

# PUTTING YOURSELF ON THE MAP WITH THE GMAP PROCEDURE

Michael Davis, Bassett Consulting Services, Inc., North Haven, Connecticut

## ABSTRACT

Many users are fascinated by maps although they seem too complex to be produced by an average user. This paper introduces to 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 discusses 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 match 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 once enlivened 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 Part 3.

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 US 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 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/Line files show street intersections, rivers, and other landmarks.

If the TIGER/Line 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/Line files that you need from the Census Bureau. However, these files and much more can be purchased from Geographic Data Technology, Inc. in Lebanon, New Hampshire. They are a reputable company that has been in business since 1980. Call them at (800) 943-8727.

Another reputable vendor for coordinate data is Claritas. This company has offices in New York City and elsewhere. Call them at (800) 876-6732. Claritas sells map coordinates as SAS data sets or you can download the polygons you need through Claritas Connect, which is their online product.

### Part 3: Converting Boundary Polygon Coordinates to SAS Data Sets

In Part 2, it was observed that boundary polygon coordinates could be obtained from the from several sources, including online services. However, it is sometimes necessary to convert the raw files to a format that can be used with the GMAP procedure.

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

Some raw data files repeat 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.

Raw data files used with desktop computers often are in comma-delimited format, with double quotations around polygon labels. Special SAS coding techniques are needed to read these files successfully.

Finally, map coordinates obtained from commercial sources often are recorded as latitude and longitude coordinates. They must be converted to X-Y coordinates. The following is an example from an earlier presentation of using the SAS System to perform these conversions:

```
filename rawdata 'MAXLINK .bas file' ;
data hasug ;
  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;
run;
```

The DLM= tells SAS to accept comma-delimited data. The MISSEVER 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;
run;

data hasug;
  set hasug;
  by county;
  drop line;
  if last.county then delete;
run;
```

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;
run;
```

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 PROC GMAP features used 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 using a desktop computer with modest performance. 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 follows:

```
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 2 2 M          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/GRAPH 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 sample prism map that appears at the end of this paper, along with the SAS code used to generate it.

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

```

pattern6 v=m3n90 color=black;
pattern7 v=m3x color=black;
pattern8 v=m5n45 color=black;

```

Pattern statements control how the polygons will be filled in by the GMAP procedure and the order in which they are used is governed by the values being charted. To see how each pattern looks, see the sample map:

```

Pattern 1= Windham   Pattern 5= New London
Pattern 2= Tolland   Pattern 6= New Haven
Pattern 3= Middlesex Pattern 7= Fairfield
Pattern 4= Litchfield Pattern 8= Hartford

```

If we had specified a pattern v=e, an empty fill would have been used. Alternatively, pattern v=s would have resulted in a solid black fill.

```

proc gmap map=county data=hasug;
  id county;
  prism count / discrete nolegend
  annotate=ctyname;
  label count= "Members";
  title1 F=swissb H=5 pct
  'HASUG Membership by County';
run;

```

The DISCRETE option identifies the response variable, count, as a numeric variable with discrete, not continuous values. NOLEGEND suppress 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/GRAPH

```

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 nobsv=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 nobsv=nobsv;
  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) *
  0.333333);
  ycg=((ycg * areinv + ycom) *
  0.333333);
  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 used when printing maps from Microsoft Windows using HP LaserJet printers.

## **PART 6: Printer Drivers and Printing Maps**

During the past few years, the SAS System has made major advances in the way in which the handling of graphical output has been integrated into the desktop operating environment. As a result, when printing maps from a desktop platform using Release 6.11 or later, there are two approaches to creating hardcopy maps:

- SAS/GRAPH device drivers
- SAS/GRAPH interface drivers

A SAS/GRAPH device driver produces the data stream needed to generate a graphics image. It must be specified within your SAS program. A SAS/GRAPH interface driver produces output using the Windows device driver for the select graphics device.

Under Release 6.11, printing using the SAS/GRAPH interface driver is often the best solution. One can use any printer that is supported by Windows instead of having to rely on the availability of a "native" SAS/GRAPH for that printer.

