

Programmarium and Atlas of Statistical Cartography: How to Find/Show Your Data with PROC GMAP

LeRoy Bessler, Miller Brewing Company

Abstract and Introduction

Most SAS® software sites have large amounts of data which include geographic unit area designators (in the USA, state code is probably the commonest). Though this data can be reported in various tabular formats, a geographic effect (such as a proximity effect) will not easily be revealed without a statistical map. Also, for presentation, a visual image is more interesting than a listing.

But the map can be supplemented with detail look-up data. The detail can be a complete list in key sequence, a ranking report based on response level, or just a Top NN List of the NN most significant (i.e., highest response) geographic areas. A Top 10 table is concise enough to use as a slide. Better yet is to inlay the detail on the respective geographic areas of the map.

This introductory tutorial provides complete program code to build well-designed SAS/GRAPH® statistical maps. I go beyond the traditional tutorial in presenting innovations: (a) Automatic Rationale-based Response Range Assignment (ARbRRA)--the solution to what I call The Four Color Map Problem; and (b) Subichin's blanking for map annotation--a feature long needed in SAS/GRAPH. ARbRRA eliminates the need to accept default assignment of response ranges or to inspect a sorted list of all responses to "hand pick" ranges, which may be arbitrary anyhow. Blanking assures annotation is always readable, even when area fills are grey shades or dark colors.

Sample code is intended to be readily adapted--just point it at your data, customize a few details, and you're done. Though the examples use the USA map, the principles apply to county or international maps as well.

No knowledge of SAS/GRAPH is required. But annotation details are only commented in the code, not explained below.

GOPTIONS Statement

The GOPTIONS statement specifies the device driver, device-related parameters, physical size of the graph display area, etc.

I used an AFP-compatible, black-and-white laser printer, capable of printing grey shades. In the programs, you will see familiar color names. Software translates them into grey shades. Upon request, I can furnish details of GOPTIONS, GDDM nickname, JCL, etc., and comments as to what works well, and what doesn't. GOPTIONS are not shown here.

GMAP Types

PROC GMAP supports SURFACE maps, BLOCK maps, PRISM maps, and CHORO (i.e., choropleth) maps. The first three types are 3D, and have serious interpretability and useability problems. The extra dimension (for special effect) interferes with, rather than enhances, communication.

PROC GMAP Statement

```
PROC GMAP
  MAP=
  DATA=
  ANNOTATE=
  GOUT=
  ALL;
```

MAP= specifies the map data set.

DATA= specifies the response data set.

ANNOTATE= specifies the annotate data set.

GOUT= specifies the output graphics catalog (not covered here).

ALL indicates that every geographic area must be drawn, even if there is no response value for it.

ID Statement

The ID statement is a list of one or more variable names, used to identify each distinct geographic area. The ID variable(s) must be present in both the map and response data sets.

CHORO Statement

```
CHORO respvar
  ANNOTATE=
  CEMPTY=
  COUTLINE=
  CTEXT=
  DISCRETE
  LEGEND=LEGENDnn
  NOLEGEND
  LEVELS=
  MIDPOINTS=
  MISSING
  XSIZE=
  YSIZE=
  NAME=
  DESCRIPTION= ;
```

NAME and DESCRIPTION are used when the output map is routed to a graphics catalog (not covered here).

XSIZE= and YSIZE= can be optionally used to size the map. (GOPTIONS HSIZE and VSIZE are used to size the entire graphic display area, while XSIZE and YSIZE affect only the map within that available area.)

ANNOTATE= specifies the (optional) annotate data set.

LEGENDnn identifies the (optional) custom LEGEND statement.

COUTLINE= specifies outline color for non-empty geographic areas. Default outline is same color as area fill color. I like BLACK.

CEMPTY= specifies outline color for empty geographic areas. You must also use the ALL option on the PROC GMAP statement, or empty areas (i.e., those for which there is no response value) are not drawn. If you use the ALL option, but not CEMPTY=, then the first entry in the GOPTIONS COLORS= list or device driver colors list is used.

CTEXT= specifies the color to be used for legend text. You can instead specify color on LABEL= and VALUE= options of the LEGEND statement. If you do neither, SAS/GRAPH uses the CTEXT= value in a GOPTIONS statement (if present), or the first entry in the COLORS= or colors list. CTEXT= in a GOPTIONS statement specifies the color to be used for all text on the GMAP output for which there is no color specified by other means.

