

Visualize the Geography of your Business Insights using SAS® MAP tools

Xingrong(Candice) Zhang, Alliance Data Card Services, Columbus, OH

ABSTRACT

Illustrating business insights on maps do not only give analysts easier ways to show different findings across different regions but also give business leaders visual revelations of critical business insights to aid their planning and decision making. The GMAP procedure in SAS® allows users to show data geographically. This paper will demonstrate how to present business insights using PROC GMAP with real life examples and show how additional map features can be added to SAS maps to make it visually stimulating by implementing annotated datasets. Sample code and a macro for map cosmetics will be provided.

INTRODUCTION

Regional disparity is one of interests for analytical work across many industries. For example, a medical institution may want to know the geographical distribution of people with a certain disease to support a pharmaceutical research, or, a marketing company may need to know which states have increasing/decreasing sales to better allocate campaign investment. As an analyst, it can be very easy to pull numbers for these kinds of questions. What is more challenging is finding the best way to show the numbers concisely and strikingly. One solution is to use the GMAP procedure and create the map you like.

This paper will showcase some geo-maps as examples of visualizing statistics by location. We will briefly go over some basics about GMAP procedure, including data source for map tiles, different map styles and how to use annotate datasets. To make it easier for beginners to get hands on experience, two SAS macros will be provided to create maps used in the examples. We will explain how to prepare your datasets and do the set ups for parameters in the macros.

With the utilization of annotate datasets, it is very flexible to decorate your SAS maps. Once you know the principle of creating annotate data and pulling map tiles, you'll be able to plot your own maps to suit specific business demand. The intention of this paper is to give new GMAP users an introduction of how we can produce self-defined maps in SAS® and bring some potential business applications of geo-maps.

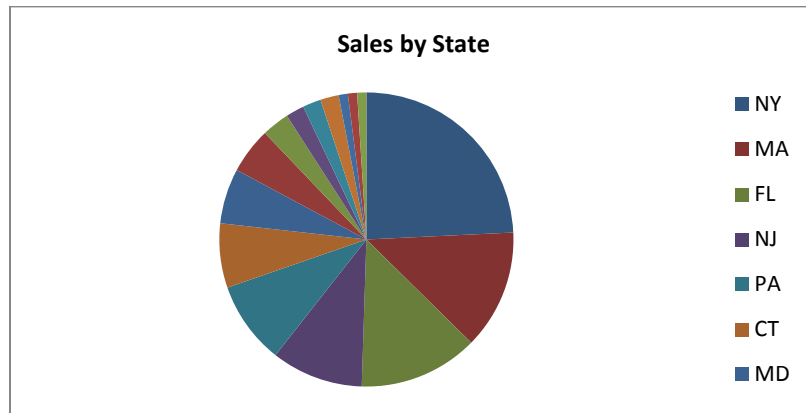
REAL LIFE EXAMPLES

EXAMPLE 1 - SALES DISTRIBUTION IN THE UNITED STATES

We'll take retail industry as example. A retail service company has recently acquired a new client A, who has stores in multiple states. Thus one of the retail service company's interests is to understand how Client A's business is going in each area. Suppose we have pulled sales and number of stores by state for client A. If we show the information using a table as below, it could be very overwhelming.

State	Sales	Percentage of Total Sales	Number of Stores
NY	\$191,091,450	24%	42
MA	\$104,733,644	13%	24
FL	\$99,667,148	13%	31
NJ	\$79,863,918	10%	22
PA	\$68,084,407	9%	15
CT	\$55,323,695	7%	12
MD	\$44,132,371	6%	10
VA	\$37,192,681	5%	11
...
AK	\$200	0%	0

If we create a pie chart, it becomes easier to capture which states have high or low sales.



How about plotting it on a U.S. map? We don't need to picture the location of the states in our mind. Instead, we can see it visually in Figure 1. Now we can see that the stores are heavily located in east coast, where more than 90% of sales come from. The map is colored by the number of stores in each state. The darker the color is, the more stores there are in that state. From this graph, we can clearly see that Client A's business is mostly located on the east coast. We are only showing percentage of sales by states. However, users can modify the SAS macro to display actual sales amount on the map if desired.

When we visualize this kind of geographical data points on a map, it can be very easy for the audience to absorb and quickly relate to the information. Also, it provides an efficient way for analyst to deliver findings.