To print using the SAS/GRAPH interface drivers under Microsoft Windows or Windows 95, include the following statement in your map program before the GMAP procedure to select the display driver:

```
GOPTIONS device= win ;
```

Actually, device= win is default so you might wish to try running your program without this option. If needed, you can supply other SAS/GRAPH (graphic) options on this statement. The most common option is rotate (or norotate), used to control whether the map image is printed in portrait or landscape mode.

Graphical output generated in this manner is temporarily stored in the WORK.GSEG catalog as GRSEG entries. The default name for the GRSEG entries is the name of the procedure that created them. Thus, for GMAP procedure output, the first catalog entry is GMAP (GMAP.GRSEG). The next image is saved to GMAP1, then GMAP2 and so on.

To review the maps, put down the Globals menu and select Graph. In the Graph window, one can scroll through the GRSEG entries (images) using the Page Up or Page Down keys. If the number of entries in WORK.GSEG becomes unmanageable, you can delete obsolete entries from the Catalog window or the entire WORK.GSEG catalog from the Libraries window.

When you want to print the GRSEG entry that you are viewing, pull down the File menu and select Print Setup. Accept the printer and connection shown or select another. Then pull down the File menu a second time and select Print. From this menu, you can print directly to a printer or to file.

If you are using an earlier release of the SAS System or wish to use an approach that can be ported to non-desktop platforms, then using the SAS/GRAPH device drivers may be the approach to use.

How do you know which device driver to use? One approach is to start up an interactive SAS session on the platform on which SAS/GRAPH is to be used. From the Libraries window, select the SASHELP libref and open the DEVICES catalog for browsing. Scroll through the list until you find an appropriate printer driver.

If you want to see the driver settings, type in the following program:

```
PROC GDEVICE catalog=sashelp.devices ;  
run ;
```

Put a question mark next to the driver for which you would like to see more detailed information. Select Browse. Incidentally, if you wish to see the details for the Windows SAS/GRAPH interface drivers, they are stored in the SASUSER.DEVICES catalog.

A few years ago, a wonderful pair of articles appeared in **Observations**<sup>™</sup>, 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). If one is using SAS/GRAPH with a HP LaserJet printer, it might be worthwhile to seek out 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. If you are working with a complex map or a slow computer, consider starting with a low resolution device driver. As an example, for an HP

LaserJet, you might try the HPLJ0 driver (75 dpi or dots per inch) or the HPLJ5P2 (150 dpi). Another feature of the HPLJ5P2 driver is that HP LaserJet printers with one megabyte (or less memory) memory work with this driver. When you use a 300 dpi driver, (e.g., the HPLJS2 driver) at least 1.5 megabytes or more printer memory may be needed.

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.

A drawback of using PCL graphic drivers is that at higher resolutions, older model printers may take a long time to print your map. One alternative is to use a HP-GL or HP-GL/2 driver. If you are using an HP LaserJet III series printer or later version, HP-GL/2 is usually built-in. So all you or your driver has to do is switch the printer to the HP-GL/2 mode.

If your printer does 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, software such as Print-A-Plot® from Insight Development Corporation (1-800-825-4115) is a possibility. Another possibility is for your or your company to invest in a faster printer.

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 those 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 sample map that appears at the end of this paper, 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, printer manufacturers have recently developed excellent color printers that can be obtained for only a few hundred dollars. 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. The sample maps shows how this may be done.

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.

If a PostScript printer is not available, HP LaserJet III and later model 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 visualize 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  
Vice President  
Bassett Consulting Services, Inc.  
10 Pleasant Drive  
North Haven CT 06473-3712  
(203) 562-0640  
(203) 498-1414 (fax)  
Internet: 0002395748@mcimail.com

## REFERENCES

Bessler, LeRoy. (1996), "Better Low-Cost InfoGeographics Today: Improved Readability and Usability for PROC GMAP Output," Proceedings of the Twenty-First Annual SAS Users Group International Conference, 21, 820-829.

Blodgett, John and McIntyre, John. (1990), "Software Tools for Processing U.S. Census Bureau TIGER Line Files," Proceedings of the Fifteenth Annual SAS Users Group International Conference, 15, 942-946.