DISCRETE specifies that a numeric response variable is to be treated as a discrete, rather than a continuous, variable. The default action is then to assign a different fill pattern and color combination to each value of the response variable. To group the responses into ranges, a user-written format must be used.

LEVELS= specifies the number of response levels (i.e., ranges) to be used for a continuous (i.e., numeric) response variable. SAS/GRAPH determines the range boundaries.

If the response variable is numeric, but neither the LEVELS= option nor the DISCRETE option is used, then SAS/GRAPH picks the number of levels based on an algorithm that depends on the number of distinct ID variable values.

LEVELS is mutually exclusive with DISCRETE and MIDPOINTS.

MIDPOINTS= specifies the response levels for numeric values as a list of numbers, a from ... to ... by ... expression, or both. The case of character response variables is not covered here.

Statements Not Covered

Although used, space does not permit explanation of TITLE or LEGEND statements. As an aside, I mention that use of a block effect or drop shadow for legends serves no useful purpose, only cluttering the image, and needlessly prolonging graph creation- and print-time.

PATTERN Statements

I recommend PATTERN statements to cover the number of response levels or ranges. I do not recommend default handling of fill patterns and fill colors.

For each level or range, define a statement of the form

PATTERNnn V=MSOLID COLOR=colname;

If the map has four response levels or ranges, you need statements PATTERN1, PATTERN2, PATTERN3, and PATTERN4, each with a distinct color. If you desire WHITE and it is not on the device palette, use V=MEMPTY (then color doesn't matter).

Figure 1: SAS/GRAPH Defaults Unacceptable

Figure 1 is an unacceptable map of a real data set, using (see program in Exhibit 1) PATTERN statements, a COUTLINE, and otherwise a default invocation of PROC GMAP.

The ranges chosen (their midpoints are shown in the map legend) by the PROC GMAP default algorithm are adversely influenced by two values, viz., 736 and 447; all other values are below 179.

Figure 2: Four Color Map Problem Solved

Even without an extreme result as Figure 1, it is better to make a deliberate choice of ranges, based on a rationale. In principle, that requires you to have knowledge of the data distribution. But first doing a PROC SORT and PROC PRINT so data can be inspected is inconvenient, and can result in an arbitrary decision anyhow.

In a prior paper ("Effective and Efficient Information Delivery for Executive Management", in *Proceedings of the Seventeenth Annual SAS Users Group International Conference*, SAS Institute Inc. (Cary, N.C.), 1992), I emphasized that, typically, a small subset of the observations account for a large majority, or even almost all, of the total response. A Top 10 or Top NN Report (i.e., some one-page-or-less report) usually suffices, often accounting for 80% to 99% of the total response. In this data set, the Top 10 states account for 66.4% of attendance. (Over the last five years, 63.1%.) With 50 states and DC, the Top 10 states are by definition above the 80th percentile.

Regardless of the specific choices, it is natural to break up the total range based on percentiles. I generally like the 20th percentile, median (the 50th percentile), and 80th percentile. The resulting four ranges can be characterized as very low, below median, above median, and very high. I would use more ranges if there were many more geographic areas to differentiate (e.g., all counties in the USA), but would always include the median as a break point. (Other mapmakers might use mean and standard deviation to develop ranges.)

Percentile-based ranges create a talking point for the map, as opposed to relying on software defaults. The solution uses "software intelligence". For other examples, see my paper "Software Intelligence: Applications That Customize Themselves", in *Proceedings of the Eighteenth Annual SAS Users Group International Conference*, SAS Institute Inc. (Cary, N.C.), 1993.

The program in Exhibit 2 shows how to let software automatically solve the Four-Color Map Problem. Give the program data, it sets four ranges automatically. You need only pick four colors, and supply title text. I call this technique Automatic Rationale-based Response Range Assignment (ARbRRA).

If desired, the legend could display the actual numeric ranges, instead of the text strings I used—modify the PROC FORMAT VALUE statement by replacing the right-hand side of each assignment with the left-hand side enclosed in double quotes.

Figure 3: Simple Annotation, Without Blanking

This map (Figure 3, done with the program in Exhibit 3) displays the response values. The POSITION annotate variable used in the program is explained on pages 521-524 of *SAS/GRAPH Software Reference, Version 6, First Edition, Volume 1*. The code for DATA Step ANNOVALU is an improved version of that on page 10 of *SAS/GRAPH Software: Map Data Sets (SAS Technical Report P-196)*. But I am not satisfied with this result.

Since not everyone knows each state name just by shape and relative location on the map, it could be "nice to have" that identification. One solution is two-line annotation--on the upper line the State abbreviation, and on the lower line the Count. Better yet is three-line annotation, on the bottom line the Rank based on Count; or you might show Percent of Whole.

Now a "must have". Still missing from SAS/GRAPH is "blanking". Behind each annotation we need a rectangle of white space within the solid color (or grey shade) in the geographic area. Text is often unreadable on a background of dark or intense color, and especially on some grey shades.

How to do blanking was reported by Steven J. Subichin in "Enhanced Useability for Annotation on SAS/GRAPH Maps", in *WISAS Proceedings, Volume 5, June Issue*, South Milwaukee, Wis.: WISAS Inc., 1993.

Figure 4: Blanking and Enhanced Annotation

This map (Figure 4, done with the program in Exhibit 4) illustrates three-line annotation, and Subichin's blanking. It uses an extension of the program presented in his paper.

The program in Exhibit 4 includes my recommended adjustments to the state center coordinates from MAPS.USCENTER.

Also, the map incorporates a star to highlight the conference location. (Instead, the city marker could have been the city name, or "SUGI 14".) See DATA step CITYSTAR for how to do this. In Appendix A is a program to list the GMAP US cities. For a city not listed, pick the nearest city that is listed, and adjust coordinates.

Of course, DATA step ANNODATA could be simplified. E.g., for two-line annotation, use POSITION values for 1/2 cell above and 1/2 cell below, and consider making the boxes smaller.

Notices

SAS and SAS/GRAPH are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. " denotes USA registration.

All SAS code in this paper was tested, but can only be presented "as is". Any adopted by you must be tested by you.

Author

Dr. LeRoy Bessler
Miller Brewing Company
P.O. Box 482
Milwaukee, Wisconsin 53201-0482, USA
Telephone 414-931-2773

```
DATA USCITIES;
SET MAPS.USCITY;
STATENAME = FIPNAMEL(STATE);
IF CAPITAL = 'N' THEN CAPITAL = ' ';
RUN;
PROC PRINT DATA=USCITIES UNIFORM LABEL;
PAGEBY STATE;
BY STATE STATENAME;
ID STATE STATENAME;
VAR CITY X Y CAPITAL POP;
LABEL STATE='ID';
LABEL STATENAME='State';
LABEL CAPITAL='Capital?';
TITLE 'SAS/GRAPH MAPS.USCITY File';
RUN;
```

Appendix A: Program to List GMAP US Cities

```

DATA RESPONSE; /* input data */
INFILE ddname;
INPUT STATEABB $ VALUE;
STATE = STFIPS(STATEABB); /* convert two alphabetic characters to numeric ID */
RUN;

PATTERN1 V=MEMPTY C=BLACK;
PATTERN2 V=MSOLID C=GRAY;
PATTERN3 V=MSOLID C=GOLD;
PATTERN4 V=MSOLID C=GREEN;
PATTERN5 V=MSOLID C=DAB;
PATTERN6 V=MSOLID C=BLUE;

TITLE . . . ;

PROC GMAP MAP=MAPS.US DATA=RESPONSE ALL; /* map */
ID STATE;
CHORO VALUE / COUTLINE=BLACK;
RUN;
QUIT;

```

Exhibit 1: Program for Figure 1

```

DATA RESPONSE; /* input data */
. . .

PROC UNIVARIATE DATA=RESPONSE NOPRINT PCTLDEF=2; /* get min, max, percentiles */
VAR VALUE;
OUTPUT OUT      = STATS
      MIN      = MIN
      MAX      = MAX
      PCTLPTS  = 20 50 80
      PCTLPRE  = PCTL;
RUN;

DATA _NULL_; /* boundaries for response ranges as global variables */
SET STATS;
%GLOBAL MIN B1 B2 B3 B4 B5 B6 MAX;
CALL SYMPUT('MIN',TRIM(LEFT(MIN      )));
CALL SYMPUT('B1',TRIM(LEFT(PCTL20    )));
CALL SYMPUT('B2',TRIM(LEFT(PCTL20 + 1)));
CALL SYMPUT('B3',TRIM(LEFT(PCTL50    )));
CALL SYMPUT('B4',TRIM(LEFT(PCTL50 + 1)));
CALL SYMPUT('B5',TRIM(LEFT(PCTL80    )));
CALL SYMPUT('B6',TRIM(LEFT(PCTL80 + 1)));
CALL SYMPUT('MAX',TRIM(LEFT(MAX      )));
RUN;

PROC FORMAT; /* build format to be used by PROC GMAP for ranges & legend text */
VALUE VALUEFMT &MIN - &B1 = 'Up To 20th Pctl'
              &B2 - &B3 = 'Up To Median '
              &B4 - &B5 = 'Up To 80th Pctl'
              &B6 - &MAX = 'Above 80th Pctl';
RUN;

PATTERN1 V=MSOLID C=GRAY;
PATTERN2 V=MSOLID C=GOLD;
PATTERN3 V=MSOLID C=GREEN;
PATTERN4 V=MSOLID C=DAB;
LEGEND1 LABEL=(F=SIMPLEX) SHAPE=BAR(3.4,0.8) DOWN=1;
TITLE . . . ;

PROC GMAP MAP=MAPS.US DATA=RESPONSE ALL; /* map */
FORMAT VALUE VALUEFMT.;
ID STATE;
CHORO VALUE / COUTLINE=BLACK DISCRETE LEGEND=LEGEND1;
RUN;
QUIT;

