographic data item. There are two methods for creating a geographic data item: pre-defined and custom. If a data item contains a location, such as a country or state, then you can create a pre-defined value. If you want to use a street address or specific location, then you'll need to create a custom data item. This topic shows you how to use each method.

USING PRE-DEFINED GEOGRAPHIC DATA ITEM

To keep things easy, SAS Visual Analytics has predefined geographic data elements ranging from the general values, such as country names, to specific values, such as ISO country codes. If your categorical data item matches one of the pre-defined values, then you can use it to make a geographic data item. The pre-defined values have the geo-coded information stored so you don't have to know the exact location – SAS Visual Analytics does that work for you.

In the following example, let's change the country location of our customer into a geographic data item. There are 9 geographic data item choices ranging from country name to US Postal Code. Your data item is the country name spelled out, so it matches the **Country or Region Name** choice.

Display 4. Create a geographic data item using a predefined value

Here's how to create your data item:

1. Click on a location-based data item, such as customer country.
2. Select Geography from the Classification drop-down list. The Geography Item window appears.
3. Select **Country or Region Names** from the Name or code context drop-down list. This pre-defined value matches our data.
4. Click OK to create the geographic data item in the Geography measures area

## CREATING A CUSTOM GEOSPATIAL DATA ITEM

All geospatial data items represent a location on the Earth. A specific point has a set of coordinates called latitude and longitude. When using a location's latitude and longitude, you are referencing the location.

To create a custom geographic item, you must have the latitude and longitude coordinates available in the dataset. Some datasets already have the latitude and longitude available. If you want to add this information you can choose from multiple methods, which we will discuss shortly.

Locating a Customer with Coordinates

It's possible to describe any location on Earth with a set of coordinates. In the following display, the map shows the location of a customer called Australian Collectables in Melbourne, Australia.

Customer Name ▲	xLAT Customer	xLong Customer	Territory
Asian Treasures, Inc.	52.39	-7.84	EMEA
Atelier graphique	330.51	-10.90	EMEA
Australian Collectables, Ltd	-871.39	3338.76	APAC
Australian Collectors, Co.	-2081.94	7974.06	APAC
Australian Gift Network, Co	-412.23	2295.02	APAC
Auto Associés & Cie	878.47	38.36	EMEA
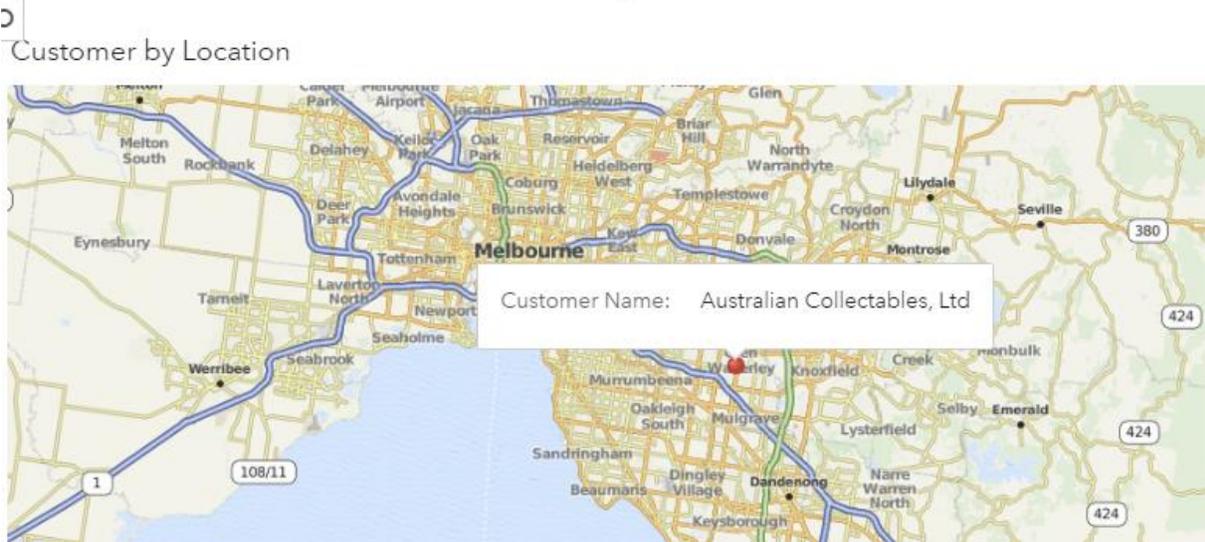
Display 5. Using latitude and longitude to locate a customer

Above the map we see a table that contains its Latitude and Longitude coordinates. These values are based on the imaginary lines that describe the globe. When you have these two values you can create a custom geographic item.

Adding the Custom Coordinates to Your Geo Data Item

In the previous example, the latitude is contained in the **xLat Customer** data item and the longitude is contained in the **xLong Customer** data item. These values can be associated with any categorical data item that makes sense.

In this example, we use the Customer Name, but we could have also used the Customer Code or Order Number if that better fit our report. This value is what the users see when they hover over a coordinate point (as shown in Display 5).

Here are the steps:

1. Use any categorical value that makes sense. We are using company name.

2. Click new data item, then select **Geography**.

3. Select your data items for latitude and longitude in the appropriate fields. Your new data item appears in the Geography area.

Tip! When naming your latitude and longitude raw data, use a x or z so the value sorts toward the bottom of your Measures.

LOOKING UP GEOSPATIAL COORDINATES

To geocode an address, you add the geographic coordinates (latitude and longitude values). When geocoding a city, county or postal code, the coordinates represent the center of the geographic region. Here are two ways you can geocode addresses.