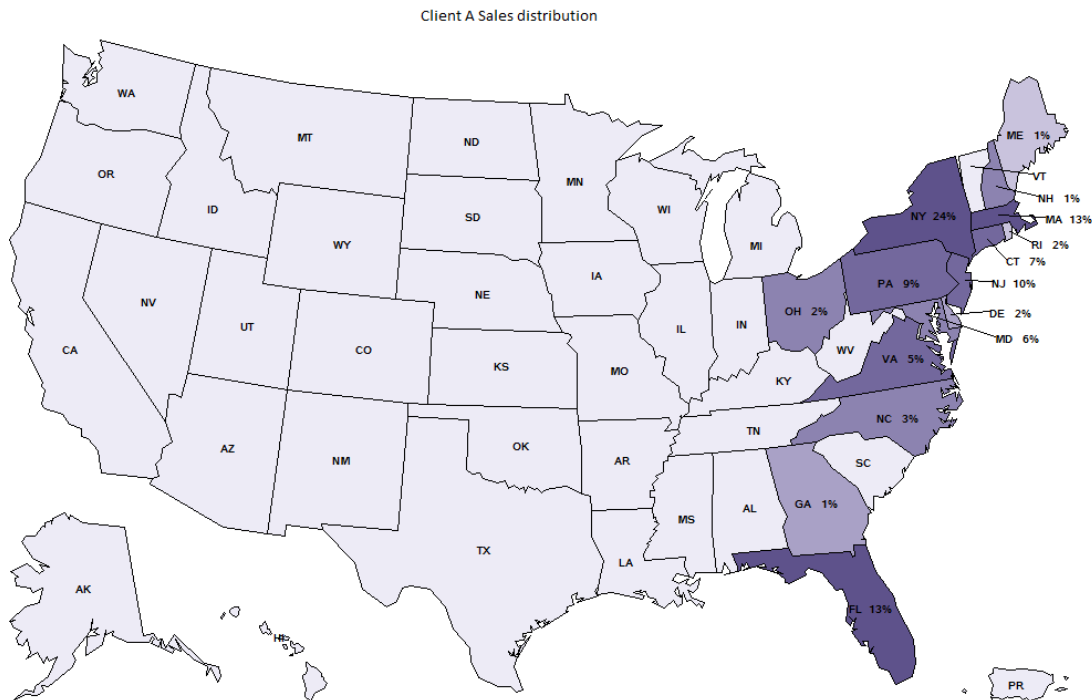


Figure 1. Sales Distribution and Store Locations.

EXAMPLE 2 - LOCATING BUSINESS OPPORTUNITIES

In this example, we are looking at a retailer C, who currently has 1,000 physical stores across the United States. With the hyper growth of their business, they are considering opening 10 new stores within the next year. Then the question comes: where should they open new stores? To answer this question, the analytics team gets the location of their current stores and the location of high value customers, to see if there are some locations with high value customers but no stores.

The result is shown as Figure 2. The red stars are current stores, while the green dots are high value customers defined as customers who meet a certain criteria. Both symbols are drawn by the zip code of the location. The green dots are zip code with more than 3,000 high value customers. You can easily observe that the stores are heavily located in the upper east coast, California, Florida, Texas and Chicago area. All these places have big cities and high density of high value customers. But there are still some places where there are considerable amount of high value customers but no stores nearby to support these customers. Some examples are Idaho, north Kansas, and west Ohio.