Dolan, Robert and Platt, Julie. (1992), "SAS/GRAPH Drivers for the HP LaserJet," *Observations*, 1:3, 10-18.

Dolan, Robert and Platt, Julie. (1992), "Customizing HP LaserJet Device Drivers," *Observations*, 1:4, 20-32.

SAS Institute Inc., SAS Technical Report A-107, *Creating Your Own SAS/GRAPH Map Data Sets with a Digitizer*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS/GRAPH Software: Volume 1, Reference, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS/GRAPH Software: Volume 2, Reference, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1991), *SAS/GRAPH Software: Usage, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), SAS Technical Report P-196, *SAS/GRAPH Software: Map Data Sets, Release 6.06*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1991), SAS Technical Report P-208, *SAS/GRAPH Software: Changes and Enhancements to Map Data Sets, Release 6.07*, Cary, NC: SAS Institute Inc.

Wind, Jay Jacob. (1991), "SAS/GRAPH Software: All the Maps You Will Ever Need," Proceedings of the Sixteenth Annual SAS Users Group International Conference, 16, 737-742.

```

***** ;
* ;
* Sample Map for SUGI 22 Presentation * ;
* Michael Davis, Bassett Consulting Services, Inc. * ;
* ;
***** ;

```

```

goptions ctext=black ftext=swissl ;
pattern1 v=m1 color=black ;
pattern2 v=m1n45 color=black ;
pattern3 v=m1n90 color=black ;
pattern4 v=m1x color=black ;
pattern5 v=m3n45 color=black ;
pattern6 v=m3n90 color=black ;
pattern7 v=m3x color=black ;
pattern8 v=m5n45 color=black ;

```

```

data ctyname;
  length function style color $ 8 text $ 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 ;
  datalines ;

```

```

LABEL 021.00 047.00 5 5 5 Fairfield SWISSB BLACK
LABEL 021.00 047.00 8 5 5 827,645 SWISSB BLACK
LABEL 045.00 085.00 5 5 5 Hartford SWISSB BLACK
LABEL 045.00 085.00 8 5 5 851,783 SWISSB BLACK
LABEL 020.00 072.00 5 5 5 Litchfield SWISSB BLACK
LABEL 020.00 072.00 8 5 5 174,092 SWISSB BLACK
LABEL 062.00 059.00 5 5 5 Middlesex SWISSB BLACK
LABEL 062.00 059.00 8 5 5 143,196 SWISSB BLACK
LABEL 043.00 058.00 5 5 5 New Haven SWISSB BLACK
LABEL 043.00 058.00 8 5 5 804,219 SWISSB BLACK
LABEL 085.00 069.00 5 5 5 New London SWISSB BLACK
LABEL 085.00 069.00 8 5 5 254,957 SWISSB BLACK
LABEL 066.00 082.00 5 5 5 Tolland SWISSB BLACK
LABEL 066.00 082.00 8 5 5 128,699 SWISSB BLACK
LABEL 081.00 084.00 5 5 5 Windham SWISSB BLACK
LABEL 081.00 084.00 8 5 5 102,525 SWISSB BLACK

```

```

;
data census ;
  input
  @01 county 2.
  @04 countynm $10.
  @15 census 6.
  ;

```

```

  datalines ;
01 Fairfield 827645
03 Hartford 851783
05 Litchfield 174092
07 Middlesex 143196
09 New Haven 804219
11 New London 254957
13 Tolland 128699
15 Windham 102525
;