Using the GEOCODE Procedure

The GEOCODE PROCEDURE allows you to lookup a variety of geospatial locations: address, city, or postal code. Using your input data set, the procedure tries to lookup the matching coordinates.

This example is from the SAS documentation. It uses the SASHELP.GEOEXM to look up the street address and even assign the census tract. Very complete information!

```
proc geocode                          /* Invoke geocoding procedure   */
 method=STREET                        /* Specify geocoding method    */
 data=WORK.CUSTOMERS                   /* Input data set of addresses */
 out=WORK.GEOCODED                    /* Output data set with X/Y values */
 lookupstreet=SASHELP.GEOEXM          /* Primary street lookup data set */
 attributevar=(TRACTCE00);            /* Assign Census Tract to locations */
run;
```

There are many good papers that explain how to use the GEOCODE procedure. I've listed the ones I found most useful in the Recommended Reading section at the end of this paper. Also note that SAS Support covers this topic in-depth at their Maps Online website.

Using Third-Party Lookup Services

Third-Party databases provide application programming interfaces (APIs) that allow you to send an address and have it return the geo coordinates for the address. The address may be complete with the house number and street name or it can just be the city. A quick internet search reveals multiple services are available.

Google hosts a geo lookup API that you can use for free. The company limit the number of daily lookups, so you can purchase the service if you need to handle a larger population. To start using the service, you'll need to register for a Google API key. You can learn how to do that and more about the service from the Geocoding API site.

The following display shows my example data set. I used this data set as input to a macro that looks up each address using the Google GEO API. The complete code is available in Appendix A.

customerNumber	addressLine1	aggre ssl i	City	State	postalCode	Country
103	54, rue Royale		Nantes	Pays de la Loire	44000	France
112	8489 Strong St.		Las Vegas	Nevada	83030	United States
114	636 St Kilda R...	Level 3	Melbourne	Victoria	3004	Australia
119	67, rue des Cin...		Nantes	Pays de la Loire	44000	France
121	Pellygata 35		Sarpsborg	Ostfold	1706	Norway
124	738 A St.		San Rafael	California	94901	United States

Display 6. Input data set for the Google Geo API

In this data set, I have a column for each address variable. The macro concatenates all address variables into one variable called address. Note that any extra spaces are removed and an asterisk with plus sign is used to separate the address elements. This is how the API expects to receive the address:

```
myAddress=catx(
        translate(strip(addressline1),'+',' '),'*+',
        translate(strip(city),'+',' '),'*+',
        translate(strip(state),'+',' '),'*+',
        translate(strip(country),'+',' '), '*+',
        translate(strip(postalCode),'+',' '));
```

The LookupLatLong macro is called in the next line. We need a way to join the resulting information to our original data set. In this example, the customerNumber variable is passed to the macro with the address. This variable is renamed to yourJoinVar:

```
CALL EXECUTE(
'%LookupLatLong('||%NRBQUOTE(myAddress)||','||customerNumber||');' );
```

After running the code, the output dataset contains three new variables. The Status variable contains the result from the Google API. If the status was OK, the xLat and xLong fields are populated. If the address was not found, the status indicated ZERO_RESULTS. For instance, the address for customer number 112 was not found. Use the xLag and xLong variables in your data set so you can plot a custom geo data item with your report.

status	xlat	xlong	address	yourJoinVar
OK	47.21565390	-1.55674480	54 Place Royale, 44000 Nantes, France	103
ZERO_RESUL...	.	.		112
OK	-37.85344990	144.98282680	636 St Kilda Rd, Melbourne VIC 3004, Austr...	114
OK	47.21993860	-1.55575060	67 Cours des 50 Otages, 44000 Nantes, Fra...	119
OK	59.28331290	11.12001270	Pellygata 35, 1706 Sarpsborg, Norway	121
OK	37.96522820	-122.5088795	737 Francisco Blvd E, San Rafael, CA 9490...	124

Display 7. Results from the Google Geo API lookup

Using GeoMap Objects

Once you have the geographic data item then you can create a geomap object. In SAS Visual Analytics 8.*x*, there is an object called geomap. This object allows you to create three different maps based on your data. Each one of the objects can be used to bring different viewpoints to the story. In this topic, you'll how to create map object with example of each map type.

CREATING A GEOMAP DATA OBJECT

If you pull the geographic data item to the canvas, SAS Visual Analytics creates the map for you automatically. You can then select which map type you want to use from the Options > Map > Type drop-down list. There are three choices: Bubbles, Coordinates, and Regions.


Display 8. Adding a Geomap Data Object

GET TO THE POINT WITH COORDINATES MAP

Perhaps you've heard people talk about tornado alley – it's an area down the middle of the United States where tornadoes occur more frequently. Tornadoes are powerful and scary storms that produce wind speeds capable of sending a wood board through a metal car door. An EF5/F5 tornado has a strength of 230 mph+ (370 kph) winds!

These storm events are responsible for massive property damage and loss of lives. Coordinate maps are excellent at showing exactly where events occur. In the following display, the markers indicate where EF4(red) and EF5 (blue) tornadoes touched down.
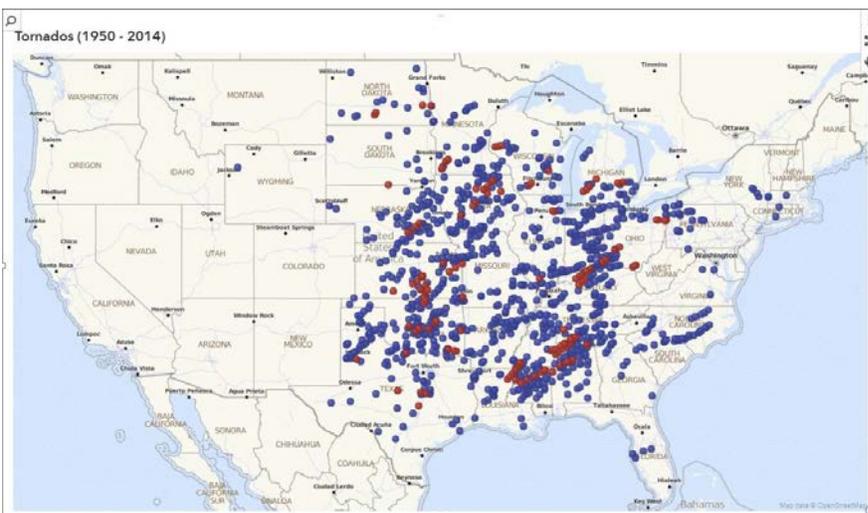
While the data points are chaotic – it's clear where severe tornados occurred.