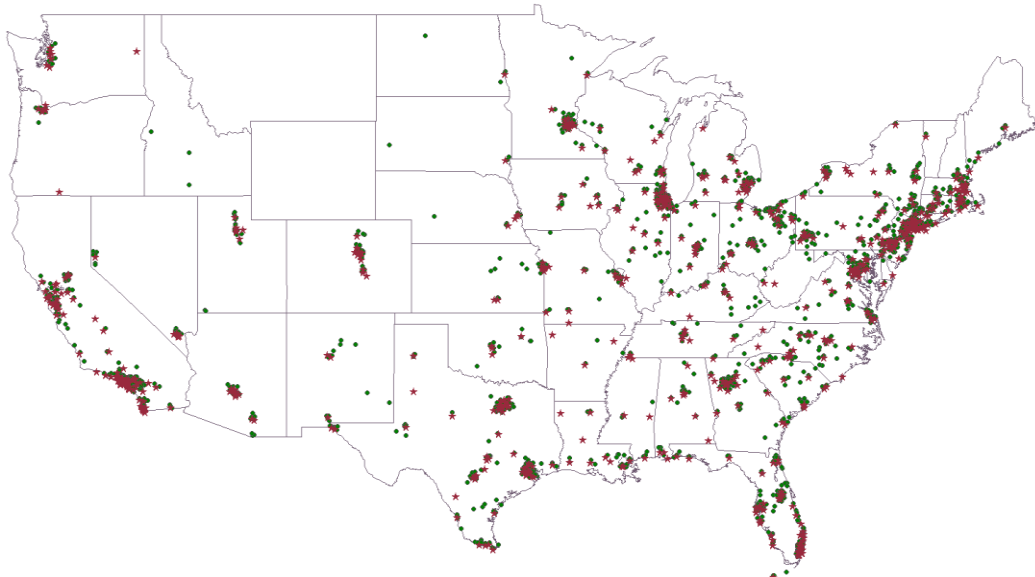


Figure 2. Stores and High Value Customers

We can even dive deeper into one state and get an even more detailed view. The store locations and high value customer distribution is shown in Figure 3. County names are labeled on the map. Lorain could probably be a good location to open a new store since there are substantial amount of high value customers in this county but no store yet.

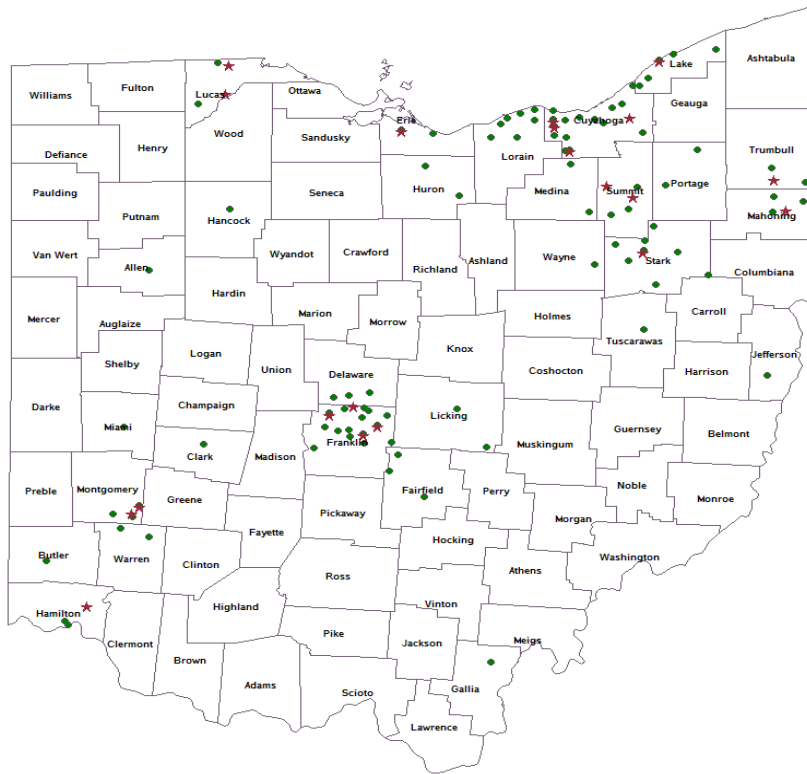


Figure 3. Stores and High Value Customers in Ohio

THE BASICS ABOUT GENERATING MAPS IN SAS®

The above examples show how geo-maps can help in creating business reports that are easy to understand. With GMAP procedure in SAS®, we can actually plot many types of maps and add annotations as desired. Here are the key things you need to know to become an expert on SAS maps.

TYPE OF MAPS

Typically, you can plot four types of maps in SAS®: Block Map, Choropleth Map, Prism Map and Surface Map. The difference in these maps lies in the way to visualize the response variable. As long as you have the idea of what to show on the map, you can choose any type based on your need or preference.

The first type is the block map, where response variables are presented by bars with height relative to response value. In the block map example in Figure 4, the response variable is population in each country. The larger the population is, the higher the bar is for that country.

The second type of map is the choropleth map. The two examples given earlier are all plotted from choropleth map. In choropleth maps, response variables are represented by different colors or patterns. A simple choropleth map example is shown in upper right of Figure 4. Population of each country in Europe is represented by colors. The darker the color is, the larger the population.

Prism maps visualize response variable by polyhedrons with different height, pattern and color. The prism map example is showing population in Africa. The polyhedrons are higher and the color is darker for the countries that have higher population.

Surface map shows response variables by spikes, heights of which are relative to magnitude of response value.

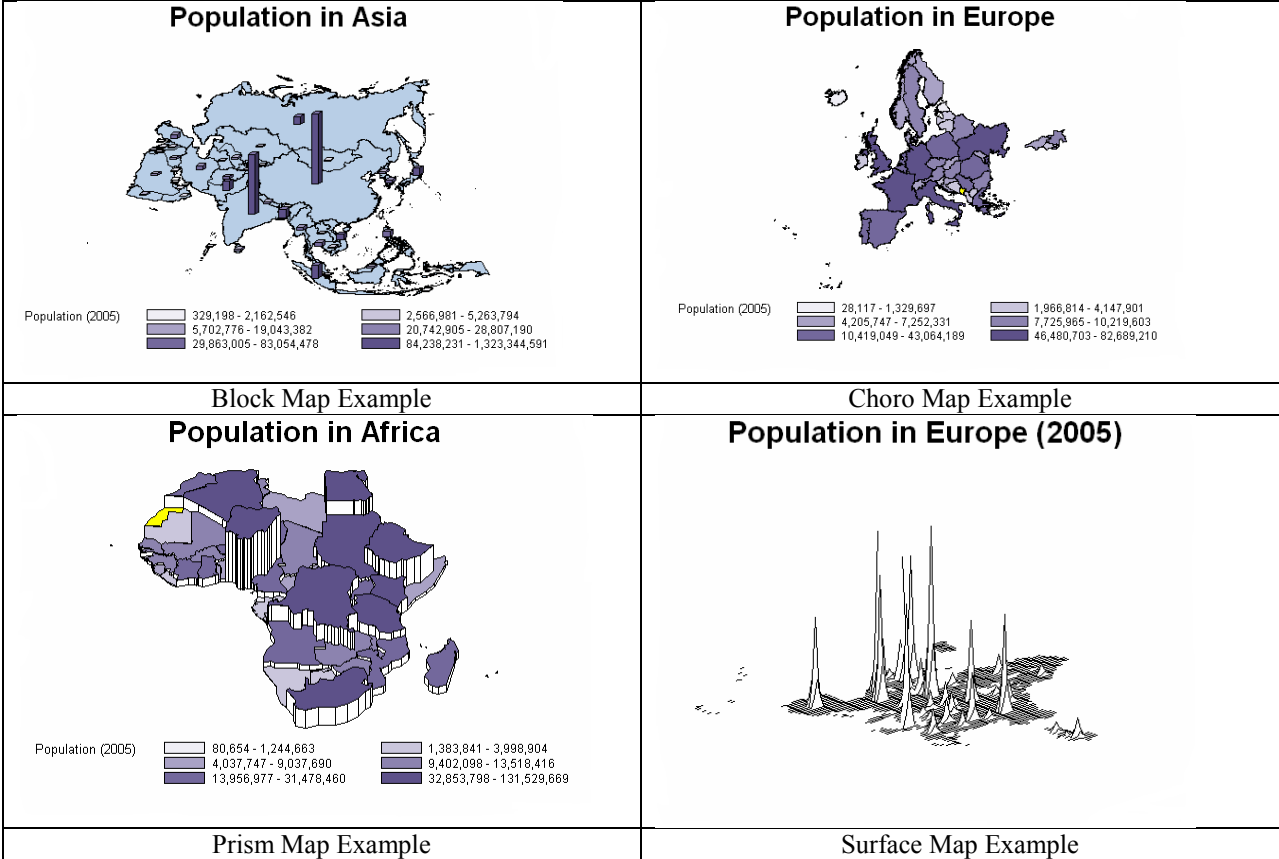


Figure 4: Map Types in Proc GMAP Source:

<https://support.sas.com/documentation/cdl/en/graphref/63022/HTML/default/viewer.htm#overview-gmap.htm>

GETTING SPATIAL INFORMATION

Now you may have some idea of which type to use and be ready to plot your geographic map. It's time to understand how a SAS map is generated. Simply speaking, there are map data libraries called Maps and Mapsgfk in SAS®. In these libraries, you can find map data that defines continents, countries, states, counties and similar area. With this spatial information, map tiles can be put together in the correct way to be demonstrated as geographic map.