```

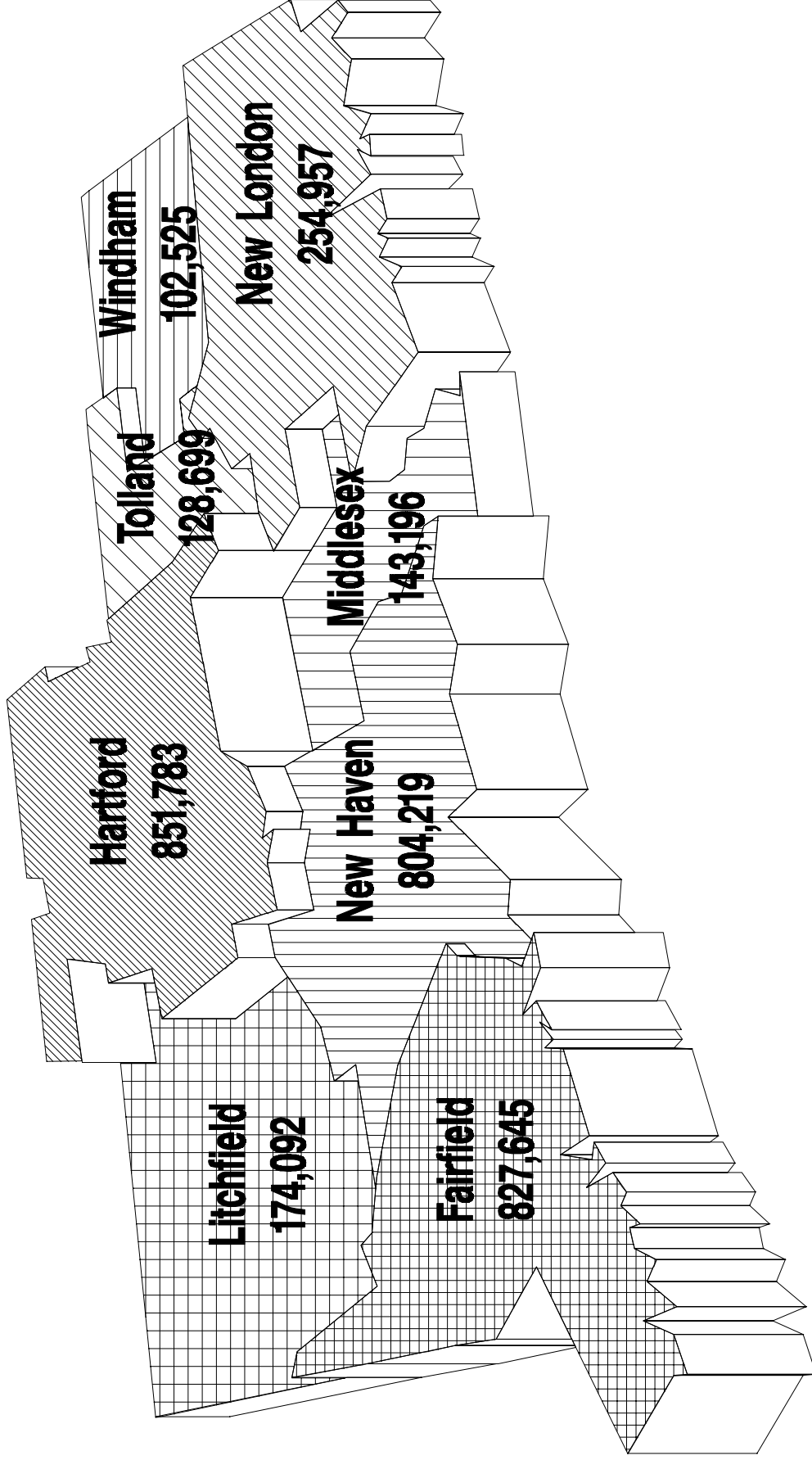


```
data county;
  set maps.uscounty;
  line= n ;
  if state= 9 ;
run;

proc sort data=county out=county(drop= line) ;
  by county line ;
run ;

proc gmap map=county data= census annotate= ctyname ;
  id county ;
  prism census / discrete nolegend ;
  label census= "Population" ;
  title1 f=swissb h=5 pct
  'Connecticut Population by County' ;
  footnote1 f=swissl j=1 h=2.5 pct
  'Prepared by Michael Davis' ;
  footnote2 f=swissl j=1 h=2.5 pct
  '
    Bassett Consulting Services, Inc.' ;
  footnote3 f=swissl j=1 h=2.5 pct
  '
    July 1996' ;
quit ;
run ;
```

# Connecticut Population by County



Prepared by Michael Davis

Bassett Consulting Services, Inc.

July 1996