Paper 138-27

Creating Maps in SAS/GRAPH®

By Jeffery D. Gilbert, Venturi Technology Partners, Kalamazoo, MI

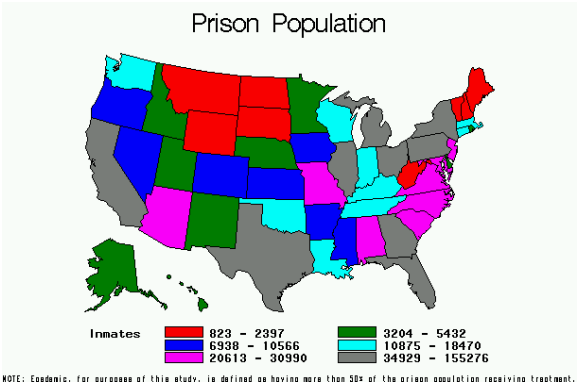
Abstract

Integrating maps in SAS/GRAPH and customized annotations presents many challenges, particularly when working with the placement of annotations on maps. This can be very tricky, and can require many iterations of trial and error to get it right. This paper will provide fundamentals on creating maps using the PROC GMAP procedure, and also give an overview of the annotate facility provided by SAS/GRAPH. The two topics will then be integrated to show how the annotate facility can be used to customize maps. Particular emphasis will be placed on two-dimensional maps, dealing with the positioning of the annotations and specific issues such as the overlapping of annotations, while three-dimensional maps will be mentioned.

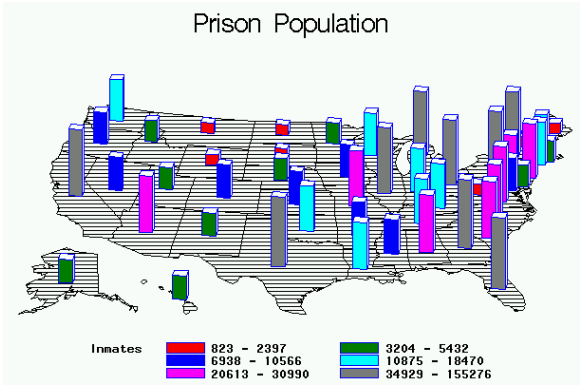
Maps

Maps are a great way to visualize geographical data, whether discrete or continuous. Using The SAS System, you can create four different types of maps:

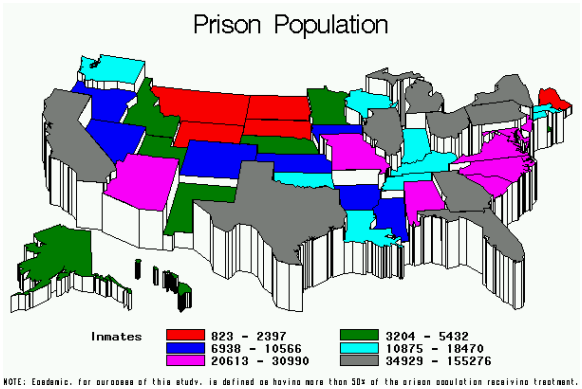
The CHOROPLETH map, which is a basic two-dimensional map.



The BLOCK map, a three-dimensional map which places blocks on each state.



The PRISM map, a three-dimensional map that can offer a unique perspective on the differences between areas.



The SURFACE map, another three-dimensional map useful for visualizing clusters. For the purposes of this paper, this type of map will not be discussed.

Map Tables

The GMAP procedure in SAS/GRAPH will produce any of the three map types, all with relatively similar syntax. Every call to the GMAP procedure will include two different tables: the MAP table and the DATA table.

The MAP table contains the x-axis and y-axis coordinates used to draw the map, as well as a variable like “state” to give meaning to the coordinates and allow for linking up to the DATA table.

The DATA table contains the actual data to be graphed. This data can be discrete or continuous. But it also needs to contain a

variable, like “state”, which corresponds with values of “state” on the MAP table so the values can be properly mapped.