This data would not have had the same impression if we had plotted it as a line chart or bar chart.

The touchdown points help you realize why those states have a higher disaster recovery budget.

Display 9. Plotting tornados with a coordinate map

COMPARE AREA DIFFERENCES WITH REGIONAL MAPS

Use the Regions map when you need to introduce a subject about location. This geospatial object helps a viewer understand where to focus their attention or understand how much variation occurs for a value. These objects are also called *choropleth* maps, which is Greek for multitude of areas.

When you start thinking about dangerous storm events, you can imagine that these events cause considerable property damage. States more prone to severe tornados will plan larger disaster recovery budgets. It would be interesting to compare the damage costs by state. Using a Geo Regional map, you can place a value over an entire region, such as a country or a state. Color is then applied over the regions to indicate the intensity of the value.

In the following figure, you can see the associated property damage cost for the tornadoes across the areas. The darker the color the costlier the storm damage. Use an average or percentage to make the values comparable or normalized. By exploring the visualization, you can easily see the areas of most damage but it's harder to understand where there is the least damage. Ensure you use a legend so the user understands the color range.
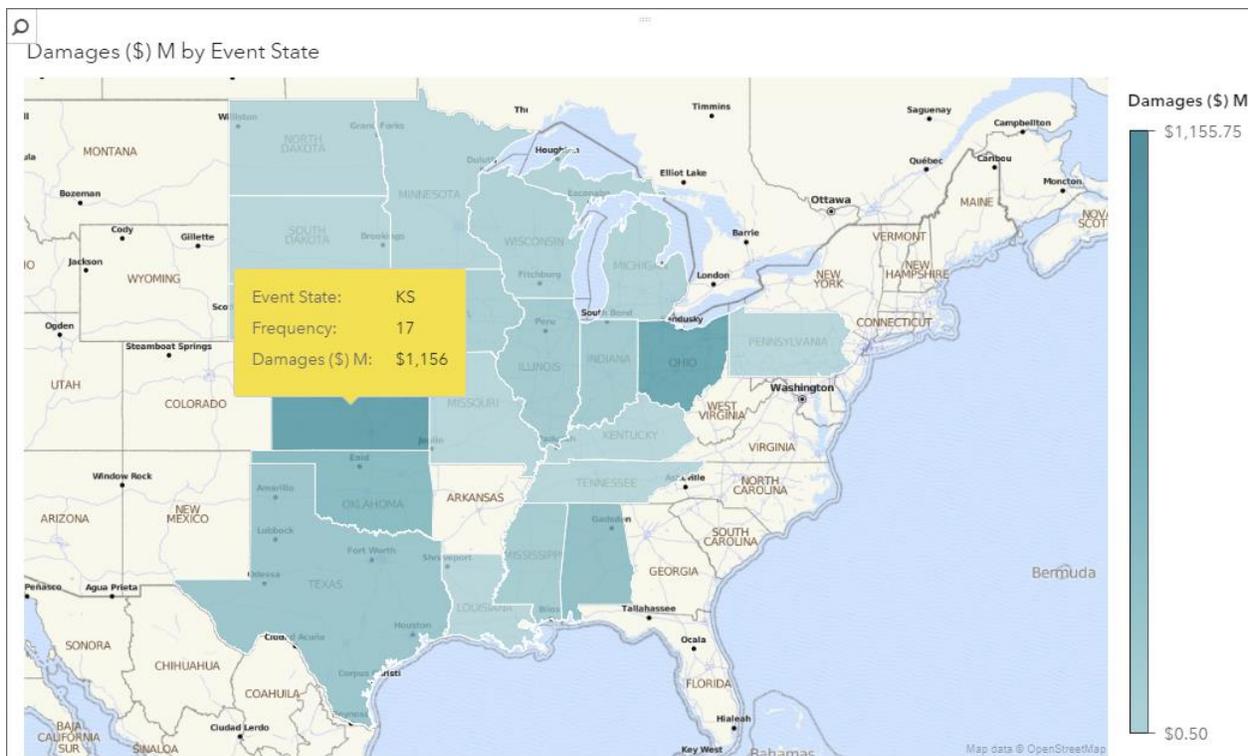
**Display 10. Regional maps show the property damage by state**

When you hover over each state, a data tip appears that contains the assigned data items values. Since Ohio and Kansas are similar in color, viewers might be interested to learn more. Most of Kansas is farm land and rural areas, while Ohio is more densely populated and industrial. Being in tornado alley, Kansas probably experiences more tornadoes and thus more crop damage. With the larger population, it might be costlier for Ohio when there is an extreme storm event.

SHOW OVERALL TRENDS WITH BUBBLE MAPS

Bubble charts receive a lot criticism for being difficult to understand. These charts can pack a lot of data into a few variables. While a layperson might spend more time trying to understand a bubble chart, it doesn't seem true for the bubble maps. Possibly it's because the user sees the map and understands it is related to location.

