Personalizing Mailings and Driving Sales Using SAS/GRAPH®

James P. Shields, IDEXX Laboratories, Westbrook, Maine

Abstract

Personalizing marketing messages to the individual customer is an effective method for increasing response and driving sales. This paper will walk through using SAS/GRAPH to create personalized maps showing the incidences of Lyme in relation to where a veterinary clinic is located. The presentation will cover all the steps that were involved in building an actual direct marketing promotion. Included will be the use of PROC GMAP, Annotate, a touch of trig, changing graph resolution, and formatting the output to better suit the printers needs.

Introduction

On occasion SAS® creates a unexpected opportunity. This happened to me while analyzing data for one of our product managers. I was using SAS/GRAPH to analyze the prevalence of Lyme disease in the United States. The data I was using was the number of cases reported by veterinarians at the zip code level. For the analysis I was using SAS/GRAPH in combination with ODS to generate maps of Lyme disease at the county level. The maps were well received by the product manager and he thought that it would be great if we could get this information to the veterinarians so they could see the prevalence of Lyme disease in their area. The idea of the mailer was born, but there was so much more opportunity we could open by using SAS.

Getting Personal

A mailer would work even better if we customize it down to the veterinarian level. Could we generate a separate map for each clinic? Could we show them among the Lyme disease incidences they were located? Could we make an individual map for each Veterinarian clinic? What should I personalize it with? A dot for the clinic? A dot for each Lyme case?

To get some ideas of what I could add to a map to personalize it I used two trusted sources for information. The first source is to do a search on the SAS website under the “Support and Training” section. Typing in “SAS/GRAPH Examples” returns a selection at the top of the list called “SAS/GRAPH Example Programs.” Under this tab are several GMAP examples that gave me ideas on what I may want. The one I focused on for this graph is a map with circles called “Texas Dealer Service Areas.” The ideas of circles generated some interest for me. What if we used circle within circles to draw the veterinarian to his location? Perhaps a red target pattern would work and indicate a sense of urgency.

Creating the First Target

Generating separate maps for each clinic would be fairly easy in SAS. SAS can do a loop for each clinic, plot the map, save the results, and move onto the next map. To do this we will need to write the mapping code as a macro. When doing macro programming I generate one map for one clinic then turn “macroize” it.

The nice part of the example program from SAS is the ability to alter the code to fit my needs. The one thing I needed right away is the ability to identify a single clinic on a map. For each clinic we have a postal code and SAS provides latitude and longitude for each zip code. This zip code latitude and longitude can also be downloaded from the SAS website. Or it can be purchased through GIS vendors if you need a more current file.

I picked one clinic and started working on the presentation. Using the Annotate Facility and a special font I was able to put cross hairs over a clinic in Maine. The map is on the next page.
The annotate facility lets a SAS user take their standard graphical output, in this case from SAS/GRAPH, and add lines, circles, dots, text, or any shape that you can figure out the trigonometry needed to draw. The Annotate Facility allows the user to supply PROC GMAP with an annotate dataset telling SAS what to do when drawing the graph. For the above graph the annotate dataset is below.

<table>
<thead>
<tr>
<th>Y</th>
<th>X</th>
<th>xsys</th>
<th>ysys</th>
<th>when</th>
<th>text</th>
<th>color</th>
<th>function</th>
<th>style</th>
<th>Position</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.751731</td>
<td>1.562282</td>
<td>2</td>
<td>2</td>
<td>A</td>
<td>D</td>
<td>Red</td>
<td>label</td>
<td>special</td>
<td>+</td>
<td>1</td>
</tr>
</tbody>
</table>

Xsys and Ysys indicates what coordinate system to use. Function, label, style, position, size, and text all combine to place a red label of font “special” character “d” (cross hairs) of size 1 centered (+) at the latitude and longitude indicated by X and Y. Since when = “a” this is done after the map of Maine is drawn so the alterations appear on top of the map.

However, the annotate dataset will not work if we do not change the latitude and longitude to the coordinate system recognized by GMAP. To do this we need to combine the annotate dataset with the Maine dataset as in the code below. “Anno” is the dataset shown above and maps.county is the county level map data set supplied by SAS. The combined dataset “combo” is projected using PROC GPROJECT. The resulting dataset is then split apart using a flag not shown on the “Anno” dataset above. The projected annotate dataset is then input into PROC GMAP as anno= and the map dataset is fed into map=. The dataset ‘state_county’ is an aggregate of the county level lyme information from the zip level data set lyme_zip.
**Adding some Bells and Whistles**

