Putting Yourself on the Map with the GMAP Procedure
Michael Davis, BlueCross BlueShield of Connecticut

ABSTRACT

Many users are fascinated by maps although they seem too complex to be produced by an average user. This paper introduces new SAS/GRAPH® users the concepts necessary to create maps. The paper offers guidance in finding those pesky coordinates required to put the GMAP procedure to work. The major features of PROC GMAP that are required to draw a geographic map are explored. The paper also discussed elements of good graphic design as they apply to maps, and presents tips on using graphic devices and drivers intelligently.

Part 1:
Introduction to SAS Maps

One of the best reasons to use SAS® rather than another fourth generation language is that resulting analysis can be produced in a format other than a tabular report. Your SAS program can produce a calendar, plot observations on an X-Y chart, summarize results on a pie chart, or even contrast values by map region. This variety of options allows you to format your analysis to matches the requirements of your audience.

Maps have a fascination for many people and were one of the reasons why our site invested in SAS/GRAPH (which you need to license in order to produce higher quality graphic images from SAS data sets). However, many persons do not realize that PROC GMAP can display any collection of polygons. As you may recall from high school geometry, a polygon is a closed geometric figure bounded by straight lines.

For example, one instructor at the SAS Institute enlivens his SAS/GRAPH class with a reproduction of a popular cartoon character. To produce the image, he merely enlarged a copy of the cartoon character on a photocopier, traced the polygons on a sheet of graph paper, and inputted the ends of each line segment (coordinates) into a SAS data set.

Besides cartoon characters, SAS can create maps that display floor layouts, corporate logos, and abstract designs. Because the data used to define the PROC GMAP output is a SAS data set, anything you can do with a SAS data set can be done with your map data. For example, if you want to stretch out a state outline lengthwise for effect, you merely multiply the horizontal (X) coordinates by a constant value.

However, first you have to find, calculate, or measure those pesky coordinates before putting PROC GMAP to work on your application. The next part of this paper will discuss the alternatives available for obtaining coordinates for geographic maps.

Part 2:
Where Do You Find Those Pesky Coordinates?

For the purpose of this paper, unless otherwise noted, coordinates refer to a series of latitudes and longitudes that describe the geographic areas to be shown.

If you have your coordinates as latitudes and longitudes, SAS can convert them to X-Y coordinates, which will be described in a future installment.

The easiest source to access for coordinates is the SAS map data sets that come with SAS/GRAPH. For example, if you want to map the 50 states or some United States counties, the coordinates are available to you at no extra charge. I recommend that you play with these coordinates to get the feel of the GMAP, GREDUCE, and GPROJECT procedures.

However, let's assume that you wanted to map a geographic area, such as zip code areas, that are not on the map data sets supplied by the SAS Institute. The lowest budget alternative is to do it yourself. You can use the "sheet of graph paper" approach described in the last installment or you can use a digitizer to generate X-Y coordinates.
If you go the digitizer route, SAS Institute Technical Report A-107 may help you. However, I did not have access to a digitizer and it seemed a better use of my time to purchase the coordinates I needed rather than to chart them out manually.

The U.S. Census Bureau has been charting the United States to aid in compiling the Census results. Their latest effort is called the TIGER/Lines files (Topologically Integrated Geographic Encoding and Referencing system). TIGER/Lines files show street intersections, rivers, and other landmarks.

If the TIGER/Lines files interest you, please see the paper presented at SUGI 15 by John Blodgett and John McIntyre. Recent SUGI proceedings have other papers that discuss other mapping techniques.

You can purchase the TIGER/Lines files that you need from the Census Bureau. However, these files and much more can be purchased from Geographic Data Technology, Inc. in Lyme, New Hampshire. They are a reputable company that has been in business since 1980. They know SAS and can guide you toward identifying and obtaining the information you need. Call them at (603) 795-2183 for further information.

I did that and found to my distress that to license the information I needed cost much more than my corporate budget permitted spending. Through some detective work, I located another reputable company named Claritas/NPDC Inc. (formerly National Planning Data Corporation), headquartered in Ithaca, New York. Their telephone number is (607) 273-8208. They offered access to the coordinates I needed through their MAX on-line service.