In the previous topic, we created a regional map to show the average damage cost from F5 tornadoes for each state. One issue with the method was that users had to hover over each state to see how many storm events were associated with each event. If a user wants details, it is a little awkward. A geo bubble map resolves this issue using bubbles.

A bubble plot places a bubble on the geographic location and allows you to control two aspects of the bubble – its size and color. In the following example, the bubble size is the event count (meaning number of tornadoes) while the color shows the estimated property damages (shown with the scale).
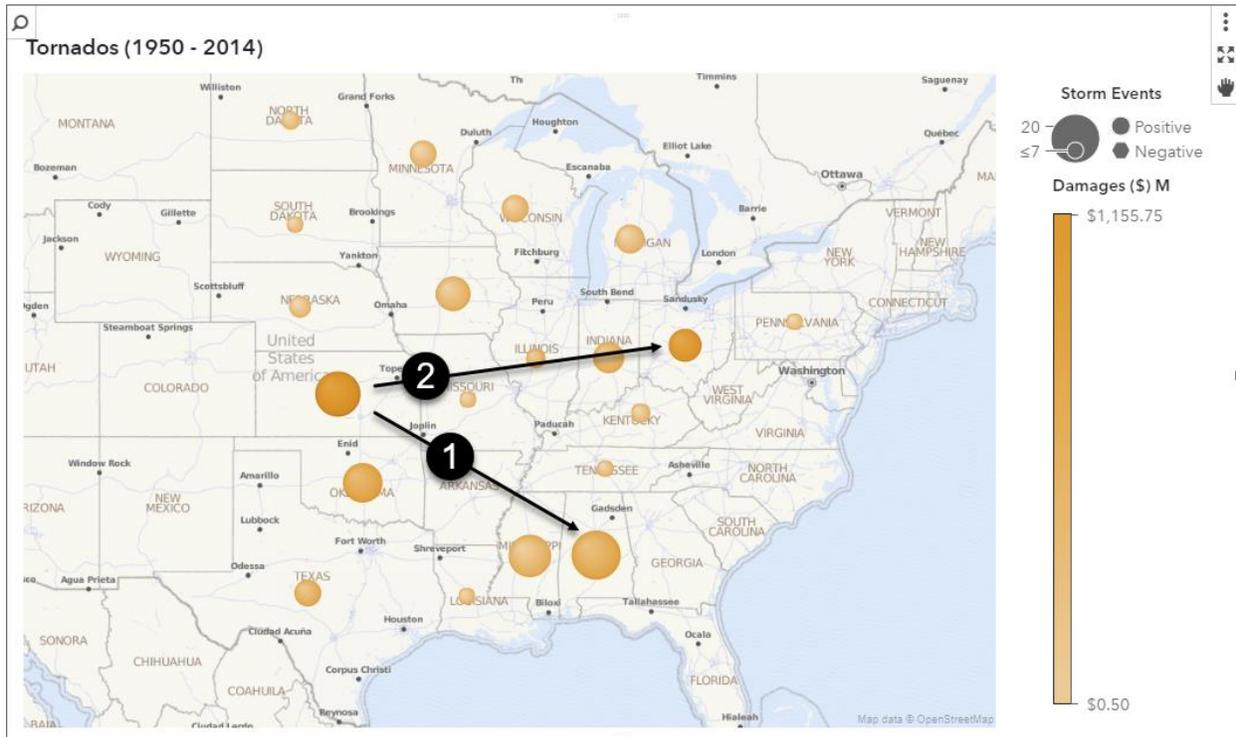
Display 11. Bubble plots show the number and damage amount

This geo image provides a richer picture. Consider Kansas in the middle of the US – there are some bubbles similar in size and color. However, we interpret these differently.

Alabama (shown with the 1) is a lighter color (meaning slightly less money) but appears to be larger (so more storms). While Ohio (shown with the 2) is just as dark (so similar costs) but is smaller so fewer storms. Ohio is more densely populated than Kansas and has several large cities. Any storm events there would be costlier due to the damage to the buildings.

OTHER SNAZZY GEOMAP EXAMPLES

Here's some of the various ways you can update or change the geo maps to highlight your data.

USING ESRI AS THE MAP SOURCE

SAS Visual Analytics can use OpenStreetMap® or Esri® as the map service. Esri provides a variety of background choices that can approve the appearance of your map or provide additional analysis information.

Adding Supporting Charts

Most geospatial object benefit from having other charts that assist with supporting the information. In this display, the shark incidents between 1997-2017 are plotted. The larger bubble at New Smyrna Beach reveals that of 203 encounters, not one of the attacks were fatal but most occurred when the person was surfing. An article from the Daytona-Beach News-Journal noted that there were 9.5 million visitors in 2016 alone. Considering that number, the incidents are much rarer than the media would have you believe.