```

Exhibit 2: Program for Figure 2

```

. (get input data
. create global variables
. determine min, max, & percentiles
. create format for response ranges & legend)

PROC SORT DATA=RESPONSE OUT=TOANNO;
BY STATE;
RUN;

PROC SORT DATA=MAPS.USCENTER OUT=CENTERS; /* get coords. of the state centers */
BY STATE DESCENDING OCEAN; /* if OCEAN = 'Y', must precede OCEAN = 'N' */
RUN;

DATA MERGED; /* merge values to annotate and coordinates of the state centers */
MERGE TOANNO(IN=INRESP) CENTERS; BY STATE;
IF INRESP;
RUN;

DATA ANNODATA; /* put value at center of each state, except coastal states */
SET MERGED;
RETAIN FLAG;
IF N_ = 1 THEN FLAG = 0;
XSYS = '2';
YSYS = '2';
WHEN = 'A';
IF FLAG = 0
THEN DO;
SIZE = 1.00;
COLOR = 'BLACK';
STYLE = 'SIMPLEX';
FUNCTION = 'LABEL';
IF FIPSTATE(STATE) = 'HI' THEN Y = Y + .025; /* for Hawaii, shift label up */
TEXT = LEFT(VALUE);
IF OCEAN = 'N' THEN POSITION = '5'; /* locate at center */
ELSE POSITION = '6'; /* in this case, also justify left */
OUTPUT;
END;
ELSE DO;
FUNCTION = 'DRAW';
SIZE = 1.00; /* arrow thickness */
OUTPUT;
FLAG = 0;
END;
IF OCEAN = 'Y'
THEN DO;
FUNCTION = 'MOVE';
X = X - .005;
OUTPUT;
FLAG = 1;
END;
RUN;

. (PATTERNS, LEGEND, TITLE)

PROC GMAP MAP=MAPS.US DATA=RESPONSE ANNO=ANNODATA ALL; /* map
. . .

```

Exhibit 3: Program for Figure 3

```

. (get input data
. create global variables
. determine min, max, & percentiles
. create format for response ranges & legend)

```

```

PROC SORT DATA=RESPONSE OUT=RANKED;
BY DESCENDING VALUE;
RUN;

```

```

DATA RANK;
SET RANKED;
RANK = _N_;
RUN;

```

```

PROC SORT DATA=RANK OUT=RANK;
BY STATE;
RUN;

```

```

PROC SORT DATA=RESPONSE OUT=TOANNO;
BY STATE;
RUN;

```

```

PROC SORT DATA=MAPS.USCENTER OUT=CENTERS; /* get coords. of the state centers */
BY STATE DESCENDING OCEAN; /* if OCEAN = 'Y', must precede OCEAN = 'N' */
RUN;

```

```

DATA CHANGES; /* changes empirically selected by LeRoy Bessler */
INFILE CARDS; /* to be applied to maps.uscenter coords as of SAS Release 6.08 */
INPUT STATE OCEAN $ CHANGEX CHANGEY;