The MAX service was attractive since the license cost was based primarily upon the number of polygons downloaded. Since the cartographic outlines of the 169 Connecticut MCDs (Minor Civil Divisions), also known as cities and towns, typically do not change from year to year, it seemed silly to pay for them on an annual basis.

The MAX service was also an economical way to purchase zip code area polygons as well. Zip code areas do change periodically. For example, Bethany just received its own zip code, 06524. With MAX, you can purchase just the polygons needed. The MAX service requires its own software, MAXLINK, and there is a minimum charge each month, regardless of activity.

The only hitch is that the MAX polygons are designed for use with Claritas/NPDC’s own mapping program and are not in a format that SAS can use directly. However, this shortcoming is easily overcome as you will see in next part of this paper.

Part 3:
Converting MAXLINK Boundary Polygon Coordinates

In Part 2, it was observed that boundary polygon coordinates could be obtained from the MAXLINK service of Claritas/NPDC. However, to use them with PROC GMAP, the following conversions must take place:

First, the separate polygon label records, which appear at the start of each set of polygon coordinates must be eliminated. The name of the polygon must be copied as a variable to each coordinate record (observation).

Next, MAXLINK repeats the first set of coordinates at the end of the coordinate list to close off the polygon. However, SAS closes off the polygon automatically so the last coordinates must be dropped.

MAXLINK records appear in comma-delimited format, with double quotations around polygon labels. These records must be read into a SAS data set.

Finally, MAXLINK records appear as latitude and longitude coordinates. They must be converted to X-Y coordinates.

The following code may be used to perform these conversions:

```plaintext
filename rawdata 'MAXLINK .bas file' ;
infile rawdata dlm=',' missover;
retain county;
drop Z;
length COUNTY $18 Z $20;
input X Y Z;
```
if substr(Z, 1, 3) = ' ' then
  county = compress(Z, '');
  line = _n_;
if X < -1 then output;
r
The DLM= tells SAS to accept comma-delimited data. The MISSOVER option tells SAS to set the value of variables to missing instead of going to the next line to fill the program data vector.

The RETAIN statement holds the value of the polygon label until the data for the next polygon label is encountered. The COMPRESS function strips the double-quotes from the polygon label.

The line = _n_ generates the sequence numbers, needed to preserve the order of polygon coordinates through the sort. Without this statement, your GMAP output will look like what happens when the cat chases a ball of yarn. The x < -1 subsetting IF statement suppresses output of polygon label record.

Next, we sort the data by county and line to alphabetize the observations and to strip off the first pair of coordinates for each county that is repeated at the end of the coordinate list for each county.

proc sort data=hasug;
  by county line;
r
run;

data hasug;
set hasug;
by county;
drop line;
if last.county then delete;
r
Next, we use the GPROJECT procedure to convert the latitude and longitude coordinates to X-Y coordinates. As you may remember from early schooling, map projection is akin to removing the peel of an orange and attempting to lay it flat.

Some distortion is inevitable as you change coordinates from a spherical system to a plane. The GPROJECT procedure defaults to the Albers projection method but depending on your map, another method may yield more satisfactory results.

The EASTLONG option specifies that the value of longitude increase to the east instead of the west. The DEG option specifies that the coordinates are in degrees instead of radians.

proc gproject data=hasug out=hasugmap eastlong deg;
  id COUNTY;
r
Now that our boundary coordinates are in the format that the GMAP procedure can use, we will see in the next part of the paper some of the presentation choices available when drawing a map with SAS.

Part 4: Major Features of PROC GMAP

The third part of this paper covered the conversion of latitude and longitude coordinates into the format used by PROC GMAP. In this section, the major features of PROC GMAP required to draw a geographic map will be explored.

When you invoke the GMAP procedure, the PROC GMAP statement specifies the map data set containing the boundary coordinates, as discussed in previous installments. The response data set must also be defined. This is accomplished by including the DATA= parameter in the PROC GMAP statement.