When a viewer clicks on a map bubble, the donut and bar charts are updated. This gives the viewer a way to explore the data and gain new insights. The interactive features of SAS Visual Analytics allow you to click on the bubble and have the charts on the side update.
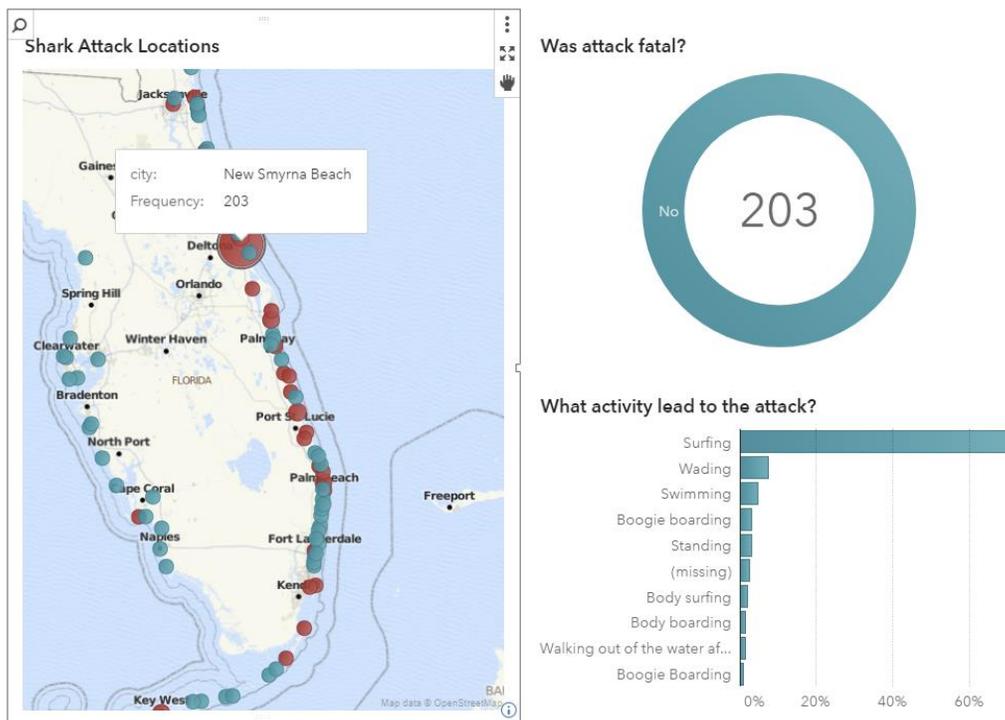

Display 12. Supporting a geomap with other data

To add interactivity between your data object, you use the Actions menu. It allows you to select which data object (geo map) causes the other data objects (donut chart, bar chart) to be filtered based on the selection.

The user can explore the map with their eyes and click on the areas of interest to understand the data more.

Changing the Background Map

While it's clear that many shark attacks occur at New Smyrna Beach, it also shows areas where there are no reported attacks. Hint: It's the Everglades! The OpenStreetMap doesn't reveal that relationship as well as the Esri World Topographic Map. This map background shows lists provides more information about the landscape.
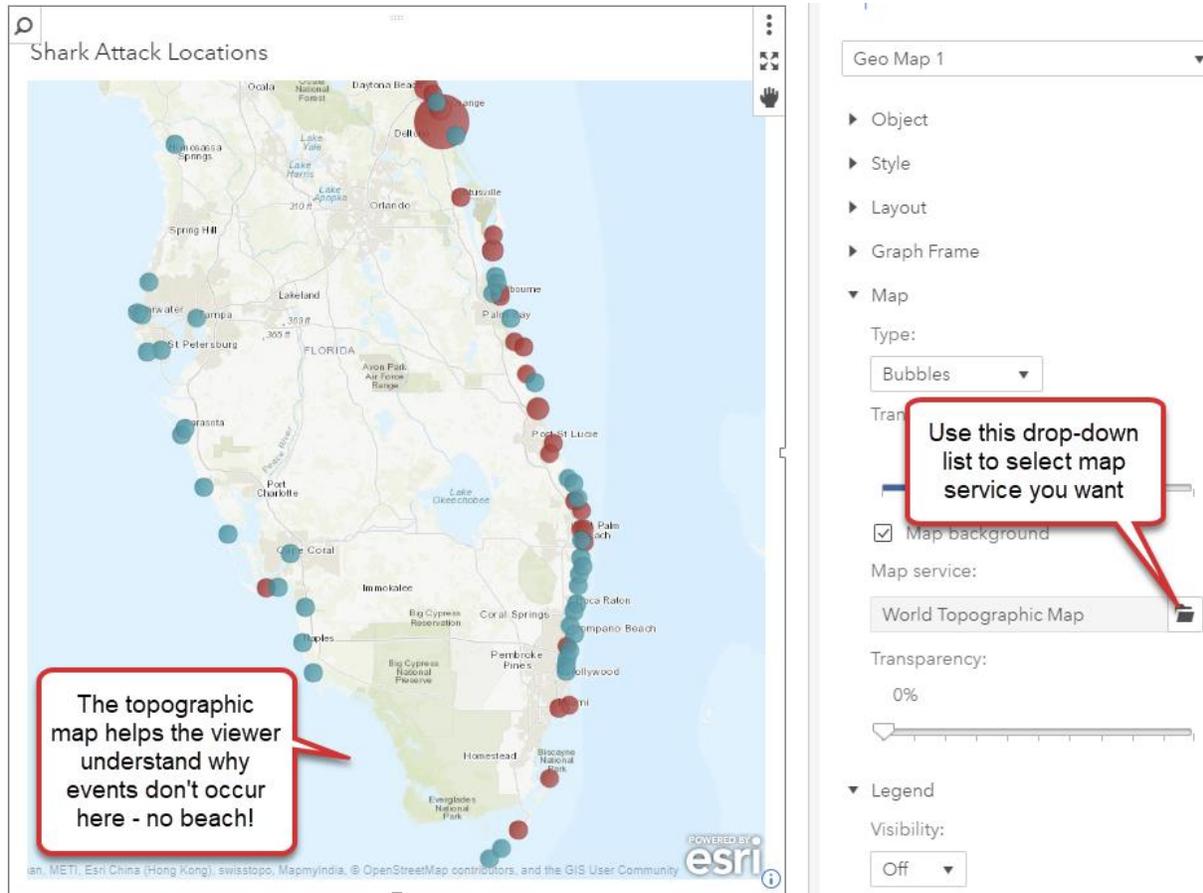
Display 13. Shark incidents in Florida

In the Options area, you can select the Map service from the pop-up window. For this example, I am using the World Topographic Map. This background helps the viewer understand the major landmarks. It's important to see that there are no shark attacks around the Everglades. This area is quiet literally a swamp – so not much a beach attraction except to the alligators.