So we have a cross hair on a map of Maine. This certainly will not attract the attention of Veterinarians. To create our target we need to add some more circles. Available from the example map from SAS called “Texas Dealer Service Areas” is the code defining distance to form a circle. The code was used as a base to build the circles.

```sas
data anno;
  set Zip_lat_long_lyme_cust;
  retain xsys ysys '2' flag 1 when 'a';
  length text $25 color function style $8;
  drop xold yold;
  d2r=3.1415926/180;
  r=3958.739565;
  SRVAREA = 30;
  xcen=abs(x);
  ycen=y;
  do degree=0 to 360 by 0.1; /* thirty mile radius */
    if degree=0 then do;
      function='point';
      line=1;
      style='msolid';
      color='red';
      size=2;
    end;

Evaluation notes were added to the output document. To get rid of these notes, please order your copy of ePrint IV now.
end;
else do;
  function='draw';
  color='red';
  size=1;
end;

y=arcsin(cos(degree*d2r)*sin(srvarea/R)*cos(ycen*d2r)+
cos(srvarea/R)*sin(ycen*d2r))/d2r;
x=xcen+arcsin(sin(degree*d2r)*sin(srvarea/R)/cos(y*d2r))/d2r;

x=atan(1)/45*abs(x);
y=atan(1)/45*y;
output;
end;
.
.
.

The biggest change to the anno dataset is from the “do degree=0 to 360 by 0.1” loop. This starts at degree 0, draws a point x, y representing 30 miles from the center of the circle. The next time through the loop it moves 0.1 degrees from the previous observation and connects the two points. It continues to output a new record each time until it completes 360 degrees. Now the “Anno” dataset looks like the following:

<table>
<thead>
<tr>
<th>Y</th>
<th>X</th>
<th>xsys</th>
<th>ysys</th>
<th>flag</th>
<th>when</th>
<th>text</th>
<th>color</th>
<th>function</th>
<th>Style</th>
<th>degree</th>
<th>line</th>
<th>size</th>
<th>position</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.759309</td>
<td>1.562282</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>a</td>
<td>red</td>
<td>point</td>
<td>msolid</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.759309</td>
<td>1.5623</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>a</td>
<td>red</td>
<td>draw</td>
<td>msolid</td>
<td>0.1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.759309</td>
<td>1.562318</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>a</td>
<td>red</td>
<td>draw</td>
<td>msolid</td>
<td>0.2</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.759309</td>
<td>1.562337</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>a</td>
<td>red</td>
<td>draw</td>
<td>msolid</td>
<td>0.3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.751731</td>
<td>1.562282</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>A</td>
<td>red</td>
<td>label</td>
<td>special</td>
<td>360.1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>+</td>
</tr>
</tbody>
</table>

The last line lets SAS know to put the cross hair at the center of the circle.

Also note that ‘SRVAREA’ is the radius of the circle we want drawn, ‘r’ is the radius of the earth, and d2r is the pi/180 with is to convert the latitude and longitude from degrees to radians.

The meet of the calculation is the formula:

\[
y = \arcsin\left(\cos(degree*d2r) \times \sin(srvarea/R) \times \cos(ycen*d2r) + \cos(srvarea/R) \times \sin(ycen*d2r)\right) / d2r; \\
x = xcen + \arcsin\left(\sin(degree*d2r) \times \sin(srvarea/R) / \cos(y*d2r)\right) / d2r;
\]

This provides the x and y coordinates to draw the circles.

The capabilities of annotate to draw any shape that you specify the coordinates for unleashes a whole host of possibilities. Unfortunately it takes a solid knowledge of trigonometry, or a good source, to draw the more interesting shapes.

The result is on the next page, but it certainly will not impress the veterinarians as to the importance of testing for Lyme disease. Some real bells and whistles are needed.
By repeating the command from above for a radius of 60 miles we can add another circle with a radius of 60 miles. And the cross hairs provided by SAS as a special font do not look that smooth. We should replace this with our own cross hair. This can also be done by altering the previous code:

1) Draw a circle with a radius of 5 miles.
2) Point to 90 degrees using the point function.
3) Move the pointer to 270 degrees and use the draw function.
4) Repeat for 0 degrees and 180 degrees to add the vertical line.

The creative team also requested that information be added to the maps to make them more meaningful. On the creative piece the graphs will be placed on there will be numbers informing the veterinarian of the number of lyme cases within 60 miles and 30 miles of his clinic. So we would like to label the circles appropriately. Again this can be done using commands in the annotate data set:

1) Point to a point 0 degrees slightly farther than 30 miles with function='Label'.
2) Add the text ‘60 Miles’.
3) Do the same for the 30 miles circle.
Also added, to give a better reference for the veterinarians were the three top cities for each state. This data can be added to each state using maps.uscity provided with SAS. This also includes 1990 population size. Adding these to the annotate dataset we then have the following map.

Partial Circles

We now have a map that we can use, but we have partial circles when the location of the clinic is too close to the edge of the map. This is not esthetically pleasing and presents a problem in states with a high number of clinics near the edge (ex. Philadelphia area) or for small states (ex. Connecticut).

While looking through a map dataset I realized I could create a fake county on the map dataset. I would get enough room for all the circles if I took the maximum and minimum value of the latitude and longitude and used this to create the new ‘county’ drawn between the two coordinates.