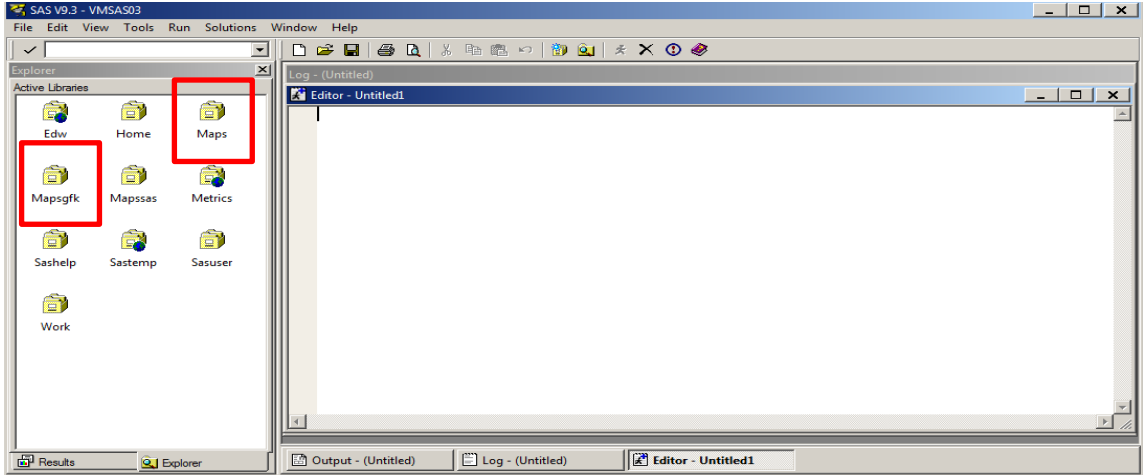


Figure 5. Map Data Library

You can use either Maps or Mapsgfk library. The Maps library comes in place first. However, Mapsgfk has some advantage over the other one. Spatial information in Mapsgfk comes from a single source, making it more accurate and updatable. Eventually, Mapsgfk will replace Maps.

LINKING YOUR DATA WITH MAP TILES

When you have location-based data, you can use it to define the geo map area. The most important thing you need to be concerned about is using the right spatial data that you can link your geo data with. In either Maps or Mapsgfk library, you can find map data at different levels. The rule here is you need to choose a map data that has same level information as your data. For example, if your data is sales in different states in USA, you would need to use spatial information for USA on state level. You can link your data easily with mapsgfk.us by state to pull map information about USA at state level.

Also, you can use your data to color the area of importance, as in example 1 where states with physical stores are colored in purple. This can be done by using choro statement in Proc GMAP. For the example here, you can put variable that represents number of stores in choro statement.

ANNOTATING THE MAP

Proc GMAP can take user defined annotate dataset to cosmetic the map. By defining things like position, function, style, text... in a data step, users can create the annotate dataset as demanded. It can take a whole paper to explain how to create annotate dataset. We will not discuss it in details in this paper. However, you can find some ready to use annotate macros on support.sas.com. The link below gives a summary of these macros.

<https://support.sas.com/documentation/cdl/en/graphref/63022/HTML/default/viewer.htm#annodict-using-macro.htm>

READY TO USE MACROS

We provide macros to generate maps in the two examples shown in this paper in the appendix. User instruction is listed below.

MACRO 1: SHOW STATISTICS FOR EACH STATE IN THE UNITED STATES

This macro generates choropleth map of USA on state level. It has options to color states by one statistics, and to show at most three other statistics at the center of each state.

To prepare data for first macro, users need to have state names, numbers and text to show (if any) in the dataset. State name is the only variable required, and others are optional. One example of user prepared data is in Figure 6. Several call outs about the user data:

1. State name is required, and need to be renamed so that its variable name is not exactly "STATE". In the example below, the actual variable name is "mystate". Its label is STATE and that's why it still shows STATE in the explorer.
2. If you want to put numbers as text on the map, you need to change them to character variables. In the example below, both sales and percentage of sales are changed to characteristic variables, stored as char_sales and char_perc.

