Creating Custom Line Types for State Boundaries with the GMAP Procedure and the SAS Annotate Facility.

Steven First
Systems Seminar Consultants

Abstract

SAS/Graph software through PROC GMAP and PROC GREMOVE provides an excellent way to produce choropleth maps for county, state, regional, or other boundaries. However because PROC GMAP draws adjacent polygons independently and as a result draws common segments between areas twice, it is not normally possible to choose different line types such as a dashed line for state boundaries, a dash-dot-dash pattern for international boundaries and so forth.

This paper will explore the combination of PROC GREMOVE, PROC GMAP, and the SAS annotate facility to produce US regional maps with state boundaries drawn only once in a dashed line type. The technique will produce maps closer to what cartographers prefer, and it can be applied to any regional type map.

Introduction

PROC GMAP along with SAS Institute provided maps provides an easy way to produce US maps with the state boundaries drawn in a solid line type. A typical PROC GMAP and it's output is shown below.

A Typical PROC GMAP program

```sas
data us;
input il15 state.2. region 2.;
data lines;
ALABAMA 1 03
ARIZONA 4 00
... (Not all states shown)
pattern1 c=blue v=blank;
pattern2 c=blue v=empty;
pattern3 c=green v=half30;
pattern4 c=green v=half45;
pattern5 c=red v=half60;
proc gmap data=us map=us.map;
choro state / notangle legend(legend);
ld state;
title basemap coded 'U.S. States';
r;
```

When showing PROC GMAP to a cartographer (who is too familiar with FORTRAN and not familiar enough with SAS), I was asked the following question: "Do you mean that SAS can't draw the standard line types between states"? He was referring to dashed patterns for state boundaries, dash-dot-dash lines for international boundaries and so on. I had to admit that to my knowledge SAS could not.

After thinking about it for a while and looking for some GMAP option that could change line types and not finding one, I wondered why as well. I concluded that because PROC GMAP draws adjoining polygons (states) independently, and thus traces common lines twice usually in different directions, even if a dashed line could be somehow used, it would probably end up closer to a solid line because of the multiple passes.

This problem becomes even more apparent when creating regional maps. PROC GREMOVE easily removes common boundaries between states when creating regional maps producing a nice regional map, but no trace remains of the interior lines that were deleted.

While working on a large mapping project for a national retailer, one request was to produce a regional map and not only remove interior lines from the regional boundaries, but to also draw those regional boundaries in a bold wide solid line and to include state boundaries in a dashed pattern. Additionally I needed to label both the regional names and also the state names. The
client had a hand-drawn map which they wanted to duplicate with PROC GMAP for marketing displays.

Since I had used the SAS/GRAPH annotate facility extensively, labeling the state and regional labels was no big problem. The annotate facility can also draw different line types, so I decided to try and use it to draw those interior lines.

A pseudo-regional map

Pages 251-252 in the SAS/GRAPH User's Guide show a simple program to create the US regional map shown in Figure 2. The polygon area is a state boundary. By assigning a numeric region number to each state and also assigning a SAS format range and pattern to each response, the final map draws all states in a region using the same pattern and color. This is not a true regional map, but just a collection of adjacent states that happen to be drawn in the same pattern, thus it appears to be a regional map.

data us;
  input state 2. region 2.
  datalines;
  ALABAMA  1 0
  WYOMING 56 02
;proc format;
  value region
    0 = 'Southwest'
    1 = 'West'
    2 = 'South'
    3 = 'Northeast'
    4 = 'Midwest'
  run;
  pattern1 color = blue v = empty;
  pattern2 color = white;
  pattern3 color = green v = long;
  pattern4 color = green v = short;
  pattern5 color = red v = middle;
  pattern6 color = black v = middle;
proc gmap data=us
  merge maps.us
  id state;
  format region regfmt.;
title 'U.S. Regions';
run;
Figure 2 A pseudo-regional map

A true regional map.