The GMAP Procedure

The GMAP procedure produces maps in SAS/GRAPH. Every GMAP procedure will have four basic parts:

1) The PROC GMAP procedure call, with the MAP= and DATA= options specifying the required tables. One important option is the “ALL” option. When creating a map of states, excluding the ALL option will only map those states that have data in the DATA table. This is sometimes desirable, especially if only creating a map for a specific region. However, this is not always desirable, because if a map of the entire country is desired, and say the state of Florida has no data, then the entire country will be mapped without Florida.

   EXAMPLE:

```
PROC GMAP MAP=maps.us DATA=prison ALL;
```

2) The ID statement – this declares the variable that is to link the DATA and MAP tables together. This can be one or more variables.

   EXAMPLE:

```
ID state;
```

3) The map statement to define the exact map to be graphed. This includes one of three different statements (CHORO, PRISM, or SURFACE) to define the actual map. The variable being mapped will come next, and options should be coded in following a slash (“/”).

   EXAMPLE:

```
CHORO epedemic / discrete;
```

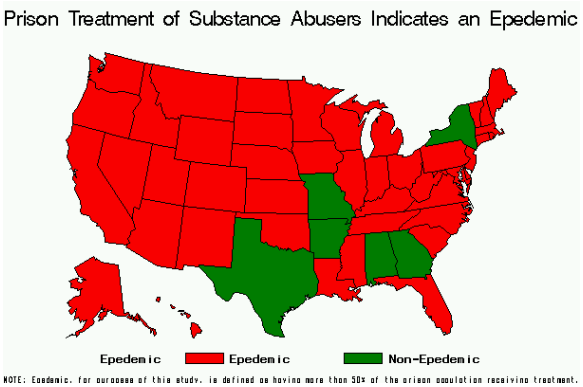
4) The RUN statement. The run statement is imperative to creating the map, telling the procedure when the last statement has been issued and that The SAS System can now create the map. Then the QUIT statement will exit the procedure. Both are necessary.

   EXAMPLE:

```
RUN; QUIT;
```

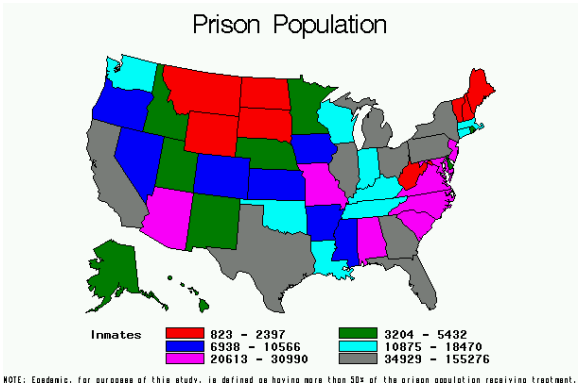
The code from the above examples will produce the following map:

```
PROC GMAP MAP=maps.us DATA=prison ALL;
  ID state;
  CHORO epedemic / discrete;
RUN; QUIT;
```



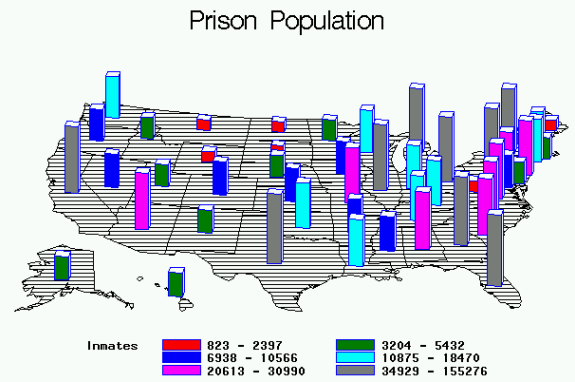
This example showed a map from a two-dimensional discrete map using the CHORO statement. Below is the statement to create a two-dimensional continuous map using the CHORO statement.

```
PROC GMAP MAP=maps.us DATA=prison ALL;
  ID state;
  CHORO inmates;
RUN; QUIT;
```