VIEWTABLE: Work.Mydata							
	STATE	count	Total sales	pct off us sales	Frequency Count	char_sales	char_perc
1	NY	54729	191091449.63	0.6481469732	42	\$191091450	65%
2	MA	23884	104733643.59	0.6811769978	24	\$104733644	68%

Figure 6. User Data Example for Macro 1

If you are using Maps library, please use this macro:

```
%state_info(dsn=mydata,state=mystate,color_var=count,text1=mystate,text2=char_sales,text3=char_perc,title='test
5',num_text=3,size=0.8,max_length_text=10);
```

If you are using Mapsgfk library, please use this macro:

```
%state_info_gfk(dsn=mydata,state=mystate,color_var=count,text1=mystate,text2=char_sales,text3=char_perc,title='test
5',num_text=3,size=0.8,max_length_text=10);
```

The settings for the two macros are the same. **Dsn** is your data that contains state, numbers or texts you like to label on the map. **State** is the variable name of state in your dataset. **Color_var** is the variable you would like to use to color each state. **Text1** to **text3** are three variables that you would like to label on each state. You can only input **text1** and omit **text2** and **text 3** to show only one variable on each state. Similarly, if you omit **text1**, **text2** and **text3**, then there will be no label on each state. **Title** is the head of the map. **Num_Text** is number of variables you are using as label for each state. It should be consistent with number of **texti** (i=1,2,3) you are using. **Size** is the size of label. If you are showing three labels on each state, you will probably want to change size to smaller number to avoid overlap of the texts. **Max_length_text** is the maximum length of text variables used as label.

MACRO2: DROP STARS AND DOTS BY ZIP CODE ON THE U.S. MAP

This macro draws stars and dots on the U.S. map. To run the macro, users need two sets of zip codes: one to pin with red stars and the other to pin with green dots. If you only want to have some zip codes pinned with dots, it is OK to omit the settings for stars, and vice versa. Just keep in mind that your data should contain geocoded latitude and longitude in degrees. For example, if you have a dataset called **myzip** with only one column – zip codes, you can use the following procedure to retrieve latitude and longitude which are saved in output dataset **gmyzip**.

```
Proc geocode
  method=zip
  data=myzip
  out=gmyzip;
run;
```

The macro line is as below:

```
%points_by_zip(dotdsn=gmyzip2,stardsn=gmyzip,dotsize=0.4,starsize=1.3);
```

Dotdsn is the dataset that contains zip code to be drawn as green dots. **Stardsn** is the dataset that contains zip code to be drawn as red stars. **Dotsize** is the size of dots and **starsize** is the size of stars.

CONCLUSION

This paper presents examples of creating geographic map in SAS® and illustrates the flexibility to decorate the map by using annotates. Macros are provided so that audience can produce the maps in the example using their own data easily. Also, we introduce the basics of plotting maps in SAS® to educate new users on functionality of GMAP procedure. With the help of the GMAP procedure and the macro provided here, you are set to visualize the geography of your business.

REFERENCE

- [1] “Google-like maps in SAS.” SAS Presentations at SAS Global Forum 2012, Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/rnd/papers>
- [2] “Using Annotate Macros.” SAS support document. Available at <https://support.sas.com/documentation/cdl/en/graphref/63022/HTML/default/viewer.htm#annodict-using-macro.htm>
- [3] “Differences between GFK and Traditional Map Data Sets.” SAS support document. Available at <https://support.sas.com/documentation/cdl/en/graphref/65389/HTML/default/viewer.htm#p03gkwkwlxhv5dn1drl3z922qzxd.htm>

ACKNOWLEDGMENTS

I would like to thank Delali Agbenyegah for encouraging me to write this paper. Also, I appreciate the entire Predictive Analytics team at Alliance Data Card Services for their support.

CONTACT INFORMATION

All comments and questions are encouraged and appreciated. Contact the author at:

Candice Zhang
Alliance Data Card Services
3100 Easton Square Place
Columbus, OH, 43219
candice.zhang@alliancedata.com

APPENDIX I: MACRO 1 (SALES DISTRIBUTION IN THE UNITED STATES)