Also note that the Transparency slider bar under the Map Service box. This feature can be used to control how bright the background map is. If you are using an especially bright background – this allows you to lessen the tone, so the data shows through.

Note: This functionality was introduced with SAS Visual Analytics 8.1.

Clustering Your Results

If you are using a coordinates map, you can choose to cluster the results. Clustering moves the individual points into a clustered area based on distance. In the following display, I used the shark incidents data from Display 13. When clustered markers are enabled, each marker shows a number that indicates the number of points represented. If you zoom or pan around the map, the clustering is reevaluated.

To change your custom coordinate map to using clustering, do the following:

1. Click the **Cluster adjacent markers** checkbox.

   <optional> You can use the Maximum clusters box to adjust the number of clusters allowed and the Cluster Spacing bar to indicate the amount of distance between the clusters. Generally, a lower value creates more clusters and a higher value creates fewer clusters.

   Tip! The display rules may change the bubble color if you have one set.

2. When using the clustering maps, sometimes it makes more sense to change the background. For this geomap the cities and roadways are less important. The story is more about the ocean than the land. I chose the Light Canvas Base for the background.

Display 14. Clustering the Coordinates

Adding Network Analysis

The network analysis object shows the relationship between two entities with a series of linked nodes. This data visualization technique is more commonly associated with social network analysis. You can add a background map to a Network Analysis object.

In this example, I wanted to see how close the customers were to each office. I used the postal codes for each customer and office as the geographic items. The size of each node is based on the total revenue from each customer. I wanted to understand where the larger customers were located and if we needed to change our strategy for them. The other benefit was the size also made the office node larger because it was the value of all customer orders. Quickly you'll see that the San Francisco and Paris offices generate the most revenue.

Check out the video to see the analysis and then how I created the network analysis object.

Display 15. Network Analysis

Changing the Coordinate Look

You can change the way your coordinates appear. For instance, this analysis grouped consumers in a neighborhood. Adding the houses helps the viewer to cement in their minds – these are customers or people in their homes – not just data points. From the Styles menu, you can select Initial marker shape drop-down to select the shape you want to represent your coordinates. There are several shapes available such as stars, diamonds, and hexagons.
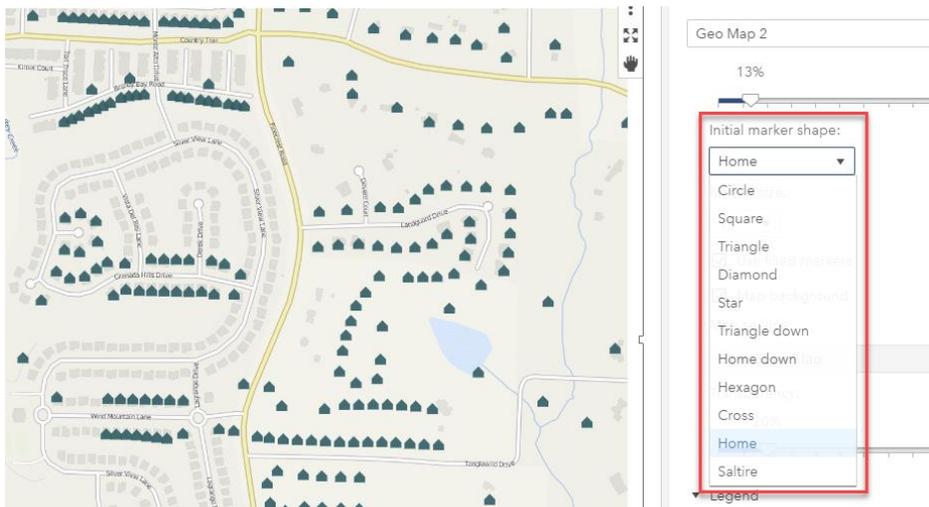

Display 16. Using different marker shapes

Adding Your Own Boundaries

You can use shapefiles to outline custom areas. A shapefile is a file format that contains geospatial vector data. You can think of it as the directions for drawing a shape – such as a county. Developed and regulated by Esri as a (mostly) open specification, the shapefile format spatially describes geometries as either points, polylines, or polygons.

The following shapefile created the outlines for the counties in North Carolina for this regional map.
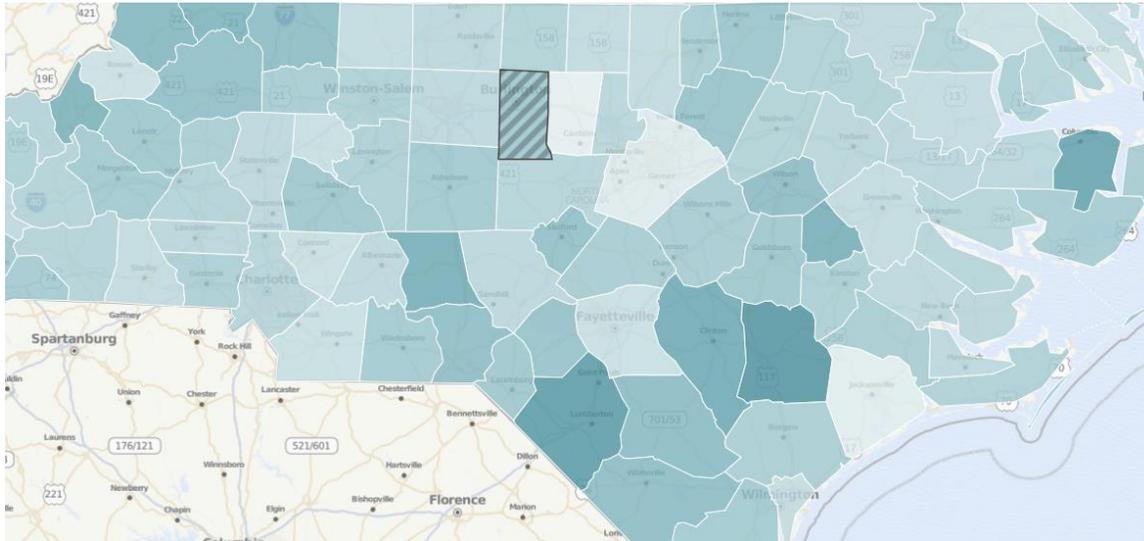
Display 17. Example of NC Shapefiles plotted in SAS Visual Analytics