```

```

CARDS;
01 N -0.0037 +0.0000 AL
02 N +0.0050 +0.0000 AK
04 N -0.0030 +0.0000 AZ
05 N -0.0025 +0.0037 AR
08 N +0.0000 +0.0020 CO
09 Y -0.0140 -0.0050 CT
10 Y -0.0025 -0.0060 DE
11 Y -0.0100 -0.0200 DC
12 N +0.0040 -0.0060 FL
15 N +0.0050 +0.0250 HI
16 N -0.0050 +0.0000 ID
17 N -0.0015 +0.0050 IL
18 N -0.0013 +0.0000 IN
19 N +0.0000 -0.0020 IA
20 N +0.0000 -0.0015 KS
21 N +0.0000 +0.0035 KY
22 N -0.0015 +0.0080 LA
23 N -0.0025 +0.0060 ME
24 Y +0.0250 -0.0200 MD
25 Y -0.0050 +0.0015 MA
26 N +0.0010 -0.0030 MI
28 N -0.0013 +0.0000 MS
29 N -0.0037 +0.0000 MO
30 N +0.0070 -0.0005 MT
33 Y -0.0010 +0.0150 NH
34 Y +0.0275 -0.0200 NJ
36 N +0.0025 +0.0015 NY
37 N +0.0000 +0.0035 NC
38 N +0.0000 +0.0020 ND
39 N -0.0050 +0.0025 OH
42 N +0.0000 +0.0045 PA
44 Y +0.0000 -0.0100 RI
45 N +0.0065 +0.0015 SC
46 N +0.0000 +0.0020 SD
47 N +0.0000 +0.0025 TN
50 Y -0.0500 +0.0400 VT
51 N +0.0000 -0.0015 VA
53 N +0.0100 +0.0000 WA
54 N -0.0020 +0.0010 WV
55 N +0.0010 +0.0000 WI
56 N -0.0050 +0.0020 WY

```

```

;
RUN;

```

```

PROC SORT DATA=CHANGES OUT=CHANGES; /* sort */
BY STATE DESCENDING OCEAN;
RUN;

DATA CENTERS; /* shift of coordinates for states needing it */
MERGE CENTERS CHANGES(IN=CHANGE);
BY STATE DESCENDING OCEAN;
IF CHANGE THEN DO;
  X = X + CHANGEX;
  Y = Y + CHANGEY;
END;
RUN;

DATA MERGED; /* merge rank & response value with coords of the state center */
MERGE RANK TOANNO(IN=INRESP) CENTERS; BY STATE;
IF INRESP;
RUN;

DATA ANNODATA; /* put all text at center of each state, except coastal states */
SET MERGED;
RETAIN FLAG;
IF _N_ = 1 THEN FLAG = 0;
XSYS = '2';
YSYS = '2';
WHEN = 'A';
IF FLAG = 0
  THEN DO;
    SIZE = 0.70;
    COLOR = 'BLACK';
    STYLE = 'SIMPLEX';
    FUNCTION = 'LABEL';
    TEXT = FIPSTATE(STATE);
    IF OCEAN = 'N'
      OR
      FIPSTATE(STATE) = 'VT'
      THEN POSITION = '2';
    ELSE POSITION = '3';
    OUTPUT;
    TEXT = LEFT(VALUE);
    IF OCEAN = 'N'
      OR
      FIPSTATE(STATE) = 'VT'
      THEN POSITION = '5';
    ELSE POSITION = '6';
    OUTPUT;
    TEXT = 'R=' || TRIM(LEFT(RANK));
    IF OCEAN = 'N'
      OR
      FIPSTATE(STATE) = 'VT'
      THEN POSITION = '8';
    ELSE POSITION = '9';
    OUTPUT;
  END;
ELSE DO;
  FUNCTION = 'DRAW';
  SIZE = 1.00; /* arrow thickness */
  OUTPUT;
  FLAG = 0;
END;
IF OCEAN = 'Y'
  THEN DO;
    FUNCTION = 'MOVE';
    IF FIPSTATE(STATE) = 'VT'
      THEN X = X - .005;
    ELSE Y = Y - .020;
    OUTPUT;
    FLAG = 1;
  END;
END;
RUN;