Version using Maps library

```
%macro
state_info(dsn=,state=,color_var=,text1=,text2=,text3=,title=,num_text=,size=,max_length_text=);
  data uscenter;
    set maps.uscenter(drop=long lat);
    state2=fipstate(state);
  run;
  proc sort data=uscenter;by state2;run;
  proc sort data=&dsn. out=temp;by &state.;run;
  data temp;
    set temp;
    state2 = &state.;
  run;
  data mycenter;
    merge temp(in=a) uscenter(in=b);
    by state2;
    if a and b;
  run;
  %if &num_text=1 %then %do;
  data center1;
    length function $ 8;
    retain flag 0 xsys ysys '2' hsys '3' when 'a';
    set mycenter;
    function='label';style = "'Albany
AMT/bold'";text=&text1.;size=&size.;position='2';
    if ocean='Y' then do;
      position='6';
      output;
      function='move';
      flag=1;
    end;
    else if flag=1 then do;
      function='draw';
      size=.25;
      flag=0;
    end;
    output;
  run;
  data center;
    length text $ &max_length_text.;
    set center1;
  run;
```



```

%end;
%else %if &num_text=2 %then %do;
    data center1;
    length function $ 8;
    retain flag 0 xsys ysys '2' hsys '3' when 'a';
    set mycenter;
    function='label';style = "'Albany
AMT/bold'";text=&text1.;size=&size.;position='2';
    if ocean='Y' then do;
        position='3';
        output;
        function='move';
        flag=1;
    end;
    else if flag=1 then do;
        function='draw';
        size=.25;
        flag=0;
    end;
    output;
run;
    data center2;
    length function $ 8;
    retain flag 0 xsys ysys '2' hsys '3' when 'a';
    set mycenter;
    function='label';style = "'Albany
AMT/bold'";text=&text2.;size=&size.;position='5';
    if ocean='Y' then do;
        position='6';
        output;
        function='move';
        flag=1;
    end;
    else if flag=1 then do;
        function='draw';
        size=.25;
        flag=0;
    end;
    output;
run;
    data center;
    length text $ &max_length_text.;
    set center1 center2;
run;
%end;
%else %if &num_text=3 %then %do;
    data center1;
    length function $ 8;
    retain flag 0 xsys ysys '2' hsys '3' when 'a';
    set mycenter;
    function='label';style = "'Albany
AMT/bold'";text=&text1.;size=&size.;position='2';
    if ocean='Y' then do;
        position='3';
        output;
        function='move';
        flag=1;
    end;
    else if flag=1 then do;
        function='draw';
        size=.25;
        flag=0;
    end;
end;

```

```

        output;
run;
    data center2;
    length function $ 8;
    retain flag 0 xsys ysys '2' hsyst '3' when 'a';
    set mycenter;
    function='label';style = "'Albany
AMT/bold'";text=&text2.;size=&size.;position='5';
    if ocean='Y' then do;
        position='6';
        output;
        function='move';
        flag=1;
    end;
    else if flag=1 then do;
        function='draw';
        size=.25;
        flag=0;
    end;
    output;
run;
    data center3;
    length function $ 8;
    retain flag 0 xsyst ysys '2' hsyst '3' when 'a';
    set mycenter;
    function='label';style = "'Albany
AMT/bold'";text=&text3.;size=&size.;position='8';
    if ocean='Y' then do;
        position='9';
        output;
        function='move';
        flag=1;
    end;
    else if flag=1 then do;
        function='draw';
        size=.25;
        flag=0;
    end;
    output;
run;
    data center;
    length text $ &max_length_text.;
    set center1 center2 center3;
run;
%end;
data plotdata;
    set &dsn.;
    rename &state. = statecode;
run;
    proc gmap map=maps.us data=plotdata all;
        title &title.;
        id statecode;
        choro &color_var. /anno=center;
    run;
%mend state_info;

```

Version using Mapsgfk library:

```

%macro
state_info_gfk(dsn=,state=,color_var=,text1=,text2=,text3=,title=,num_text=,size=,max_
length_text=);

```

```

data uscenter;
    set mapsgfk.uscenter(drop=long lat);
run;
proc sort data=uscenter;by statecode;run;
proc sort data=&dsn. out=temp;by &state.;run;
data temp;
    set temp;
        statecode = &state.;
run;
data mycenter;
    merge temp(in=a) uscenter(in=b);
        by statecode;
        if a and b;
run;
    %if &num_text=1 %then %do;
        data center1;
            length function $ 8;
            retain flag 0 xsys ysys '2' hsys '3' when 'a';
            set mycenter;
            function='label';style = "'Albany
AMT/bold'";text=&text1.;size=&size.;position='2';
            if ocean='Y' then do;
                position='6';
                output;
                function='move';
                flag=1;
            end;
            else if flag=1 then do;
                function='draw';
                size=.25;
                flag=0;
            end;
            output;
        run;
        data center;
            length text $ &max_length_text.;
            set center1;
        run;
    %end;
    %else %if &num_text=2 %then %do;
        data center1;
            length function $ 8;
            retain flag 0 xsys ysys '2' hsys '3' when 'a';
            set mycenter;
            function='label';style = "'Albany
AMT/bold'";text=&text1.;size=&size.;position='2';
            if ocean='Y' then do;
                position='3';
                output;
                function='move';
                flag=1;
            end;
            else if flag=1 then do;
                function='draw';
                size=.25;
                flag=0;
            end;
            output;
        run;
        data center2;
            length function $ 8;
            retain flag 0 xsys ysys '2' hsys '3' when 'a';
            set mycenter;

