SUBSETTING AND COMBINING SAS' MAP DATA SETS
TO PRODUCE CUSTOM MAPS

Susan J. Slaughter, Consultant

Background

Sophisticated mapping capabilities set SAS/GRAPH apart from many other graphics packages; yet using these maps is, for the most part, like using a series of black boxes. The mapping procedures: PROC GMAP, GPROJECT, GREDUCE, and GREMOVE are some of the most inscrutable procedures in the SAS arsenal. It needn't be this way: understanding a few concepts can enable you to flexibly and creatively manipulate maps.

Using only the map data sets that come bundled with SAS/GRAPH you can map the US with its individual states and counties, and Canada with its provinces and census districts. Many other map data sets including international boundaries and US census blocks are available from the SAS Institute and from other companies.

The SAS/GRAPH manual includes examples of how to isolate an individual region such as a state or group of states, and how to remove boundaries between states creating larger regions. However, in many instances only a more customized map will do. For example, SAS does not provide a map data set that includes both the US and Canada. And the SAS/GRAPH manual never mentions that it is even possible to subset a map segment, such as to produce a map of the San Francisco Bay area. With SAS/GRAPH you have the flexibility to combine and subset maps, but doing this requires an understanding of how map data sets work.

Map Data Sets

All map data sets contain at least 4 variables:

1) id
2) segment
3) x
4) y.

The id variable is a code that identifies each state, province, county, or census district in a meaningful way. For example, the STATES data set uses the US Postal Service's FIPS codes for an id variable. The id variable may have different names. For example, in the STATES data set it is called "state" while in the CANADA4 data set it is called "province."

Segment is an arbitrary number that is shared by all the points that form one continuous polygon. In the STATES data set each state has its own segment number; in the CANADA4 data set each province has its own segment number.

Not surprisingly, the x and y coordinates indicate location and can be expressed in two different ways. If the map is unprojected, then x and y represent latitude and longitude measured in radians. For projected maps, x and y are expressed in units internal to SAS. The difference between these two ways of expressing x and y coordinates turns out to be extremely important.

Some map data sets contain a fifth variable, density, with values from 1 to 6, which can be used to reduce the resolution of a map.

The mapping procedures are: PROC GMAP, GREDUCE, GREMOVE, and GPROJECT. GMAP, of course, is the procedure that actually draws maps. In spite of its name GREUCE does not reduce the resolution of your map; instead GREUCE calculates the density of each point. This can be used in a subsequent data step to reduce the resolution of a map by reducing the number of points and thereby reducing the size of your map data set. You can often reduce the size of your data set by as much as 80% without any visible decrement in map quality. This is a handy way to make your jobs run faster and your plots draw faster. If you are using a map data set that already includes density as a variable, then using PROC GREUCE is unnecessary. GREMOVE removes internal boundaries combining smaller segments into larger segments. GPROJECT projects map coordinates. The concept of projecting maps probably seems natural to a cartographer, but it is not obvious to laymen. Since latitude is measured
from east to west, an unprojected map drawn on a standard x-y grid is backward. That is, east is to the left and west to the right. Projecting a map fixes this and also adjusts for other distortions resulting from drawing a spherical world on a flat piece of paper.

Combining Map Data Sets

Figure 1, the US and Canada, is an example of a simple map that combines data sets: the STATES and CANADA4 map data sets. Appendix 1 contains the code used to create this map. In brief, this program selects points from the two data sets based on density, removes the internal boundaries of each using PROC GREATE, combines the two using a SET statement, projects the new map data set using PROC GPROJECT, and maps it with PROC GMAP. Note that in order to be combined, two map data sets must be in the same format—either both projected or both unprojected. If you are unsure whether a particular map data set is projected or unprojected, check the SAS/GRAPH manual.

Figure 2 shows a more unique map, one of the Atlantic coast from Nova Scotia to Georgia. Appendix 2 contains the code used to create this map. The first data step selects the desired provinces from the CANADA4 map data set, and assigns a fictitious state variable to be used as an id later. A second data step selects 16 states from the STATES data set. A third data step combines the provinces and states into one data set. Since the data are unprojected, PROC GPROJECT is needed to project the map with state as the id variable.

Subsetting Map Data Sets

Suppose you want a map of the San Francisco Bay area such as in Figure 3. Using the techniques discussed so far, you could select California from the STATES data set, but that is the closest you could get. The program in Appendix 3 makes just such a map. Since unprojected map data sets use latitude and longitude as their x and y coordinates, you can use an ordinary map to find the real latitude and longitude of the area you want and select those points only. This map uses only the points between 37 deg. 14 min. and 38 deg. 23 min. longitude, and between 120 deg. 19 min. and 124 deg. 20 min.

Figure 1. Canada and the U.S., two map data sets combined into one.
Figure 2. The Atlantic coast from Nova Scotia to Georgia, made using parts of two map data sets.

latitude. To do this, the latitude and longitude must be converted from degrees and minutes to radians by this formula:

\[ \text{Radians} = \pi \times \frac{\text{degrees}}{180} \]

One way of coding this in SAS terms is:

\[ \text{Radians} = \frac{(4\times\text{atan}(1)\times(\text{degrees}+(\text{minutes}/60)))}{180}; \]

Subsetting by \( x \) and \( y \) has the rather undesirable side effect of producing an incomplete line segment. If you printed this data with PROC GMAP, SAS would automatically complete the segment by connecting the last data point with the first data point and thereby creating the spurious and bizarre border shown in Figure 4. To avoid this you must somehow add data points to make a neat and meaningful closure. To do it, find the minimum and maximum, and first and last coordinates for latitude and longitude and use these to draw a box around the map. A data step was used to find the minimum and maximum \( x \) and \( y \) values, using a RETAIN to compare each observation to the last. Using the END= data set option, only the final observation was output.