```

```

DATA CITYSTAR; /* city marker annotation */
SET MAPS.USCITY;
IF CITY = 'San Francisco'; /* get coordinates for San Francisco */
XSYS = '2'; /* do not change */
YSYS = '2'; /* do not change */
WHEN = 'A'; /* do not change */
POSITION = '5'; /* adjust, if desired, for limited repositioning */
SIZE = 1.00; /* adjust size, if needed */
STYLE = 'MARKER'; /* font comprised of markers (see also MARKERE) */
COLOR = 'BLACK'; /* change color, if desired */
TEXT = 'V'; /* this marker is a star */
X = X - 0.010; /* for this city, shift left for better visibility */
RUN;

DATA ANNODATA; /* append city marker to other annotation data */
SET ANNODATA CITYSTAR;
RUN;

PROC FREQ DATA=MAPS.US; /* count the points on each segment of the state */
TABLES STATE*SEGMENT / OUT=COUNTS NOPRINT;
RUN;

PROC SORT DATA=COUNTS; /* order state & segment ID by decreasing point-counts */
BY STATE DESCENDING COUNT;
RUN;

DATA BIGSEG; /* output 1 record per state with segment ID of biggest segment */
SET COUNTS;
BY STATE DESCENDING COUNT;
IF (FIPSTATE(STATE) ^= 'MI' AND FIRST.STATE)
OR /* for Michigan, biggest segment has smaller point-count */
(FIPSTATE(STATE) = 'MI' AND NOT FIRST.STATE);
RUN;

DATA DELOCEAN; /* won't need boxes for Hawaii & in ocean */
SET CENTERS;
IF FIPSTATE(STATE) = 'HI' THEN DELETE;
IF OCEAN = 'Y' | LAG(OCEAN) = 'Y' THEN DELETE;
/* above IF because states annotated in the ocean have 2 sets of coordinates */
RUN;

DATA INNER; /* keep state, segment ID, & coordinates only for states with box */
MERGE BIGSEG DELOCEAN(IN=NOOCEAN); BY STATE;
IF NOOCEAN;
KEEP STATE SEGMENT X Y;
RUN;

DATA BOXES(KEEP=STATE SEGMENT X Y); /* make boxes for inner states */
SET INNER;
RX=X; RY=Y; /* save X & Y coordinates */
X=.; Y=.; OUTPUT; /* not intuitive, but needed */
X=RX-.0140; Y=RY+.0125; OUTPUT; /* upper left corner */
X=X; Y=Y-.0280; OUTPUT; /* lower left corner */
X=X+.0280; Y=Y; OUTPUT; /* lower right corner */
X=X; Y=Y+.0280; OUTPUT; /* upper right corner */
X=X-.0280; Y=Y; OUTPUT; /* upper left corner AGAIN! */
RUN;

/* different application may need boxes of different size & shape */
/* if so, BOXES DATA step still needs X=. & Y=., and to close the polygon */

DATA US_WBOX; /* interleave map and box coordinates */
SET MAPS.US BOXES; BY STATE SEGMENT;
RUN;

. (PATTERNS, LEGEND, TITLE)
.

PROC GMAP MAP=US_WBOX DATA=RESPONSE ANNO=ANNODATA ALL; /* map with blanking */
/* ***** using map=maps.us will turn off blanking */
. . .

```

Exhibit 4: Program for Figure 4

Figure 1.