```

```

        function='label';style = "'Albany
AMT/bold'";text=&text2.;size=&size.;position='5';
        if ocean='Y' then do;
            position='6';
            output;
            function='move';
            flag=1;
        end;
        else if flag=1 then do;
            function='draw';
            size=.25;
            flag=0;
        end;
        output;
run;
data center;
length text $ &max_length_text.;
set center1 center2;
run;
%end;
%else %if &num_text=3 %then %do;
data center1;
length function $ 8;
retain flag 0 xsys ysys '2' hsys '3' when 'a';
set mycenter;
function='label';style = "'Albany
AMT/bold'";text=&text1.;size=&size.;position='2';
        if ocean='Y' then do;
            position='3';
            output;
            function='move';
            flag=1;
        end;
        else if flag=1 then do;
            function='draw';
            size=.25;
            flag=0;
        end;
        output;
run;
data center2;
length function $ 8;
retain flag 0 xsys ysys '2' hsys '3' when 'a';
set mycenter;
function='label';style = "'Albany
AMT/bold'";text=&text2.;size=&size.;position='5';
        if ocean='Y' then do;
            position='6';
            output;
            function='move';
            flag=1;
        end;
        else if flag=1 then do;
            function='draw';
            size=.25;
            flag=0;
        end;
        output;
run;
data center3;
length function $ 8;
retain flag 0 xsys ysys '2' hsys '3' when 'a';
set mycenter;

```

```

        function='label';style = "'Albany
AMT/bold'";text=&text3.;size=&size.;position='8';
        if ocean='Y' then do;
            position='9';
            output;
            function='move';
            flag=1;
        end;
        else if flag=1 then do;
            function='draw';
            size=.25;
            flag=0;
        end;
        output;
    run;
    data center;
    length text $ &max_length_text.;
    set center1 center2 center3;
run;
%end;
data plotdata;
    set &dsn.;
    rename &state. = statecode;
run;
proc gmap map=mapsgfk.us data=plotdata all;
    title &title.;
    id statecode;
    choro &color_var. /anno=center;
run;
%mend state_info_gfk;

```

Appendix II: Macro 2 (Locating Business Opportunities)

```

%macro points_by_zip(dotdsn=, stardsn=, dotsize=, starsize=);
/*annotate data for star*/
data anno;
    set &stardsn.;
    retain xsys ysys '2' function 'label' flag 1 when 'a';
    size=&starsize.;
    style='special';
    /* The symbol is a star. */
    text='M';
    /* Specify the color for the symbol. */
    color='depk';
    output;
run;
/*annotate data for dot*/
data anno2;
    length function color $ 8 style $20 ;
    retain xsys ysys '2' hsys '3' when 'a';
    set &dotdsn.;
    /* All annotate points are 360-degree pies. */
    size=&dotsize.;
    function='pie';
    rotate=360;
    style='psolid';
    color='GREEN';
    output;
    /* Draw an outline around each 'dot' */
    color="gray55"; style='pempty';
    output;

```

```

run;
/*combine two annotate datasets and project x and y*/
data myanno;
    set anno2 anno;
run;
data panno;
    set myanno;
    x= x * 20037508.34 / 180;
    y= log(tan((90+y) * constant('pi')/360)) /
        (constant('pi') / 180);
    y= y * 20037508.34 / 180;
run;
/*Get map data from mapsgfk, and project x and y*/
data mymap;
    set mapsgfk.us_states(where=(statecode not in ('AK','HI')));
run;
data pmap;
    set mymap;
    x=long;y=lat;
    x= x * 20037508.34 / 180;
    y= log(tan((90+y) * constant('pi')/360)) /
        (constant('pi') / 180);
    y= y * 20037508.34 / 180;
run;
/*draw the map*/
pattern1 v=me c=grp r=50;
proc gmap data=pmap map=pmap;
    id state;
    choro state / anno=panno nolegend;
run;
quit;
pattern;
%mend points_by_zip;

```