You can get shapefiles from many different sources. The US Census Bureau is one of the more popular spots. When you want to add the shapefile to SAS Visual Analytics, you just need to ensure that you have a unique value on each line – so you need to add a sequence number. Use the following steps to turn a shapefile into a SAS dataset.

Change the Shapefile into a SAS Dataset for CAS

The shapefile contains the North Carolina (NC) counties listed by county name and county FIPS code. The US Government assigns a FIPS code to all US areas and the counties located within those areas. It's easier to join by FIPS code than names – since you don't have to worry about spelling or capitalization issues.

In the following example, we change a NC shapefile into a SAS dataset and add the unique row identifier. The data set is then ready for SAS Visual Analytics! Once you have your shapefiles, do the following in a code editor, such as SAS Studio:

```
proc mapimport
        datafile="/myPath/UScounties.shp"
        out=work.shapefile_uscnty;
        rename name=cnty_name;
run;

data shp.shapefile_uscnty_nc;
set work.shapefile_uscnty;
  where state_fips='37'; /*North Carolina FIPS Number*/

/*Just add the sequence in a data step */
seq_no=_n_;

run;
```

Using the Shapefile in SAS Visual Analytics

To use a shapefile in your report.

1. Load the new created NC county dataset (shapefile) into a CAS library.

2. Load your report data set that you want to use.

   Your data must contain either county name, federal FIPS, or county FIPS that matches the shapefile data you converted. If you do not have a data item that matches the NC County shapefile dataset – this example will not work.

3. Create a custom geography data item (as shown earlier).

   When creating the custom geographic data item, you must associate it to the shapefile. You can indicate the correct shapefile to use by selecting Custom polygonal shapes and selecting your table and variable names as appropriate.

CONCLUSION

As you have seen from the examples, there were many instances where the map was what made the difference in the analysis. When used effectively, geospatial data can reveal previously unknown patterns or assist with confirming suspicions. You could see where shark incidents were more likely to occur. You could confirm the location of the famed Tornado Alley. These examples were about the *where* of the data.

Best of all – it's simple to do! If your data has geo-based data, you can create a geo data item using the geographic items provided by SAS Visual Analytics. If you want a more custom look, then add geo-coordinates to your dataset. With SAS Visual Analytics 8.2, its easy to create geo maps and tailor the content to make your data visualization beautiful and useful.

REFERENCES

[1] OpenStreetMaps. Shapefiles Wiki. Available at: https://wiki.openstreetmap.org/wiki/Shapefiles

[2] SAS Maps Online. Available at: http://support.sas.com/rnd/datavisualization/mapsonline/html/geocode.html

[3] *Another record year for Volusia*, Flagler tourism, Available at: http://www.news-journalonline.com/article/LK/20160222/News/605064980/DN/

RECOMMENDED READING

- Aanderud, T., Collum, R., and Kumpfmiller, R., 2017. *Introduction to SAS Visual Analytics*. Cary, NC: SAS Press.

- SAS Communities - https://communities.sas.com/t5/SAS-Visual-Analytics/bd-p/sas_va

- SAS Institute Inc. 2018. SAS Visual Analytics 8.2: User's Guide. Cary, NC: SAS Institute Inc.

- Batkhan, L. SAS to the rescue: claiming your location on Google map by address only (without knowing latitude and longitude), Available at: https://blogs.sas.com/content/sgf/2015/11/03/sas-to-the-rescue-claiming-your-location-on-google-map-by-address-only-without-knowing-latitude-and-longitude/

ACKNOWLEDGEMENTS

CONTACT INFORMATION

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

Tricia Aanderud
Zencos Consulting
taanderud@zencos.com
http://www.zencos.com

APPENDIX A. LOOKING UP STREET ADDRESSES WITH GOOGLE

Use the following code to lookup street addresses with Google. You must have a Google API key assigned by Google, which is available from their website. This code is offered as is – you are free to make any changes to the code that make sense.