SUGI 14 Attendance, By State

Defaults Unacceptable

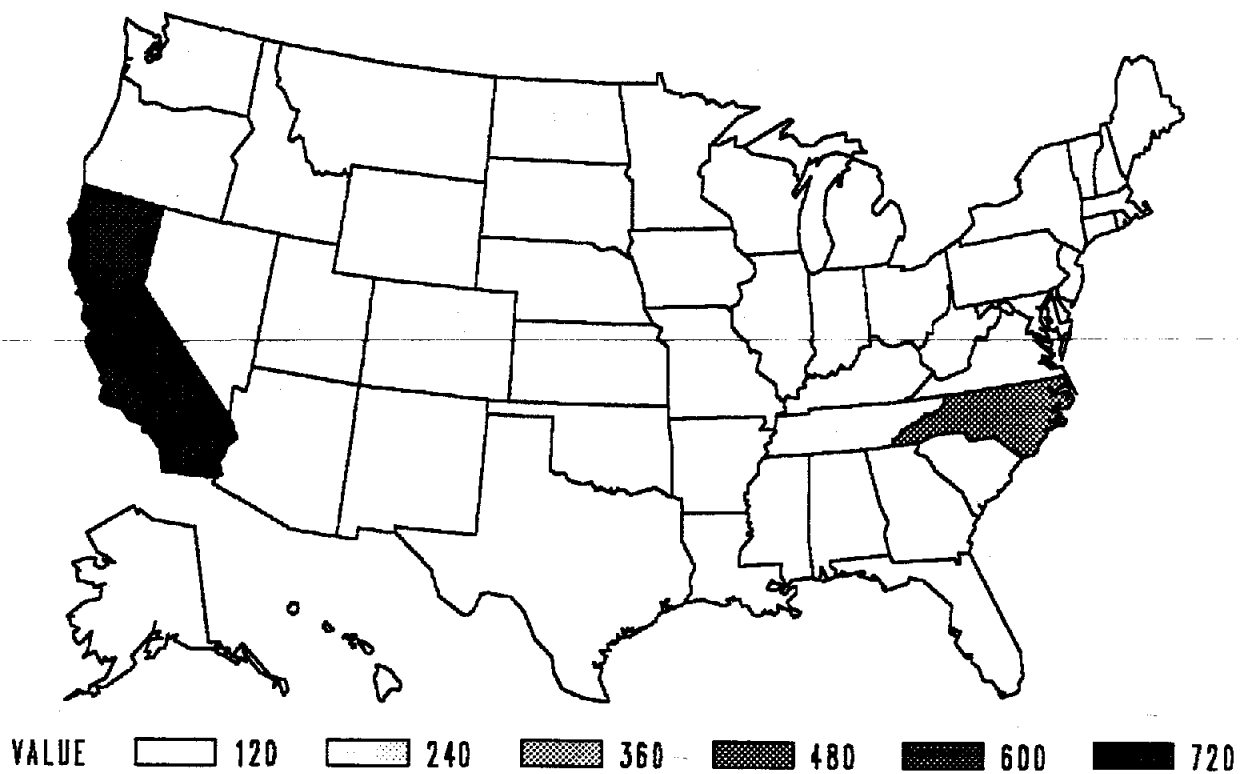


Figure 2.

SUGI 14 Attendance, By State

Versus Percentiles

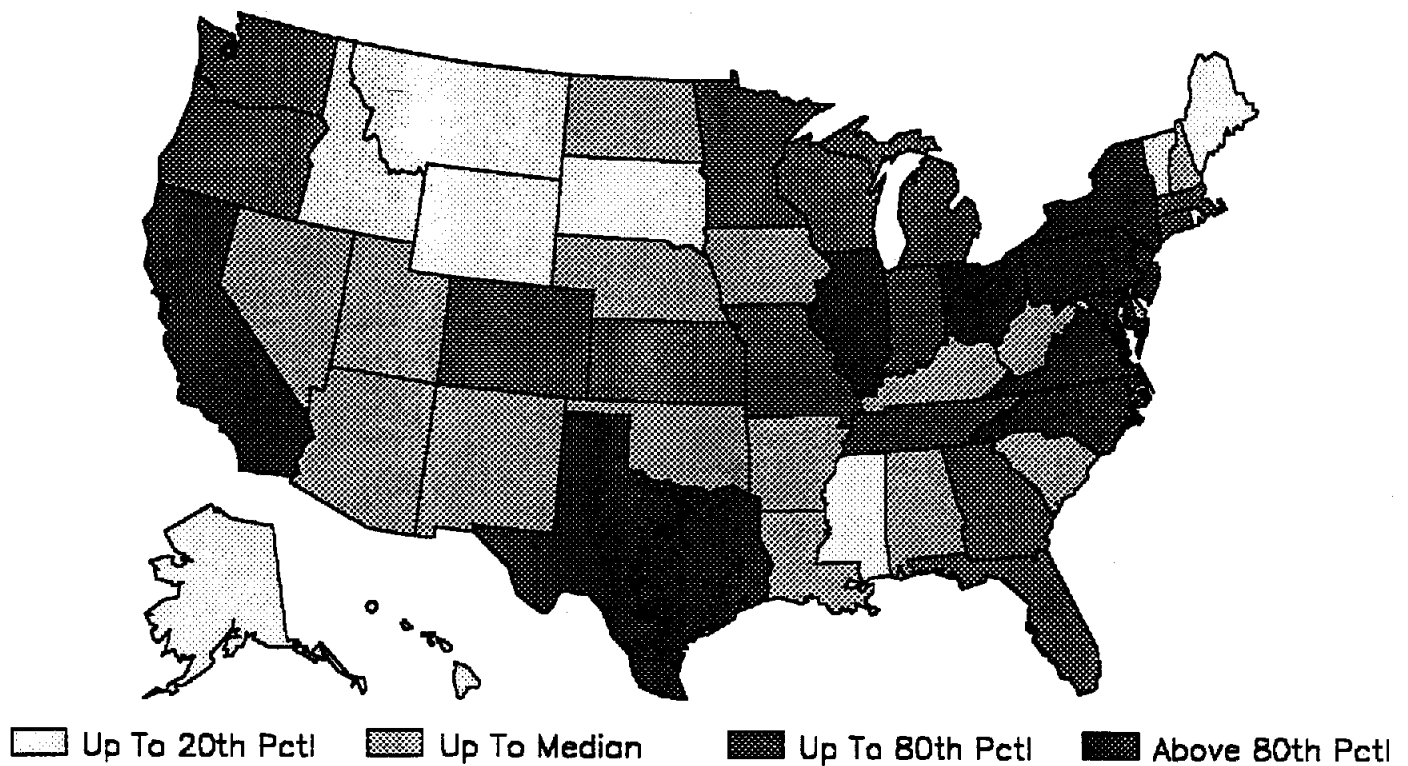


Figure 3.