Pages 327-328 in the SAS/GRAPH User's Guide show a program similar to the one below which uses PROC GREMOVE to produce a true regional map like the one shown in Figure 3.

data newus;
  merge maps.us
  newus
  by state;
  n+1;
rerun;
proc sort data=newus
  by region n;
run;
proc gremove data=newus
  map=regions;
run;
proc gmap data=regions
  map=regions;
  id region;
  format region regfmt.;
title 'U.S. Regions';
run;
Figure 3 A true regional map

What my client wanted was a combination of the two maps, plus annotation of labels. A sample of the final desired output is shown in figure 4.
Building a regional map with interior lines.

The first step is to create the regional map shown above in Figure 3 and save it. The program above called it REGIONS, but it probably should have been a permanent SAS dataset.

Next we need to somehow determine which line segments from the MAPS.US map are contained in adjacent states. This is essentially the same type of logic PROC GREMOVE must go through, but instead of discarding the interior lines, we will draw them. It is important to note that we must not just discard common points, because a corner point may exist in up to four states which might not all be in the same region. Instead we have to look at a point and compare it with the next point in the map and consider them together to be a line and try to see if that line exists in two adjoining states.

To complicate things adjoining states don’t start defining points at the same place, and they may not even be drawn in the same direction, since PROC GMAP can handle polygons drawn in either a clockwise or counter-clockwise direction. Another luxury that PROC GMAP provides is to join the very last point in a segment to the first automatically. Since we are in effect drawing lines via annotate instead, we will be responsible to get back to the starting point. This task is not simplified by the fact that many points will be discarded along the way (including perhaps the starting point).

To simplify the presentation the map shown in Figure 5 will include just four states, Arizona, Colorado, New Mexico, and Utah. This represents the worst case since a single point is included in all four states. The other points will exist in one, two, or three other states. These points will be contained in as many as four line segments drawn in different directions. The real trick here is to draw each line just once.

A print of the map dataset is shown below.