CODE FLOWCHART

This code is a little tricky in concept if you have never used a macro to lookup information. The following flowchart explains how your data set addresses are fed to the LookLatLong macro.
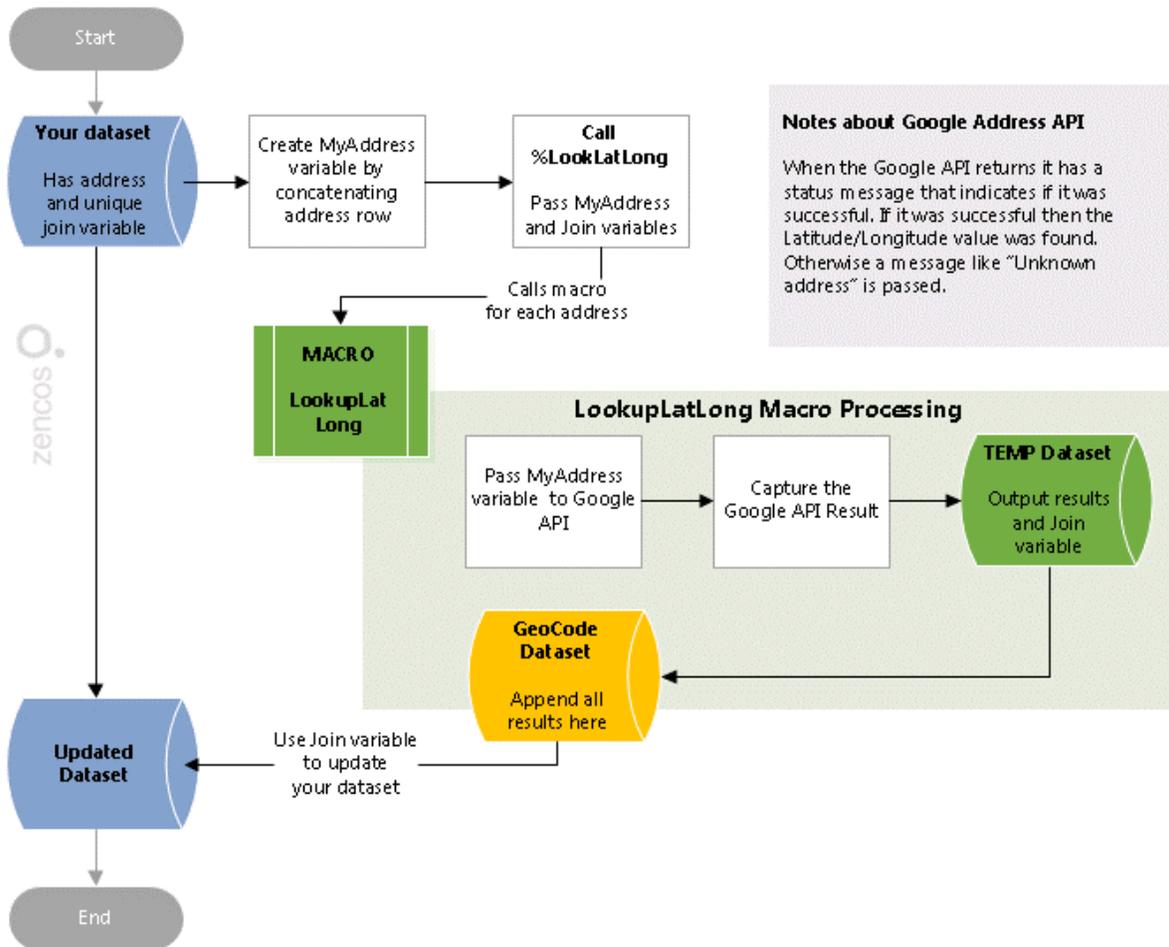
Display 18. Lookup Address Flowchart

17

USING THE LOOKUPLATLONG MACRO

```sas
/*STEP 1: Remove the existing address geocode if this is 2nd run*/
proc delete data=work.lkup_geocode; run;


/*STEP 2: This macro contacts Google API and then looks up the address */
/*It creates a dataset called: work.lkup_geocode    */


%macro LookupLatLong(
        in_addy    /*lookup street address*/
        ,yourJoinVar   /*This is the variable used to join to the main table*/
);


/* ========== Insert your Google API Key ================*/
/*Reference
https://developers.google.com/maps/documentation/geocoding/start*/


%let APIKEY=XXXXXXXXXXXXXXXXXXXXXXXXXXXXX;


/*Correct any punctuation issues within addresses to prevent issues */
%let fix_addy=%SYSFUNC(TRANSLATE(&in_addy.,%STR(,),%STR(*) ) );

data WORK.TEMP;
  INFILE
      "https://maps.googleapis.com/maps/api/geocode/xml?%NRSTR(&)address=%NRQ
      UOTE(&fix_addy.)
      ,%NRSTR(&)key=&APIKEY."
      URL   end=eof truncover debug;

  input;
  retain status xlat xlong address;

  format yourJoinVar 8. xlat xlong 12.8 status $20. address $50.;

  if index(_infile_,'status') then status= (scan(_infile_,3,'<>'));
  if index(_infile_,'formatted_address')
           then address= (scan(_infile_,3,'<>'));

  if status = 'OK' then do;
      if index(_infile_,'lat') then xLat=scan(_infile_,3,'<>');
      if index(_infile_,'lng') then xLong=scan(_infile_,3,'<>');
  end;

if EOF then do;
      yourJoinVar=&yourJoinVar.;
       output;
end;
end;
run;
```

18

```
/*Add latest address to the larger dataset */
proc append
    base=WORK.LKUP_GEOCODE
    data=WORK.TEMP
    force nowarn;
run;



/*clean up to prevent duplicates */
proc delete data=WORK.TEMP; run;
%let fix_addy=;
%mend;


%global xlat xlong status outaddress;

/*STEP 3 - call your source dataset
    You may need to change the address variables listed below*/

/*Build Address from dataset row */

data _null_;
set YourDataWithAddressesGoesHere;



/*Remove the punctuation within street address to reduce issues */
/*Replace with your variable */
addressline1=compress(addressline1,'', 'p');

myAddress=catx(
      translate(strip(addressline1),'+',' '),'*+',
      translate(strip(city),'+',' '),'*+',
      translate(strip(state),'+',' '),'*+',
      translate(strip(country),'+',' '),'*+',
      translate(strip(postalCode),'+',' ')
      );

/*Loop through the data to do the lookup for each address */
CALL EXECUTE( '%LookupLatLong('||%NRBQUOTE(myAddress)||','||YourJoinVAR||');'
);

run;
```