An ID statement is also required. It selects the variable(s) that identifies the polygons to be charted. The variable(s) must be common to both the map and response data sets. For example, to chart data by state, include the SAS variable containing the values for state in the ID statement.

Often, the biggest decision is selection of the type of map to be drawn. Version 6 offers four map types: choropleth, prism, surface, and block. While the choice of map is largely aesthetic, choropleth maps often take less memory to execute, which can be significant when running SAS/GRAPH under Release 6.04 of the SAS system. Also, choropleth maps work well with most types of response data and polygon shapes. Thus, choropleth maps are a popular choice.

One disadvantage of choropleth (and prism and surface) map is that the appearance of the response is a function of the size and shape of
the polygons being charted. If the polygons have similar shapes and sizes, there is no problem. However, if this is not the case, the interpretation of the response data may be distorted. Block maps overcome this problem and are a good choice when a limited number of polygons are shown.

A bare-bone program for a choropleth map would be:

```sas
proc gmap map=map boundary data set
  data=response data set,
  id unit area variable(s);
  choro response variable(s) / options;
DISCRETE and ANNOTATE= are two commonly selected options. DISCRETE, coupled with a user-defined format, allows categorization of the response variables. This results in a less cluttered map. To associate a user-defined format with the response variable(s), add a FORMAT statement to the PROC GMAP code shown previously.

ANNOTATE= specifies a data set containing annotations, which are frequently used as reference points. In the case of geographic maps, major cities are often included as annotations. Some find the documentation regarding annotate data sets daunting. As an aid, the following example is offered. It shows the creation of the CITIES data set through CARDS input:

data cities;
  length function style color $ 8 test $ 20
  position xsys ysys $ 1;
  input function 1-5 x 7-12 y 14-19
  position 21 xsys 23 ysys 25 text 27-36
  style 38-44 color 46-50;
  cards;
LABEL 0.0004 0.0030 5 22M SPECIAL WHITE
LABEL 0.0004 0.0030 8 2 2 Hartford SWISSL WHITE
; run;

Each of the variables shown in the preceding data step is covered in detail in the SAS/GRAFH User's Guide. The annotate data set created places a star at the indicated x-y coordinates. The text, "Hartford" appears directly below the star. The x-y coordinates were determined by interpolating the boundary coordinates of the Hartford polygon, after projection, for the value of the center.

The following example shows the pattern statement and the GMAP procedure used to produce a prism map that appeared last year in the HARTFORD Area SAS Users Group newsletter, the HASUG FLASH, reproduced as Figure 1.

```sas
  goptions ctext=black ftext=swissl;
pattern1 v=e color=black;
pattern2 v=m1 color=black;
pattern3 v=m1n45 color=black;
pattern4 v=m1n90 color=black;
pattern5 v=m1x color=black;
pattern6 v=m3n45 color=black;
pattern7 v=m3n90 color=black;