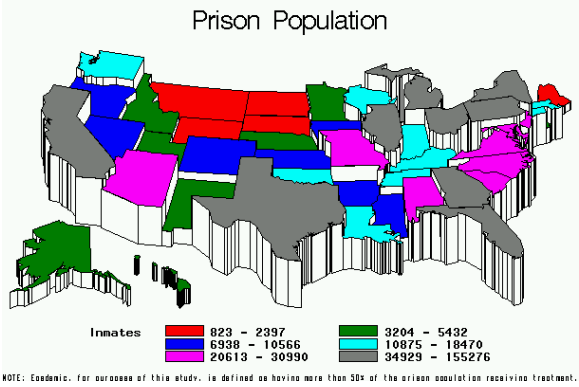
This same map can be created as a three-dimensional map, using either the BLOCK statement:

```
PROC GMAP MAP=maps.us DATA=prison ALL;
  ID state;
  BLOCK inmates / cblkout=blue;
RUN; QUIT;
```



or the PRISM statement:

```
PROC GMAP MAP=maps.us DATA=prison ALL;
  ID state;
  PRISM inmates;
RUN; QUIT;
```



Sprucing It Up

You will notice that these maps don't look particularly nice, because they use the default patterns and make it difficult to differentiate between the states, especially in black and white.

Using the PATTERN global statement can help make the distinctions clearer. The PATTERN statements take the following syntax:

1) PATTERN<x>, where x is the pattern number (up to 20). One pattern will be used for each breakout of the data – if there are seven regions (as in the case of the upcoming example), then seven PATTERN statements should be specified. Otherwise, a default pattern will be used, which can be undesired.

2) V=m<y><c><z>, where:

   a) y = an integer specifying how many lines are to be drawn,

   b) c = either N or X. An N specifies that the lines in the pattern are to be drawn at an angle, and X specifies that the lines are to be criss-crossed, and

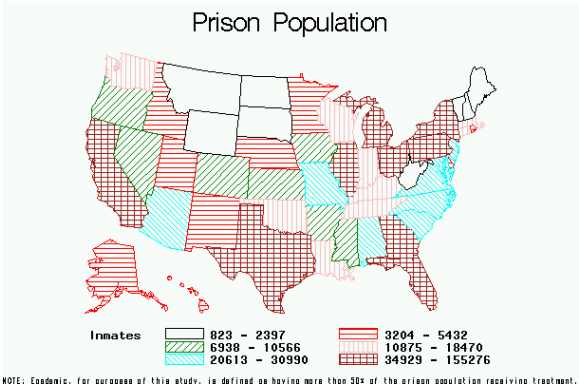
   c) z = an integer which specifies the angle at which the lines are drawn.

3) C=<color> to specify a color.

For example, the following pattern statements, when applied to the CHORO map produced earlier:

```
pattern1 v=e  c=black;
pattern2 v=m1n00  c=red;
pattern3 v=m2n45  c=green ;
pattern4 v=m3n90 c=pink  ;
pattern5 v=m4n135 c=cyan  ;
pattern6 v=m1x00  c=brown ;
```

Results in the following map:



On color or black and white output, the PATTERN global statement greatly increases the readability of this map.

Annotating Maps

Sometimes a map is needed with annotations. The SAS System offers some canned annotations through GLOBAL statements, such TITLE, FOOTNOTE, AXIS, and LEGEND. However, often custom lines need to be drawn, or custom text put in a certain spot on a map. For this, the Annotate Facility is provided.

One important note, though, is that The Annotate Facility does not work particularly nicely with any three-dimensional graphs.

Because you have to work with the x- and y-axis of the graphing region, as defined in the MAP table, and SAS/GRAPH takes those values and distorts them for the purpose of making the map three-dimensional, placing a custom annotation on a three-dimensional map becomes very tricky and volatile.
To use the Annotate Facility, two steps are required. First, create a special table corresponding to the MAP table, and then point the GMAP procedure at this special table using the ANNOTATE= option.

The special annotate table must contain eleven specific variables, which fit into three categories: what to do, how to do it, and where to do it. The variables are:

What to do

1) **Function** – an 8-byte character field, the function variable will define exactly what needs to be done. Some examples of functions are: “LABEL”, “DRAW”, “POLY”, “MOVE”, and “SYMBOL”.

2) **Text** – a character field between 1 and 200 bytes long, this will define what to display. This is needed if using the “LABEL” or “SYMBOL” functions, but is ignored for functions like “DRAW” and “MOVE”.

How To Do It

3) **Color** – an 8 byte character field, color specifies the color of the line or text to be drawn. Examples of colors are: “BLUE”, “GREEN”, “RED”, or “BLACK”.

4) **Style** – an 8 byte character field used for selecting the font or pattern of the text or line.

5) **Size** – a numeric field specifying the height or thickness of the text or line drawn.

6) **HSYS** – a one byte character field used to select the coordinate system to determine the size. Some common values of HSYS are:

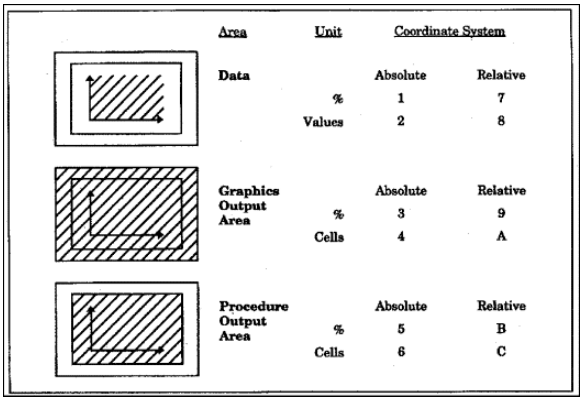


Figure 1 Source: SAS OnlineDoc

Where To Do It

7) **X** – a numeric field representing the horizontal coordinate value. This would be taken from the MAP table, and is often obtained by taking the MAP table, and finding the mean X value for each state.

8) **XSYS** - a one byte character field used to select the coordinate system to determine the exact location of the X variable. See the documentation of HSYS for some example values.

9) **Y** - a numeric field representing the vertical coordinate value. As with the value of X, this is often obtained by taking the MAP table, and finding the mean Y value for each state.

10) **YSYS** - a one byte character field used to select the coordinate system to determine the exact location of the Y variable. See the documentation of HSYS for some example values.

11) **Position** – a one byte character field telling SAS/GRAPH where to draw text in relation to the point (x,y). For example, Tells where to draw text in relation to the point (x,y). Values are typically '1' to '9' or 'A' to 'F'. The table below illustrates the values '1' to '9', where the value of '5' indicates the position should be right on the given point. For example, position=’9’ positions the text one unit below the point, and one unit to the right.

1	2	3
4	5	6
7	8	9

The letters work similarly, except they work in half-unit increments. For example, 'A' positions the text one-half unit above the point, and one-half unit to the left.

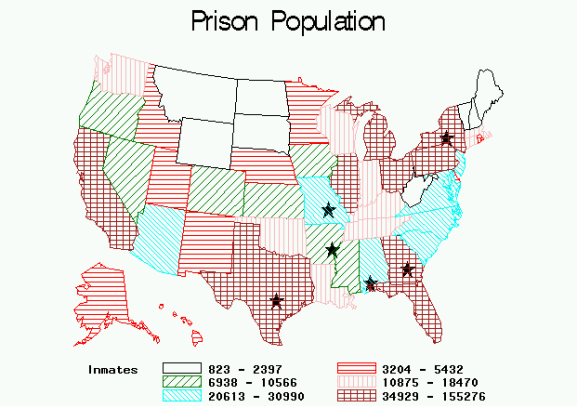
Creating the Annotate Table

To create an annotate table for use with GMAP, first get one unique value for each ID field, like STATE, using the MAP table and PROC SUMMARY. For example:

```
proc summary data=maps.us nway;
  class state;
  var x y;
  output out=anno(drop=_type_ _freq_)
 mean=;
run;
```

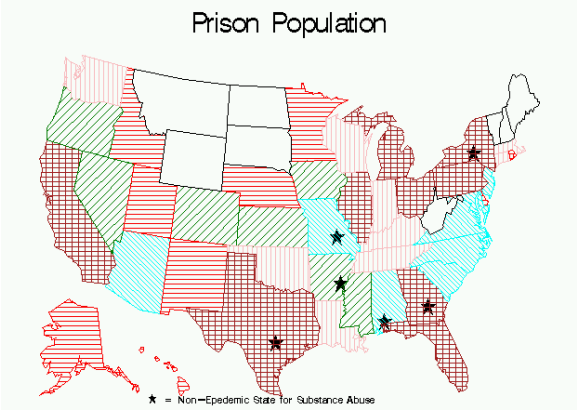
Creating the annotate table, using the variables mentioned above, is then fairly straight-forward. Essentially, for value of X and Y that an annotation is desired, an observation should be output. In the example below, only the states with the variable EPEDEMIC not equal to the value "Epedemic" will be output. Those records output will receive a "star".

```
data anno2;
  merge anno prison;
  by state;
  length function $8 text $200;
  color="BLACK";
  size=2;       hsys='4';
  xsys='2';  ysys='2';
  position='5';

  function="LABEL";
  style="SPECIAL";
  text="M"; * Special character - STAR.;

  if epedemic ne "Epedemic" then output;

run;
proc gmap all data=prison map=maps.us;
  title "Prison Population";
  id state;
  choro inmates / annotate=anno2;
run; quit;
```



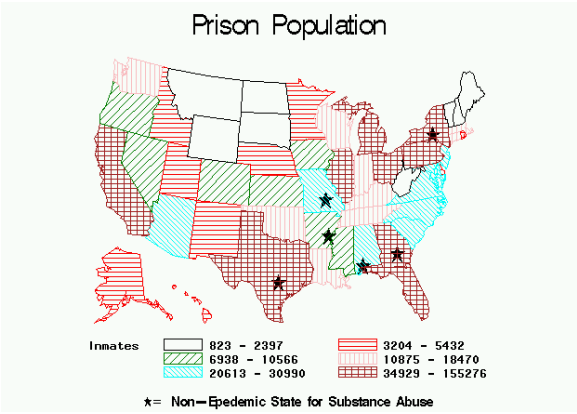
To take this a step further, a notation could be put on the map to indicate that a star refers to a non-epedemic state. The same code as above is used, with additional code to output the notation at the bottom of the report. Notice also that the NOLEGEND option is specified in this CHORO statement to eliminate the legend on this particular report, to make room for the annotation.

```
data anno3;
  merge anno prison end=end;
  by state;
  length function $8 text $200;
  color="BLACK";
  size=2;       hsys='4';
  xsys='2';  ysys='2';
  position='5';

  function="LABEL";
  style="SPECIAL";
  text="M"; * Special character for a STAR.;

  if epedemic ne "Epedemic" then output;

  if end then do;
        size=1;
        xsys="4";       ysys="4";
        x=20;           y=1;
        output;

        x=21;   y=1;
        position='6';
        size=.75;
        style="SWISSB";
        text=" = Non-Epedemic State for Substance
Abuse";
        output;
  end;
run;
proc gmap all data=prison map=maps.us;
  title "Prison Population";
  id state;
  choro inmates / annotate=anno3 nolegend;
run; quit;
```



However, it may be more desirable in this case to leave the legend on the map, and use the FOOTNOTE statement along with the previous annotate table. Here, the footnote can actually do a nicer job of annotating the map than the Annotate Facility:

```
proc gmap all data=prison map=maps.us;
  title "Prison Population";
  footnote font="SPECIAL" "M"
       font="SWISSB" "= Non-Epedemic State for
Substance Abuse";
  id state;
  choro inmates / annotate=anno2;
run; quit;
```

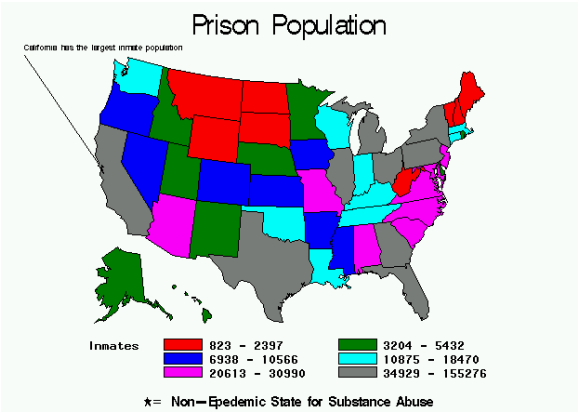


The Annotate Facility is also useful for drawing lines, as needed, on the map. For example, to draw a line from the STATE with the highest inmate population, use the same code as above, except add in the "DRAW" and "MOVE" functions:

```
* Remove patterns (global statements);
goptions reset=global;

* Create a macro variable with the value;
* Of the largest inmate population.;
proc sql noprint;
  select max(inmates)
    into :largest
       from prison;
quit;
```

```
data anno3;
  merge anno prison end=end;
  by state;
  length function $8 text $200;
  color="BLACK";
  size=2;       hsys='4';
  xsys='2';  ysys='2';
  position='5';

  function="LABEL";
  style="SPECIAL";
  text="M"; * Special character for a STAR.;

  if epedemic ne "Epedemic" then output;

  * ADDED TO DRAW THE LINE AND TEXT.;
  if inmates ge &largest then do;
       * Move the cursor to x,y;
       size=.5;
       function="MOVE";
       output;

       * Draw a line to 4,87;
       function="DRAW";
       x=4; xsys='3';
       y=87; ysys='3';
       output;

       * Add text.;
       size=.6;
       position='3';
       style="SWISSB";
       function="LABEL";
       text=trim(fipnamel(state))
              || " has the largest"
              || " inmate population";
       output;
  end;
run;
proc gmap all data=prison map=maps.us;
  title "Prison Population";
  footnote font="SPECIAL" "M"
       font="SWISSB" "= Non-Epedemic State for
Substance Abuse";
  id state;
  choro inmates / annotate=anno3;
run; quit;
```

Running this code gives the following map:



Annotate Facility Conclusion

Only one way exists to learn the Annotate Facility, and to get it right: trial and error. Some of the numbers above took several iterations of trial and error. This feature in SAS/GRAPH can, at times, be somewhat volatile if the data changes substantially. But clearly this facility adds much power to SAS/GRAPH that is needed.

Summary

SAS/GRAPH is a powerful graphing tool, particularly with respect to creating maps. While three-dimensional graphs tend to hide data results, rather than display them clearer, the two-dimensional graphs can offer huge benefits in data analysis. The Annotate Facility adds to the power, allowing custom text and drawings to be added to the facility. Much, much more information is available on both of these subjects in the SAS OnlineDoc. There is much to learn and explore.

Reference and Further Study

SAS OnlineDoc - SAS/GRAPH section.

Contact

Jeffery D. Gilbert (Jeff)
Venturi Technology Partners
5278 Lovers Lane
Kalamazoo, MI 49002
Work Phone: (616) 344-4100
Email: JefferyGilbert@yahoo.com