```sas
data combo;
  set cities(where=( state=23)) anno
    maps.counties(where=( state=23));
run;
```
proc summary data=combo nway;
output out=means(drop=_type_ _freq_) min(x) = min_x
  min(y) = min_y
  max(x) = max_x
  max(y) = max_y
;
run;

data max(keep=x y temp);
  set means;
  y=min_y;
  x=min_x;
  output;
  y=max_y;
  x=max_x;
  output;
run;

data combo;
  set max
cities(where=( state=23))
anno
maps.counties(where=( state=23));
run;

The extreme latitude and longitude values in the annotate dataset are determined by using a PROC SUMMARY. These values are then put out to the dataset "max" and combined into the combo dataset for projection. After the projection using PROC GPROJECT is run the extreme latitude and longitude values are written out to the map dataset. There is essentially a new county with no width. However, there are still county lines as seen on the map on the next page.

To remove the line you can request that any county with missing data have a boundary the same color as the background using the CEMPTY= command. Where the valued assigned to this is the color of the background. In this case the color of the background is white. Back ground colors can be set by using the CBACK= option in the goption statement. It is important to note that none of other counties can have missing values for the number of Lyme cases. Setting all missing values to zero solves the problem.

gooptions CBACK='white'

proc gmap data=state_county map=state anno=label all;
id state county;
choro lyme / coutline=black nolegend levels=2 CEMPTY='white';
run;
quit;
Color Spaces, Resolution, and File Types

Now that we have a working map everything needs to be brought together with the creative team. Their first request was to increase the resolution of the output. They needed a much higher resolution to satisfy their design requirements. And the printer needs the file in TIFF format.

To set the resolution for a TIFF file you can specify the length and width of the graphic and the number of pixels used in the Goptions statement. $X_{\text{max}}$ = sets the length along the x-axis. $Y_{\text{max}}$ = sets the length along the y axis. Combine these with $\text{xpixels}$ = and $\text{ypixels}$ = can determine the dots per in (dpi) by the formula $\text{dpi} = \frac{\text{pixels}}{\text{length}}$.

In addition, we also need a location to write our file to. Combine a filename statement and the goptions statement as below to let SAS know where to place all your graphics.

```sas
filename output "C:\Lyme\Lyme_Maps\Map04101.tif";
goptions device=TIFFP ftext="Arial" CBACK='white'
gsfname=output gsfmode=replace xpixels=4200 ypixels=3000
    xmax=7 ymax=5;
```

Evaluation notes were added to the output document. To get rid of these notes, please order your copy of ePrint IV now.
Device = TIFFP specifies that the output will be a color tiff file. In this case the resolution is 4200/7=600 dpi by 3000/5=600 dpi. It is amazing what this does to the quality of a map. The higher resolution really improves the look.

The maps are not going to be sent out as stand alone products. They are going to be printed on a separate envelope for each clinic. What we need to do is match the color background of the map to the color of the envelope. For this mailing the envelopes were lime green. We tried several methods to do this. A person from creative would read colors from the program she was working in, I would convert it to CYMK format and then produce a map which hopefully would match. This is a futile effort. Because of variations in the ‘Color Space’ (the viewing medium) the colors would not match.

Fortunately, we brought the problem to the attention of the printer since they would be matching each TIFF file to the appropriate envelop. They let us know that we did not need to do any color matching. As long as the colors on TIFF files were different for each distinct area they could identify the colors we used and change them to the colors they needed. The lime on the tiff files didn’t even need to be lime and they would change it to the right lime. This was an easy solution and saved us a lot of effort.

Macroizing the Maps

Now that all the creative hurdles have been solved the maps need to be generated for all the clinics on our file. This could be formidable when dealing with millions of maps to be made or even 100,000 maps. It takes programming time to create even one map. Multiply this with the number of clinics and the program could be running for quite a while. However, maps do not need to be generated for each clinic, because the latitude and longitude is at the zip code level. We just need a unique map for each zip code containing a clinic. There are only 40,000 zip codes in the US. The number of zip codes with veterinary clinics is substantially less than this.

‘Macroizing’ the mapping program involves converting the program for one map to a program for multiple maps by substituting the clinic specific data with macro variables. For instance, the filename for the file output becomes

filename output "C:\Lyme\Lyme_Maps\Map&zip_code..tif";

where &zip_code is the macro variable for the zip code. A macro do loop recursively passes a zip code to the mapping program as the macro variable &zip_code. This then replaces the specific information used earlier and a separate map is generated.

In addition to the maps, the printer also needs a file that ties each map to the appropriate clinic. Since each map represent is for a zip code, the printer can read the zip code from the address for that clinic and grab the appropriate map. The map is then combined with the rest of the creative piece.

Conclusion

SAS/GRAFH with the Annotate Facility opens many possibilities for users. These maps were just the start in using personalized graphics to get the attention of our customers. Even though we had some hurdles to overcome, and some doubts, in the end everything fell into place. We now have the capability to personalize not just the information, but how the information is graphically presented.

Acknowledgements

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


SAS.com SAS/Graph Examples. SAS Institute Inc.
http://support.sas.com/techsup/sample/sample_graph.html
Contact Information

James Shields
IDEXX Laboratories, Inc.
1 IDEXX Drive
Westbrook, Maine 04092
James-Shields@idexx.com
207.856.8182

5,393 incidences of Lyme disease have been reported around Fleetwood.