Pattern statements control how the polygons will be filled in by the GMAP procedure. In this example, Pattern1, v=e, is an empty fill. If we had used the pattern v=s, the fill would have been solid black. Pattern2, v=m1, specifies a thin line, no slant. Pattern3, v=m1n45 specifies a 45 degree slant while Pattern4, v=m1n90, specifies a horizontal line. Pattern5, v=m1x, specifies a cross-hatch. Pattern6 and Pattern7, v=m3n45 and v=m3n90, are similar to Pattern3 and Pattern4 except a heavier line is used and there is more space between the fill lines.

```sas
proc gmap map=county data=hasug;
id county;
  prism count / discrete nolegend
  annotate=ctyname;
  label count= "Members";
  title1 F=swissb H=S pct
      'HASUG Membership by County';
run;

The DISCRETE option identifies the response variable, count, as a numeric variable with discrete, not continuous values. NOLEGEND suppresses printing of the display showing how the patterns are mapped to the county names. Since the county names are shown on the top surface of the prisms, the legend is not necessary.

PART 5: Creating the Annotate Data Set Automatically

In the preceding example, the annotate data set was created manually. Centering the annotate labels within each polygon involved considerable trial and error. If the example had been a choropleth map, then we could have automated the generation of the annotate data set. The following code is an adaptation of GMAPCTR, one of the sample programs supplied with SAS/GRAFH.
HASUG Membership by County

FIGURE 1
data map(drop=xlast ylast npoints)
points(keep=xlast ylast npoints);
set annotate data set;
by id variable;
flag='M';
output map;
npoints + 1;
if last.town then do;
xlast=x; ylast=y;
output points;
npoints=0;
end;
run;

data centers(keep=id variable flag style
xsys ysys when function text x y color size);
length function color $8 text $25;
length xsys ysys when $1;
retain xcom ycom ptr 1 savptr 1 xold yold
xcg ycg 0 aresum 0 when 'a' xsys ysys '2';
do ptrv=1 to nobsv;
set points point=ptrv nobs=nobsv;
xcom=xlast; ycom=ylast;
xcg=0; ycg=0;
aresum=0;
firstpnt=1;
ptr=savptr;
endptr=ptr + npoints - 1;
do ptrm=ptr to endptr;
set map point=ptrm nobs=nobsm;
if firstpnt then do;
xold=x; yold=y;
savptr=ptrm + npoints;
firstpnt=0;
end;
aretri=(xcom - x) * (yold - ycom) +
(xold - xcom) * (y - ycom);
xcg=xcg + aretri * (x + xold);
ycg=ycg + aretri * (y + yold);
aresum=aresum + aretri;
xold=x; yold=y;
end;
areinv=1.0 / aresum;
xcg=(xcg * areinv + xcom) * 0.3333333;
ycg=(ycg * areinv + ycom) * 0.3333333;
flag='C';
function='label';
x=xcg; y=ycg;
size=.50;
color='black';
text=id variable;
style='SWISSL';
output;
end;
stop;
run;

This code segment gets inserted after the GPROJECT procedure since we have to match up with the x-y coordinates of the map polygons. All you have to do is substitute in the data set with the projected coordinates of the annotate features in the SET statement in the first data step.

Where you see id variable, you substitute in the name of the variable used to match up the polygon coordinates with the response data. It is not necessary to fully understand this code to use it.

The next section discusses graphic device drivers with an emphasis on techniques to speed up printing maps and other graphics when using HP LaserJet printers.

PART 6:
HP LaserJet Graphics Device Drivers and Printing Maps

Last year, a wonderful pair of articles appeared in Observations TN, the Technical Journal for SAS Software Users. Written by Robert Dolan and Julie Platt, they did an excellent job of explaining HP LaserJet graphics drivers, especially with respect differentiating between bitmap (PCL or Printer Control Language) and vector graphics (HP-GL or Hewlett-Packard® Graphics Language). Anyone using SAS/GRAPH with a HP LaserJet printer should read these articles.

With respect to producing maps, the following information may assist you. First, unless you are very skilled, creating a map involves some trial and error. Rather than waste time and CPU cycles in the production of draft maps, always start with the lowest resolution device driver available. I usually start with the HPWO driver (75 dpi or dots per inch). When I need to show a draft to someone else, I step up to the HPW5P2 (150 dpi). Another feature of the HPW5P2 driver is that stock HP LaserJet III printers without additional memory work with this driver. I reserve use of the HPWS2 driver (300 dpi) for final copies.

If you have been "blessed" with printing through a network, some additional techniques may be needed. Network printers often deduce that a spooled print job is complete and issue a page eject after a preset amount of time has elapsed. If you are producing a complex map and print directly to a network printer, you may find that your map has been sliced into
several sections, each section appearing on a separate page. How do you solve this problem? Try printing to a graphics stream file (GSF) using code similar to:

filename grafout 'c:\mysas\mymap.gsf';
goptions device=hpljs2 rotate noprompt
gsfname=grafout gsfmode=replace;