SUGI 14 Attendance, By State

Needs Subichin's Blanking

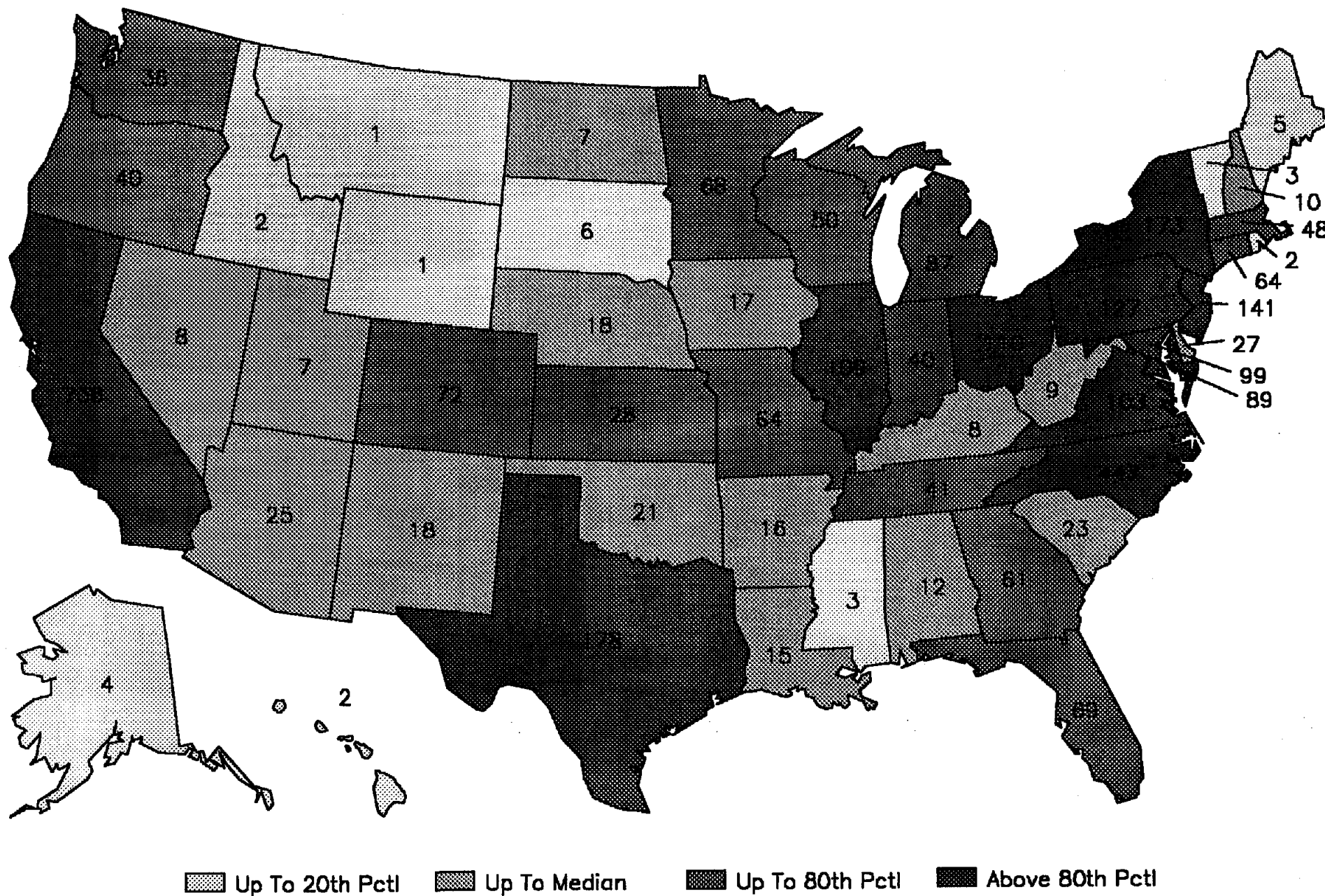


Figure 4. Map of SUGI 14 Attendance: Showing State, Count, & Rank – Uses Customized State Centers and Subichin's Blanking