<table>
<thead>
<tr>
<th>OBS</th>
<th>SEGMENT</th>
<th>STATE</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>-0.19426</td>
<td>-0.0061</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>-0.23225</td>
<td>-0.0051</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>-0.27674</td>
<td>-0.0051</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>-0.27483</td>
<td>-0.0051</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>5</td>
<td>-0.27269</td>
<td>-0.0051</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>6</td>
<td>-0.27084</td>
<td>-0.0051</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>7</td>
<td>-0.27293</td>
<td>-0.0051</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>8</td>
<td>-0.26686</td>
<td>-0.0051</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>9</td>
<td>-0.26795</td>
<td>-0.0051</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>10</td>
<td>-0.26105</td>
<td>-0.0051</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>11</td>
<td>-0.26557</td>
<td>-0.0051</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>12</td>
<td>-0.26364</td>
<td>-0.0051</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>13</td>
<td>-0.26333</td>
<td>-0.0051</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>14</td>
<td>-0.26442</td>
<td>-0.0051</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>15</td>
<td>-0.25518</td>
<td>-0.0051</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>16</td>
<td>-0.25869</td>
<td>-0.0051</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>17</td>
<td>-0.25322</td>
<td>-0.0051</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>18</td>
<td>-0.25344</td>
<td>-0.0051</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>19</td>
<td>-0.25055</td>
<td>-0.0051</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>20</td>
<td>-0.24912</td>
<td>-0.0051</td>
</tr>
<tr>
<td>21</td>
<td>4</td>
<td>21</td>
<td>-0.20913</td>
<td>-0.0051</td>
</tr>
<tr>
<td>22</td>
<td>4</td>
<td>22</td>
<td>-0.18232</td>
<td>-0.0051</td>
</tr>
<tr>
<td>23</td>
<td>4</td>
<td>23</td>
<td>-0.18096</td>
<td>-0.0051</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>24</td>
<td>-0.17280</td>
<td>-0.0051</td>
</tr>
<tr>
<td>25</td>
<td>4</td>
<td>25</td>
<td>-0.16768</td>
<td>-0.0051</td>
</tr>
<tr>
<td>26</td>
<td>4</td>
<td>26</td>
<td>-0.16013</td>
<td>-0.0051</td>
</tr>
<tr>
<td>27</td>
<td>4</td>
<td>27</td>
<td>-0.18222</td>
<td>-0.0051</td>
</tr>
<tr>
<td>28</td>
<td>4</td>
<td>28</td>
<td>-0.18284</td>
<td>-0.0051</td>
</tr>
<tr>
<td>29</td>
<td>4</td>
<td>29</td>
<td>-0.15907</td>
<td>-0.0051</td>
</tr>
<tr>
<td>30</td>
<td>4</td>
<td>30</td>
<td>-0.15679</td>
<td>-0.0051</td>
</tr>
<tr>
<td>31</td>
<td>4</td>
<td>31</td>
<td>-0.15826</td>
<td>-0.0051</td>
</tr>
<tr>
<td>32</td>
<td>4</td>
<td>32</td>
<td>-0.15824</td>
<td>-0.0051</td>
</tr>
<tr>
<td>33</td>
<td>4</td>
<td>33</td>
<td>-0.14644</td>
<td>-0.0051</td>
</tr>
<tr>
<td>34</td>
<td>4</td>
<td>34</td>
<td>-0.14624</td>
<td>-0.0051</td>
</tr>
<tr>
<td>35</td>
<td>4</td>
<td>35</td>
<td>-0.14564</td>
<td>-0.0051</td>
</tr>
<tr>
<td>36</td>
<td>4</td>
<td>36</td>
<td>-0.16232</td>
<td>-0.0051</td>
</tr>
<tr>
<td>37</td>
<td>4</td>
<td>37</td>
<td>-0.16918</td>
<td>-0.0051</td>
</tr>
<tr>
<td>38</td>
<td>4</td>
<td>38</td>
<td>-0.18232</td>
<td>-0.0051</td>
</tr>
<tr>
<td>39</td>
<td>4</td>
<td>39</td>
<td>-0.20569</td>
<td>-0.0051</td>
</tr>
<tr>
<td>40</td>
<td>4</td>
<td>40</td>
<td>-0.23410</td>
<td>-0.0051</td>
</tr>
<tr>
<td>41</td>
<td>4</td>
<td>41</td>
<td>-0.22098</td>
<td>-0.0051</td>
</tr>
<tr>
<td>42</td>
<td>4</td>
<td>42</td>
<td>-0.19596</td>
<td>-0.0051</td>
</tr>
<tr>
<td>43</td>
<td>4</td>
<td>43</td>
<td>-0.19870</td>
<td>-0.0051</td>
</tr>
<tr>
<td>44</td>
<td>4</td>
<td>44</td>
<td>-0.17280</td>
<td>-0.0051</td>
</tr>
<tr>
<td>45</td>
<td>4</td>
<td>45</td>
<td>-0.18083</td>
<td>-0.0051</td>
</tr>
</tbody>
</table>

Steps required to draw the interior lines:

Step 1. Pass through the dataset and save the first point in a SEGMENT, and add it to the end. Also add a variable called N indicating
original point order so we can get back into correct order later.

DATA STATES;
SET AZCONMUT;
BY STATE SEGMENT;
IF FIRST SEGMENT THEN DO:
SAVEX X;
SAVEX STATE;
RETAIN SAVEX SAVEST SAVESEG SAVESAVE;
DROP SAVEX SAVEST SAVESEG SAVESAVE;
END;
OUTPUT;
IF LAST SEGMENT THEN
00;
SAVEX X;
SAVEX STATE;
SAVESEG SEGMENT;
SAVESAVE SAVEST;
DROP SAVEX SAVE SEG SAVESAVE;
END;
14;MM.
If LAST. SEGMENT THEN
&0; .. REPEAT FIRST. X TO END;
X=SAVEX;
Y=SAVE;
SEGMENT=SAVESEG;
STATE=SAVEST;
11+1;
OUTPUT;
END;
KEEP STATE SEGMENT X Y N,
RUN;
PROC PRINT DATA=STATES; TITLE 'STATES BEFORE SORT';RUN;

Step 2. Delete the points obviously not in two states. Sort the map dataset into X, Y, STATE, and SEGMENT order. Pass through the dataset. If a X Y pair appears only once in the map, it obviously cannot be in two states, so that point is discarded. This step could be inserted at the beginning of step 3 and save a pass of the dataset.

PROC SORT DATA=STATES;
BY X Y STATE SEGMENT;
RUN;
DATA DUPS;
SET STATES;
BY X Y STATE SEGMENT;
IF FIRST Y AND LAST Y THEN DELETE; /* POINT IN ONLY 1 STATE */
RUN;
PROC PRINT DATA=DUPS; TITLE 'POINTS IN AT LEAST TWO STATES';RUN;

Step 3. Pass through the dataset and assign each point to a line name made up of the two adjoining state numbers and the corresponding segment numbers.

Figure 6 Line segments and states

For example points on the line between state 04 (Arizona) and state 49 (Utah) would be assigned to linelname 04001-49001. Digits 1-2 and 7-8 are the state numbers and the remaining digits are the segment numbers. The key to eliminating duplicate lines is to always use the smaller numbered state first and disregard the linelname which would be labeled 49001-04001. This is accomplished with the following program.

Three arrays are created to store the N variable, the STATE, and SEGMENT for up to four times that a point could be used. A variable called COUNT counts how many states a point is used in. When the last occurrence of a point is detected, two loops output the points and their N and LINENM values. Since the inner loop only loops through higher numbered states than the outer loop, each point is assigned only to one segment. A program listing and the print of the resulting dataset appears here.

DATA DUPS;
SET DUPS;
BY X Y STATE SEGMENT;
LENGTH LINEM $ 11;
ARRAY NARRY(*) 1-11;
ARRAY STATE(*) 1-11;
ARRAY SEGMENT(*) 1-11;
IF FIRST Y THEN COUNT=0;
COUNT=
STATE(COUNT)=STATE;
SEGMENT(COUNT)=SEGMENT;
NARR(COUNT)=NARR;
IF COUNT = 1;

604
DO J = 1 + 1 TO COONT;
  PUT (STARRY<J);Z2.) I!PUHSGAARY{J};n.) 11
  PUT (STARRY<J) ,Zl:.)
END;

1.11"

END;

END;

PROC PRINT DATA=DUP2;
  TITLE 'DUP2 ';RUN;

ANSTLINE

PROC PRINT DATA=ANSTLINE;
  TITLE 'ANSTLINE';RUN;

Step 4. The final processing step is to sort the dataset into LINEM order and generate MOVE and DRAW annotate functions. The LINE variable in this example is set to 8 which is as close to the cartographer's state boundary as I could get. It certainly could be set to any of the 32 line types SAS/GRAPH supports. One final item to note is that we have discarded several points that are no longer needed. If there is such a "gap" we need to move to the next point instead of drawing to it.

PROC sort DATA=DUP2; BY LINN M;RUN;

DATA ANSTLINE;
  SET DUP2;
  BY LINM;
  IF FIRST.LINM AND LAST.LINM THEN DELETE;
  RETAIN LINM *1;
  RETAIN XYS *2;
  RETAIN TESTS *2;
  KEEP LINM XYS TESTS;
  IF FIRST.LINM THEN /* BEGINNING OF NEW LINE*/
DO;
  FUNCTION=MOVE *;
  OUTPUT;
  SAVE=X;
  SALEV=Y;
  SALEV=W;
  END;
ELSE
  IF LAST=1 OR THEN /* IS IT NEXT POINT FROM */
DO;
  FUNCTION=MOVE *;
  OUTPUT;
  SAVE=X;
  SALEV=Y;
  SALEV=W;
  END;

PROC GMAP DATA=REGIONS;
  CHORD REGION / DISCRETE SOLID C=ANSTLINE; IN REGION;
  TITLE 'SWIS m 59001 ARIZONA, COLORADO, NEW MEXICO, UTAH';
  PATTERN C=BLACK V=EMPTY R=4;
RUN;

Step 5. Using the above dataset as an annotation dataset to PROC GMAP produces the final graph for our four states as shown here.
Running the entire country through the program and using the annotate dataset with the regional map shown earlier produces a regional map with state lines drawn by the Annotate facility.

The final product

Additional annotation as described in the SAS/GRAPH users guide can be used to produce the final Map.

Acknowledgments:

Special thanks to: Steven Subichin Miller Brewing Company
Richard Langston SAS Institute Inc.
Tom Miron Systems Seminar Consultants.

SAS and SAS/GRAPH are registered trademarks of SAS Institute Inc.