Then send your job to the printer by copying the graphics stream file to the printer. For example, on a personal computer running Microsoft MS-DOS®, the command would be:

copy c:\mysas\mymap.gsf prn /b

The /b parameter (binary copy) tells MS-DOS (and the network) to ignore end-of-file characters embedded within the graphics stream file.

As previously noted, a drawback of using PCL graphic drivers is that at 300 dpi resolution, your graphics may take a long time to run and print. An alternative to consider is to use a HP-GL or HP-GL/2 driver. If you have been using one of HP LaserJet III series, HP-GL/2 is built in so all you or your driver has to do is switch the printer to the HP-GL/2 mode.

Since at least some of the HP LaserJet III series do not provide front panel controls to switch between PCL and HP-GL/2, you have to use software to accomplish this. If the device driver you select does not switch the printer mode, then consider using software such as InPlot® from Insight Development Corporation (1-800-825-4114). On request, either Hewlett-Packard or Insight will send you a free demonstration disk that includes programs to switch between the two modes.

What do you do if you have a HP LaserJet Series II printer? While a HP-GL/2 interpreter is not on board, you are not out of luck. Pacific Data Products sells a cartridge called Plotter in a Cartridge (PIAC). Software to convert HP-GL files to other formats is also available.

I use PIAC and it works well. Any driver for the HP 7475 pen plotter (e.g., HP7475A) will work well so long as you remember that you only have one pen, whose color is black. If you are interested in the PIAC, then you or one of your site consultants should contact SAS Institute Technical Support and request documents TS-150 and TS-168 (TS-204).

There are a few drawbacks to the HP-GL and HP-GL/2 drivers that you should keep in mind. First, you need about 2 MB of memory on your HP LaserJet to print a full page of graphics. Second, some of the fonts have a "stick-like" appearance so you may find the PCL drivers more to your liking. Third and last, very complicated graphics with substantial annotation created with the HP-GL and HP-GL/2 drivers may generate a larger graphics stream file than if they had been created with the PCL drivers.

PART 7: Shading and Colors

One of the esthetic choices one has to make when they draw a map is how to help the viewer distinguish among the areas of each polygon. This is especially important where one uses the appearance of the polygons to show the value of the response data set.

As you could see from the HASUG Membership map example, one method is to fill each polygon with lines or cross-hatching. This may be a good choice when the map has to be reproduced by an ordinary photocopier. The drawback is that such maps are rarely as attractive as they could be.

Fortunately, Hewlett-Packard and other printer manufacturers have recently developed excellent color printers that can be obtained for as little as $500. If you do not have to reproduce many copies of a map, then color is an excellent way to associate values with polygons.

However, if you have to reproduce the map in volume or a color printer is not available, there are other good alternatives. First, using the techniques discussed in Part 5, you can annotate the polygons with the response values. This is a minimalist approach that works very well. An example of a map created with this approach is shown in Figure 2.

Another approach is to use gray-scales. There are two approaches to generating gray scales on laser printers. First, if your printer is capable of interpreting the PostScript® page description language, then it is simple to specify the shades of gray you wish through the PATTERN statements. The shades of gray range between GRAY00 through GRAYFF. The map shown in Figure 3 was produced on a HP...
LaserJet Series II printer with a PostScript cartridge and additional memory.

If a PostScript printer is not available, HP LaserJet III and IV printers are also capable of generating grayscales. See the first citation by Robert Dolan and Julie Platt listed in the references for more information.

Conclusion

I hope that this paper will help you begin to use maps. Properly used, results displayed on a map can better help you visual your information and draw favorable attention to your results. I have listed some other materials that you may find to be helpful references.

If this paper has helped you or if you have questions or suggestions, the author would like to hear from you. Please contact him at:

Michael L. Davis  
Staff Assistant, Research  
BlueCross Blue Shield of Conn.  
370 Bassett Road  
North Haven, CT 06473-4201  
(203) 985-7187 (voice)  
(203) 985-7030 (fax)

If you are connected to Internet, you may send messages to 0002395748@mcimail.com. If you are a CompuServe subscriber, you may send messages to 75046,1627. The author appreciates receiving tips and other insights from other SAS users.

ACKNOWLEDGEMENTS


Microsoft, MS-DOS are registered trademarks and Windows is a trademark of Microsoft Corporation, Redmond, WA.