PROC GPROJECT can be used to project this map, but in this case it results in the box being projected and therefore no longer appearing square. Also the effect of the projection just doesn't look right for a map of this small region. An alternative way of projecting the map uses a simple \( x \)-inversion routine. Simply use the minimum and maximum \( x \) coordinates to "flip" the map with this formula:

\[ x = \text{xmin} + \text{xmax} - x; \]

The resulting data can be mapped with PROC GMAP.

More ways to customize

Two more techniques should be mentioned because they can produce really effective maps. PROC GREPLAY allows you to place multiple graphics on a single page. For example you could show an entire state on one side of the page and then a blow-up of a major metropolitan area beside it. The ANNOTATE facility allows placement of special notations on
Figure 3. The San Francisco Bay Area with a box to complete the line segment.

Figure 4. The San Francisco Bay Area, the SAS software automatically completes the line segment.
your graphics, such name labels for cities or states, or pie charts showing levels of sales for each region.

Conclusions

Map data sets are no different than other SAS data sets except that they have certain predetermined variables. If you understand the role played by each of these variables: id, segment, x, y, and density; and the difference between projected and unprojected coordinates, you can creatively manipulate the data to create new maps. The mapping procedures will no longer be black boxes, but high level programming tools.

Afterword

I wish to thank Lee Watkins of The Johns Hopkins University, Homewood Academic Computing, and Michael W. Smith now at the Center for Molecular Genetics at the University of California San Diego who were both involved with an earlier project that eventually led to this paper.

The author may be contacted at 2310 Evenstar Lane #7, Davis, CA 95616, (916) 756-8434.

1. SAS, and SAS/GRAPH are registered trademarks of SAS Institute Inc., Cary, NC, USA.

Appendix 1

*---------------------------------------------------------------*
* Map of US and Canada *                                    *
*---------------------------------------------------------------*
* Select Canada by density and *                              *
* remove internal boundaries *                                *
*---------------------------------------------------------------*

data canada;
    set maps.canada4;
    if density lt 5;
    country = 'canada';

proc gremove data=canada out=canada;
    by country;
    id province;
*---------------------------------------------------------------*
* Select US by density and *                                   *
* remove internal boundaries *                                *
*---------------------------------------------------------------*

data usa;
    set maps.states;
    if density lt 5;
    country = 'usa';

Appendix 2

*---------------------------------------------------------------*
* Map of Atlantic coast *                                    *
*---------------------------------------------------------------*
* Select provinces *                                          *
*---------------------------------------------------------------*

data canada (drop = province);
    set maps.canada4;
    if density lt 5;
    if province = '12' or province = '13';
    if province = '12' then state = 92;
    else state = 93;

*---------------------------------------------------------------*
* Select states *                                             *
*---------------------------------------------------------------*

data usa;
    set maps.states;
    if density lt 5;
    if state = stfips('ME') or state = stfips('NH') or state = stfips('VT') or state = stfips('MA') or state = stfips('RI') or state = stfips('CT') or state = stfips('NY') or state = stfips('NJ') or state = stfips('PA') or state = stfips('DE') or state = stfips('MD') or state = stfips('VA') or state = stfips('WV') or state = stfips('NC') or state = stfips('SC') or state = stfips('GA');

*---------------------------------------------------------------*
* Combine data sets and project *                             *
*---------------------------------------------------------------*

data coast;
    set canada usa;
proc gproject data=coast out=coast;
   id state;
proc gmap data=coast map=coast;
   id state;
   choro state/nolegend discrete;
pattern v = e r = 20;
title;

Appendix 3

*------------------------------*
* Draw map of SF bay *
*------------------------------*
* Select California *
*------------------------------*
data sfbay (drop pi x1 x2 y1 y2);
   set maps.states;
   if state=stfips('CA ');
   pi 4*atan(1);
   x1 pi*(120+(19/60)/180;
   x2 pi*(124+(20/60)/180;
   y1 pi*(37+(14/60))/180;
   y2 pi*(38+(23/60))/180;
   if (x1 < x < x2) and (y1 < y < y2);
*------------------------------*
* Find min and max coordinates *
*------------------------------*
data minmax (keep=xmin xmax ymin ymax);
   set sfbay end=
elf;
   retain xmin xmax ymin ymax;
   if _n_ = 1 then do;
      xmin x; xmax x;
      ymin y; ymax y;
   end;
   if x < xmin then xmin = x;
   if x > xmax then xmax = x;
   if y > ymax then ymax = y;
   if _eof then do;
      put xmin= xmax =
ymin= ymax=;
      output:
end;
*------------------------------*
* Project by x-inversion *
*------------------------------*
data sfbay;
   if _n_ = 1 then set minmax:
   retain xmax xmin ymin ymax;
   set sfbay;
   x = xmin + xmax - x;
*------------------------------*
* Draw a box around the bay to *
* complete the line segment *
*------------------------------*
data sfbay (drop = xfirst xmax 
xmin ymax ymin xlast ylast);
   set sfbay;
   by segment;
   output sfbay;
   if first.segment then do;
      retain xfirst yfirst;
      xfirst = x; yfirst = y: