The Basics of Map Creation with SAS/GRAPH®

Mike Zdeb
University@Albany
School of Public Health
Rensselaer, NY

SUGI 29
MONTRÉAL
SAS® USERS GROUP INTERNATIONAL
MAY 9–12, 2004

- MAP CREATION WITH SAS

  SAS/GRAPH (PROC GMAP)
  SAS/GIS

- PROC GMAP - FOUR TYPES OF MAPS

  CHOROPLETH (THEMATIC)
  BLOCK
  PRISM
  SURFACE

- DATA STEP MATCHED-MERGE

  COMBINE TWO OR MORE DATA SETS BASED ON THE VALUE OF ONE OR MORE BY-VARIABLES

- PROC GMAP

  COMBINE A MAP DATA SET WITH A RESPONSE DATA SET BASED ON THE VALUE OF ONE OR MORE ID VARIABLES

- MAP DATA SETS

  SAS-SUPPLIED MAPS OF ALMOST EVERY COUNTRY IN THE WORLD (DEFAULT LIBREF "MAPS")

  SEVERAL DIFFERENT MAPS AT VARIOUS GEOGRAPHIC LEVELS OF CANADA AND THE UNITED STATES

  PURCHASE MAPS

  CONVERT SHAPEFILES (V9)

  MAKE-YOUR-OWN MAP

- RESPONSE DATA SETS

  DATA TO BE DISPLAYED IN A MAP

  MUST HAVE A VARIABLE IDENTIFYING GEOGRAPHIC AREAS

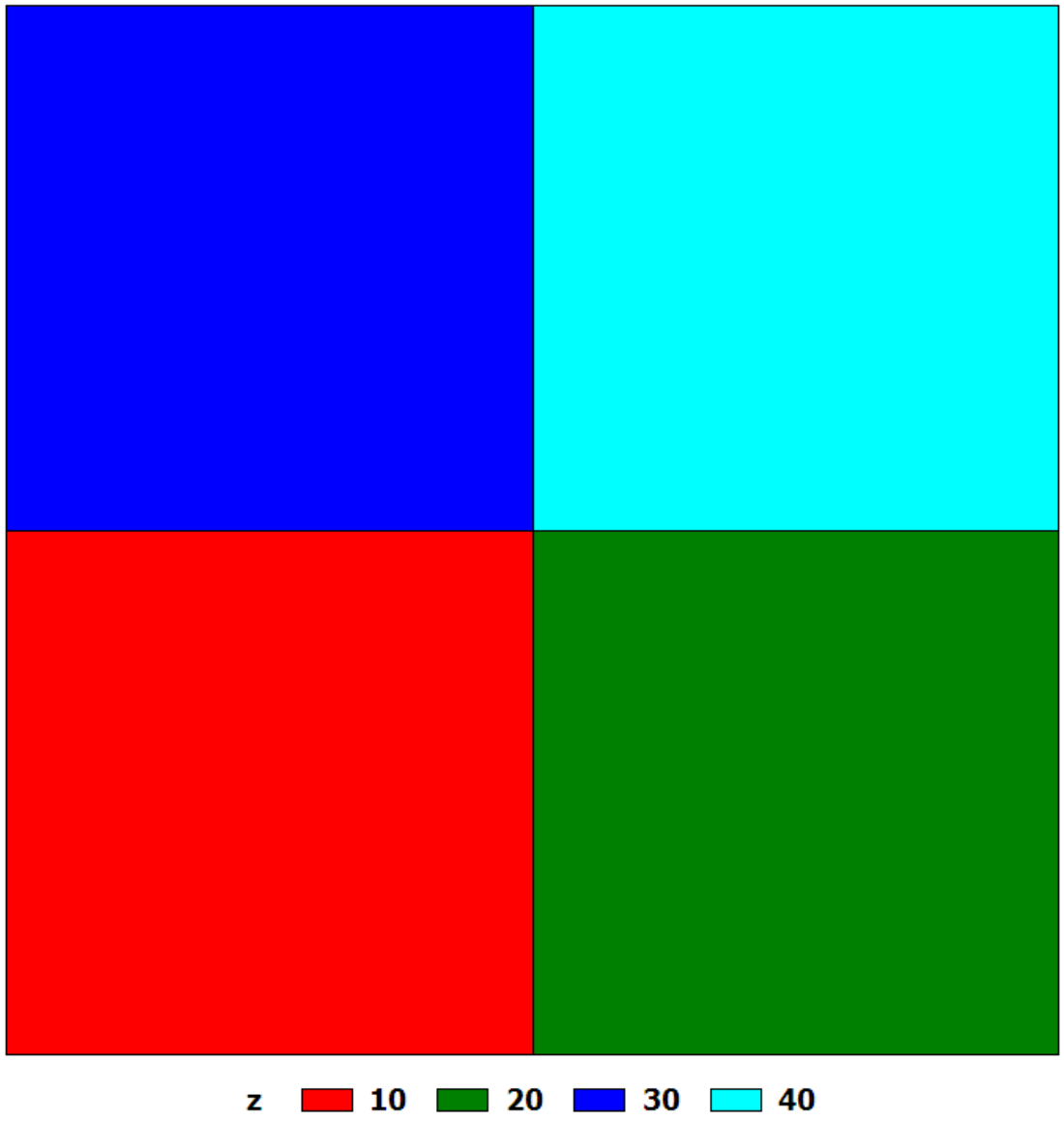
  VARIABLE MUST HAVE THE SAME NAME AND TYPE AS THE CORRESPONDING VARIABLE IN THE MAP DATA SET

MAKE YOUR OWN MAP

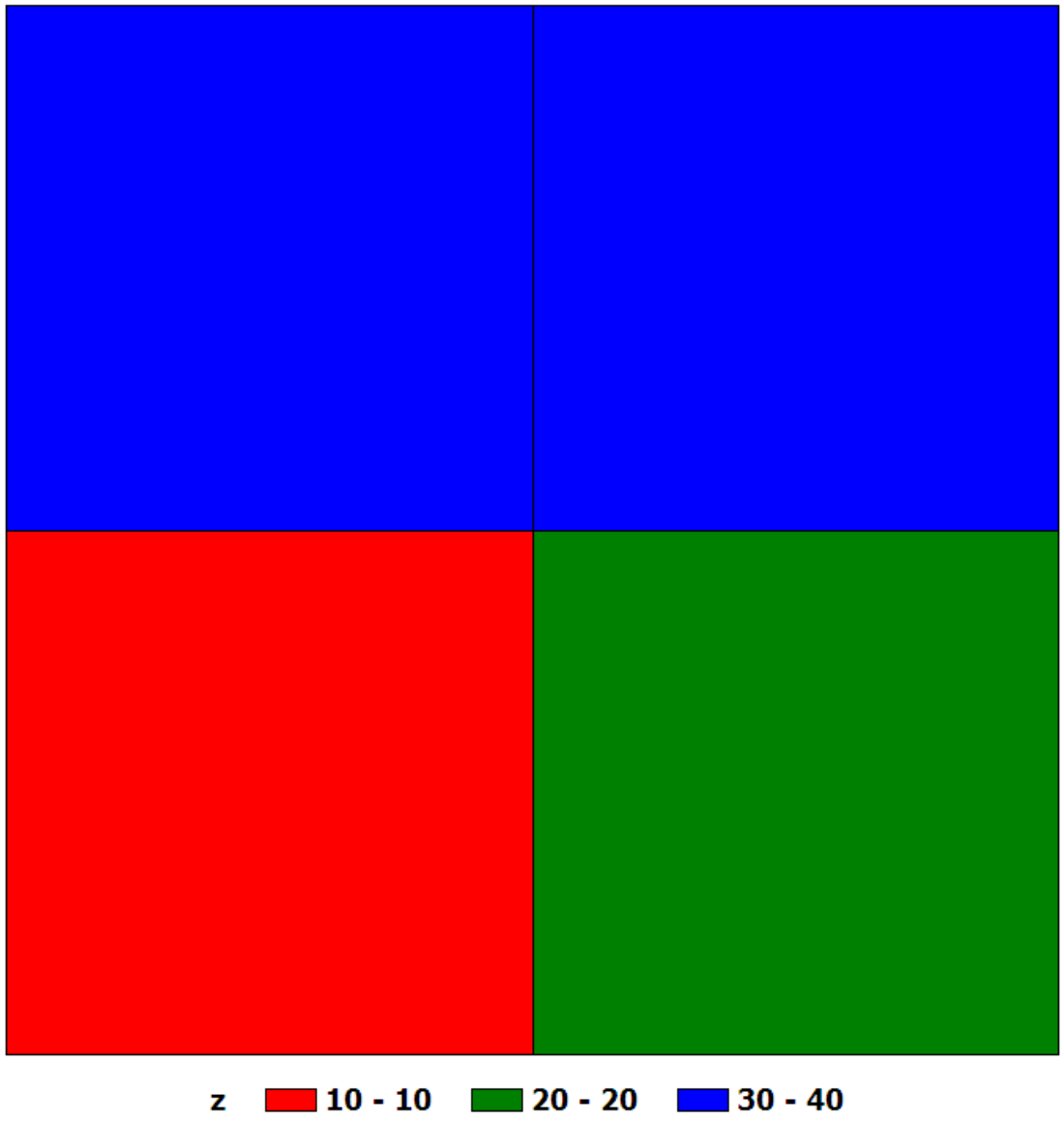
```
* the map data set;
data poly_map;
input id x y @@;
datalines;
1 0 0   1 1 0   1 1 1   1 0 1
2 1 0   2 2 0   2 2 1   2 1 1
3 0 1   3 1 1   3 1 2   3 0 2
4 1 1   4 2 1   4 2 2   4 1 2
;
run;
```

```
* the response data set;
data poly_dat;
input id z @@;
datalines;
1 10 2 20 3 30 4 40
;
run;

title 'EXAMPLE #1 - MAKE YOUR OWN MAP';

* draw the map;
proc gmap
data=poly_dat
map=poly_map;
id id;
choro z / discrete;
run;
quit;
```

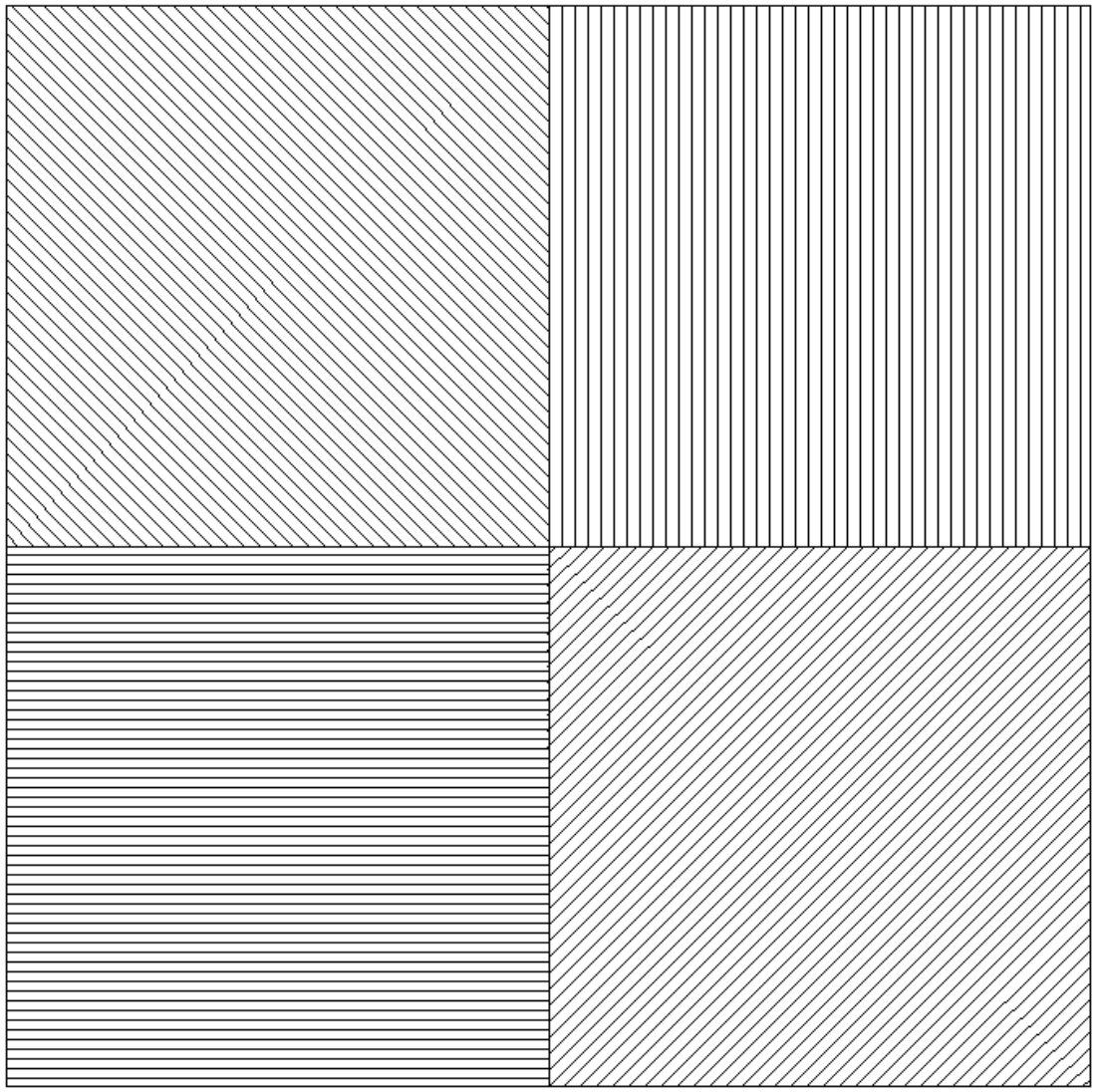
EXAMPLE #1 - MAKE YOUR OWN MAP



EXAMPLE #1 - MAKE YOUR OWN MAP (NO DISCRETE OPTION)



EXAMPLE #1 - MAKE YOUR OWN MAP (SELECT PATTERNS)



- NOTICE

  EASY

  PROC GMAP => SOPHISTICATED DRAWING PROGRAM

  DISCRETE / DEFAULT (SAS-SELECTED)

- REAL MAP

  SAS-SUPPLIED MAP DATA SET (US)

  OUTLINE MAP OF UNITED STATES
  SHOWING STATE BOUNDARIES

```
* select an empty pattern (v=me) and a title;

pattern v=me;
title 'EXAMPLE #2 - US MAP DATA SET';

* use a SAS-supplied  map data set  (US) as
  both the map and response data sets;
proc gmap
map=maps.us
data=maps.us (obs=1)
all;
id state;
choro state / nolegend;
run;
quit;
```

EXAMPLE #2 - US MAP DATA SET

- NOTICE

  US MAP DATA SET USED AS BOTH THE MAP AND RESPONSE DATA SET

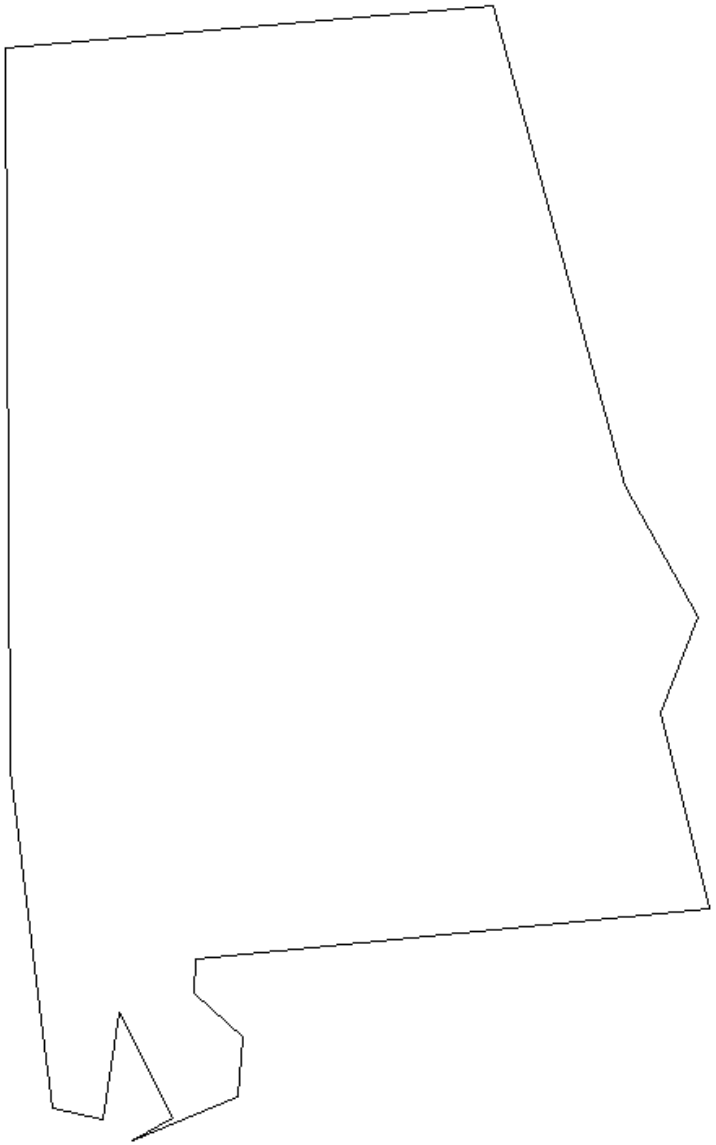
  ONLY ONE OBSERVATION USED IN THE RESPONSE DATA SET

  "ALL" OPTION USED

  DEFAULT - SHOW ONLY THOSE MAP AREAS IN THE RESPONSE DATA SET

  LEGEND SUPPRESSED (NOLEGEND)

EXAMPLE #2 - US MAP DATA SET (NO ALL OPTION USED)



- NEW YORK STATE COUNTIES

  SAS-SUPPLIED MAP DATA SET (COUNTIES)

  COUNTY-BASED MAP OF ALL STATES

  TWO GEOGRAPHIC IDENTIFIERS (STATE, COUNTY)

  USE A SUBSET OF THE COUNTIES MAP DATA SET --- SELECT NEW YORK STATE USING THE FIPS STATE NUMBER

```
* create a 'no fill' pattern (repeated 62 times
   one for each NYS county) and titles;

pattern c=black v=me r=62;

title1 'EXAMPLE #3 - MAPS.COUNTIES MAP DATA SET';
title2 'NEW YORK STATE COUNTIES';

* draw the map;
proc gmap
data=maps.counties
map=maps.counties;
id county;
choro county / nolegend;
where state eq 36;
run;
quit;
```

EXAMPLE #3 - MAPS.COUNTIES MAP DATA SET
NEW YORK STATE COUNTIES

- NOTICE

  WRONG DIRECTION

  WRONG SHAPE ("SQUISHED")

- WHY

  US MAP DATA SET - CARTESIAN COORDINATES (PROJECTED)

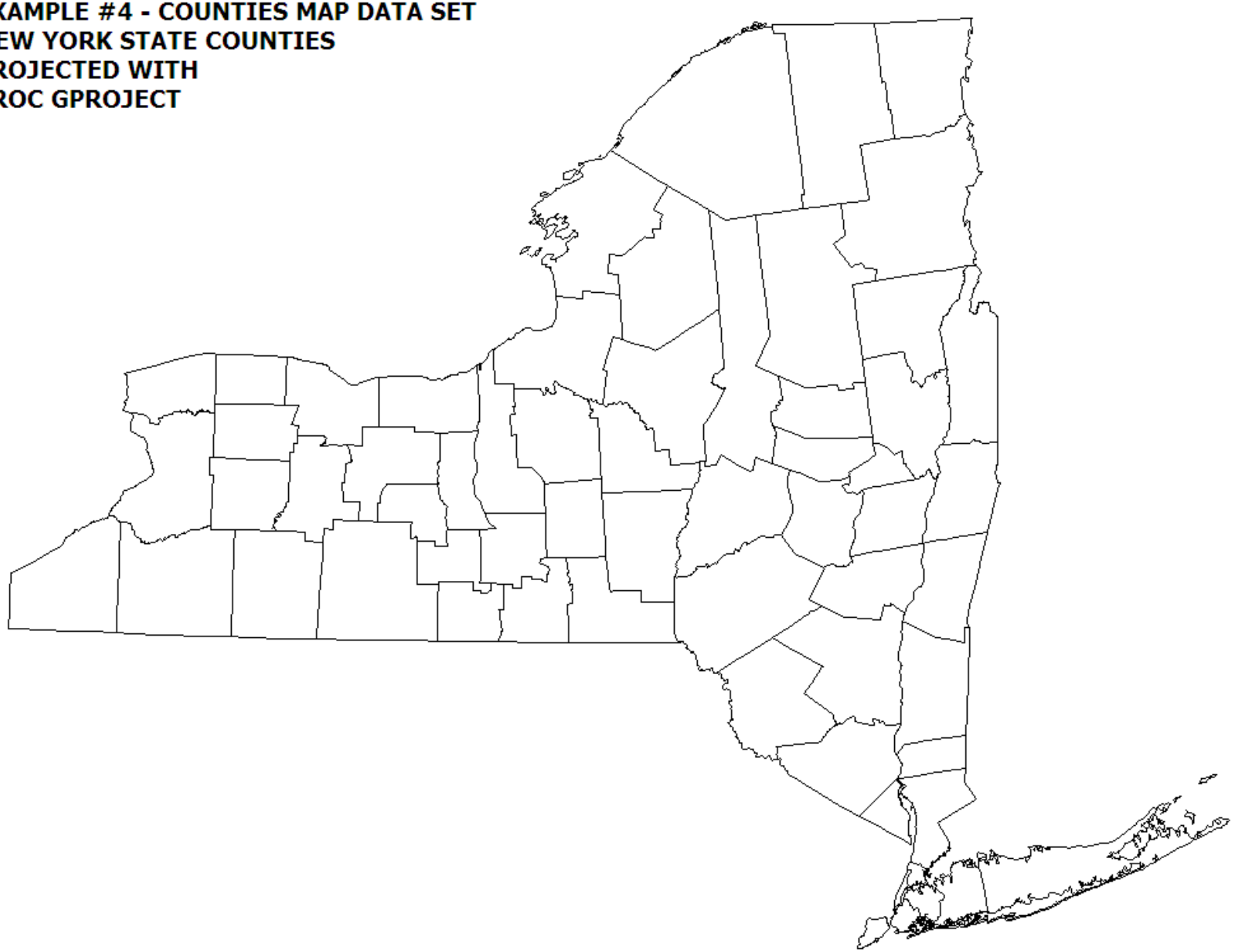
  COUNTIES MAP DATA SET - LONGITUDE AND LATITUDE (UNPROJECTED)

PROC GPROJECT

```
* create a map data set containing projected coordinates
   limit the map to New York state;
proc gproject
data=maps.counties
out=nys;
where state eq 36;
id state;
run;
```

```
* select an empty pattern (v=me);
pattern v=me;

* draw the map;
proc gmap
data=nys (obs=1)
map=nys
all;
id county;
choro county / nolegend;
note ' EXAMPLE #4 - COUNTIES MAP DATA SET'
j=l  ' NEW YORK STATE COUNTIES'
j=l  ' PROJECTED WITH'
j=l  ' PROC GPROJECT';
run;
quit;
```

EXAMPLE #4 - COUNTIES MAP DATA SET
NEW YORK STATE COUNTIES
PROJECTED WITH
PROC GPROJECT



- NOTICE

  CORRECT DIRECTION AND SHAPE

  ALMOST ALL SAS-SUPPLIED MAP DATA SETS CONTAIN BOTH PROJECTED (X, Y) AND UNPROJECTED COORDINATES (SOMETIMES REFERRED TO AS "DEPROJECTED")

  NOTE USED INSTEAD OF A TITLE

  TITLES LOCATED OUTSIDE PROCEDURE OUTPUT AREA --- NOTES SHARE PROCEDURE OUTPUT AREA

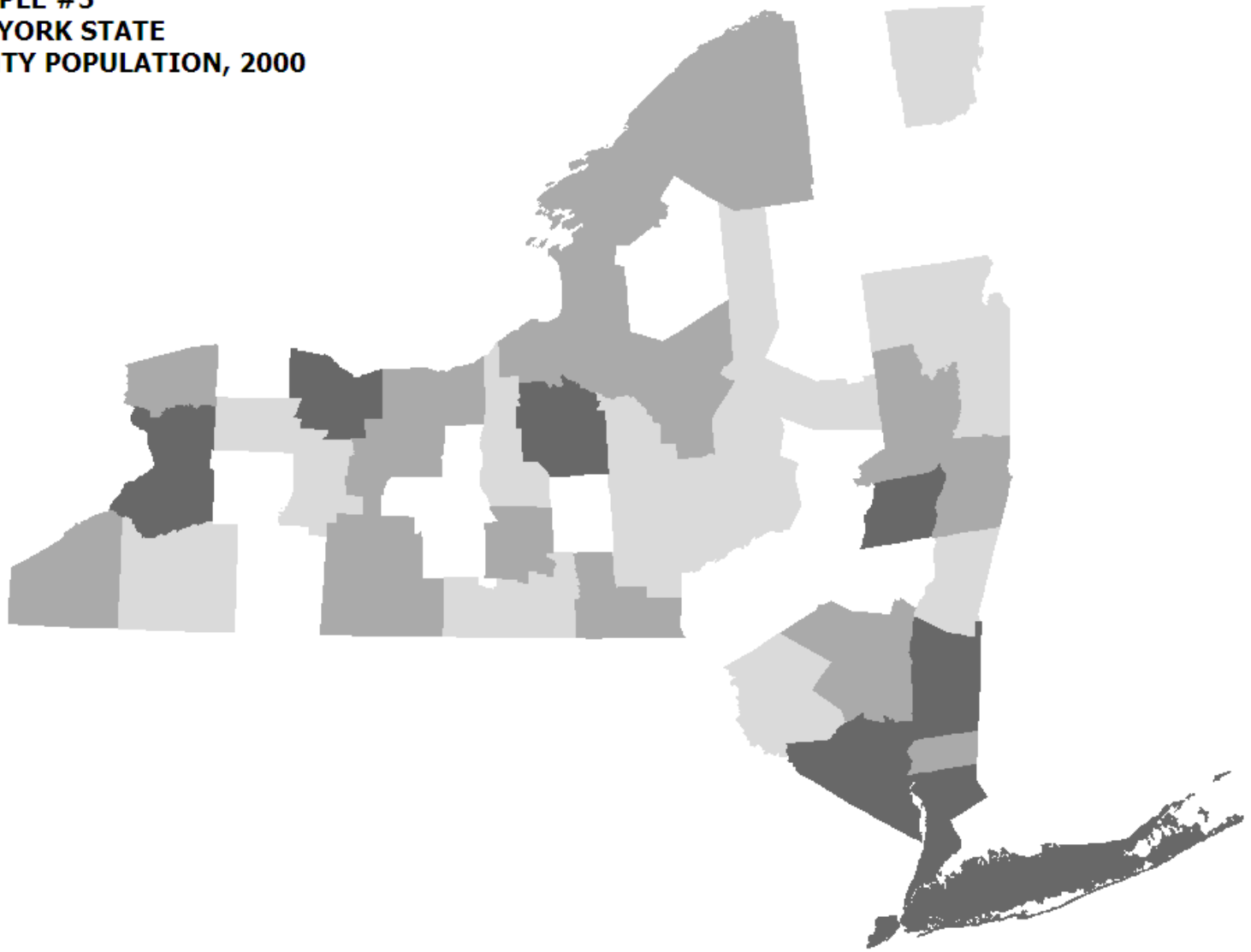
- CHOROPLETH (THEMATIC) MAP OF NEW YORK USING 2000 CENSUS DATA

```
* example 5
* create a format to group observations (counties) based on
the population;
proc format;
value pop2000_
5379 - 51134     = '5,379 - 51,134'
51401 - 91070    = '51,401 - 91,070'
93765 - 235469   = '93,765 - 235,469'
280150 - 2465326 = '280,150 - 2,465,326'
;
run;
```

```
* 2000 census population, New York State counties;
data nys2000;
input county pop2000;
datalines;
001 294565
003   49927
005 1332650
.
.
.
123   24621
;
run;

* fill patterns for the map areas (gray-scale fills);
pattern1 v=ms c=grayff;
pattern2 v=ms c=grayda;
pattern3 v=ms c=grayaa;
pattern4 v=ms c=gray68;
```

```
* draw the map;
proc gmap
data=nys2000
map=nys;
id county;
choro pop2000 / discrete;
format pop2000 pop2000_.;
note ' EXAMPLE #5'
j=l ' NEW YORK STATE'
j=l ' COUNTY POPULATION, 2000';
run;
quit;
```

EXAMPLE #5
NEW YORK STATE
COUNTY POPULATION, 2000



- NOTICE

  PATTERN STATEMENTS ALLOW 256 LEVELS OF GRAY (HEXADECIMAL NOTATION)

  WHITE=GRAYFF
  BLACK=GRAY00

  COUNTY OUTLINE COLOR SAME AS THE FILL COLOR

  LEGEND LOCATION REDUCES THE SIZE OF THE MAP (NOT IN THE PROCEDURE OUTPUT AREA)

- MAKE A FEW "ADJUSTMENTS"

  MODIFY AND MOVE THE LEGEND

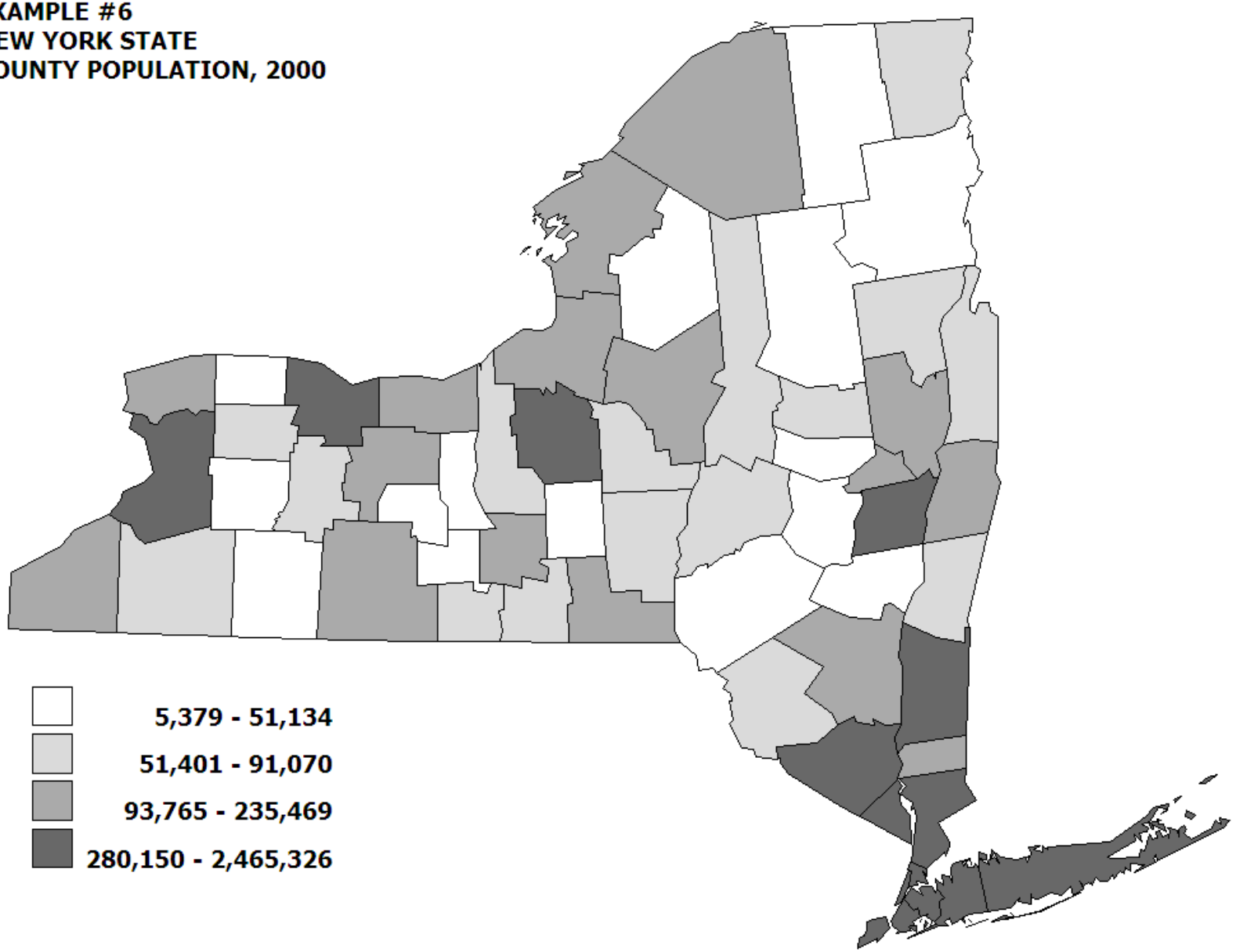
  OUTLINE THE COUNTIES

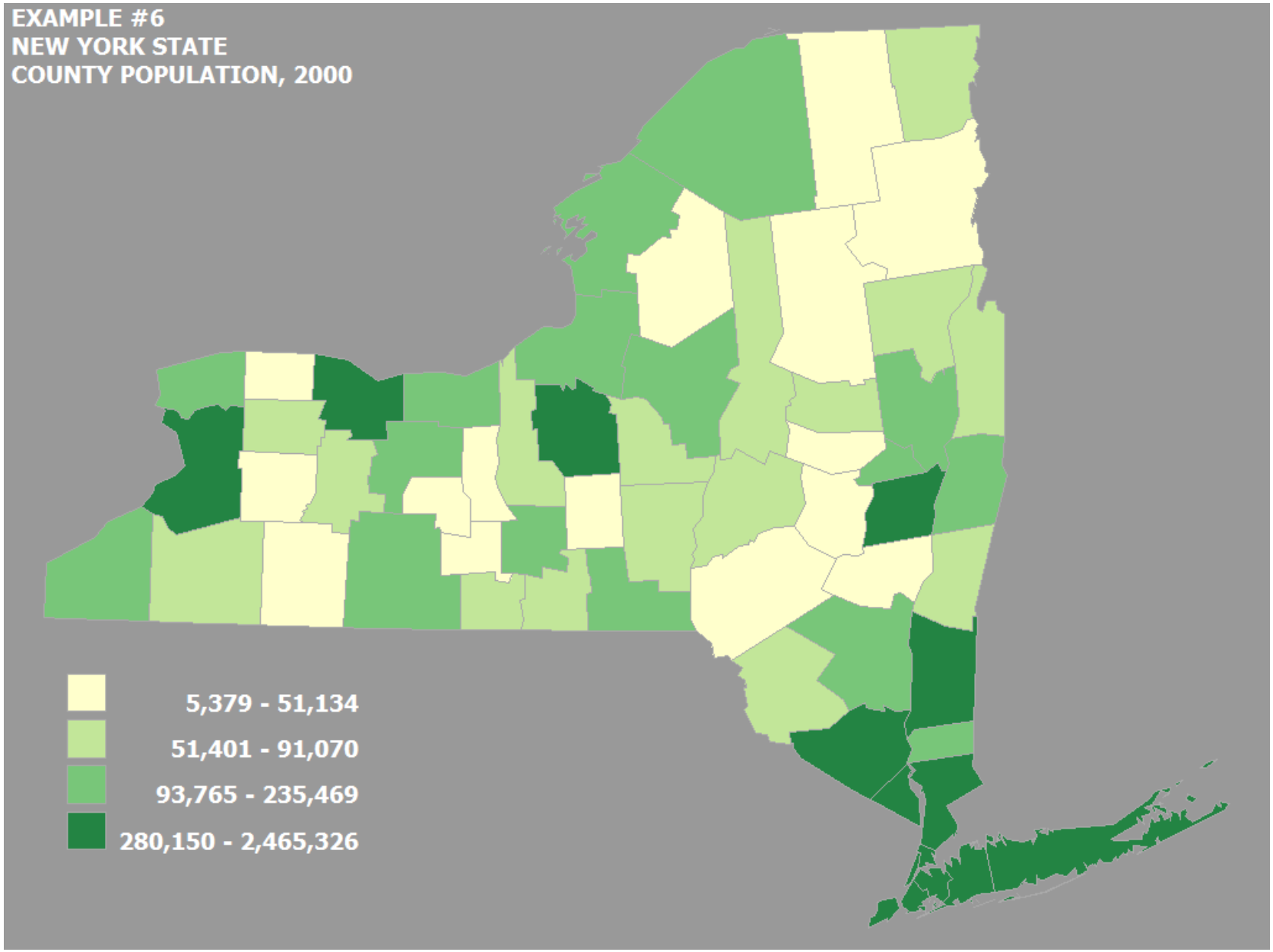
  REDUCE THE NUMBER OF X-Y COORDINATES USED TO DRAW THE MAP

  TRY ONE IN COLOR...RGB COLORS

```
pattern1 v=ms c=cxffffcc;
pattern2 v=ms c=cxc2e699;
pattern3 v=ms c=cx78c679;
pattern4 v=ms c=cx238443;
```

EXAMPLE #6
NEW YORK STATE
COUNTY POPULATION, 2000




EXAMPLE #6
NEW YORK STATE
COUNTY POPULATION, 2000
5,379 - 51,134
51,401 - 91,070
93,765 - 235,469
280,150 - 2,465,326

- NOTICE

  CAN DIFFERENTIATE COUNTIES

  LEGEND DOES NOT DECREASE MAP SIZE

  LOOKS "QUITE NICE"

  COLORS --- http://www.colorbrewer.com

- COMBINE MAP AREAS

  CREATE NEW, LARGER MAP AREAS FROM LOWER LEVEL OF GEOGRAPHY

  NEED RULES THAT ASSIGN CURRENT GEOGRAPHIC AREAS TO LARGER AREAS

  EXAMPLES NEW YORK STATE COUNTIES TO HSAS

  NEW YORK CITY ZIP CODES TO NEIGHBORHOODS

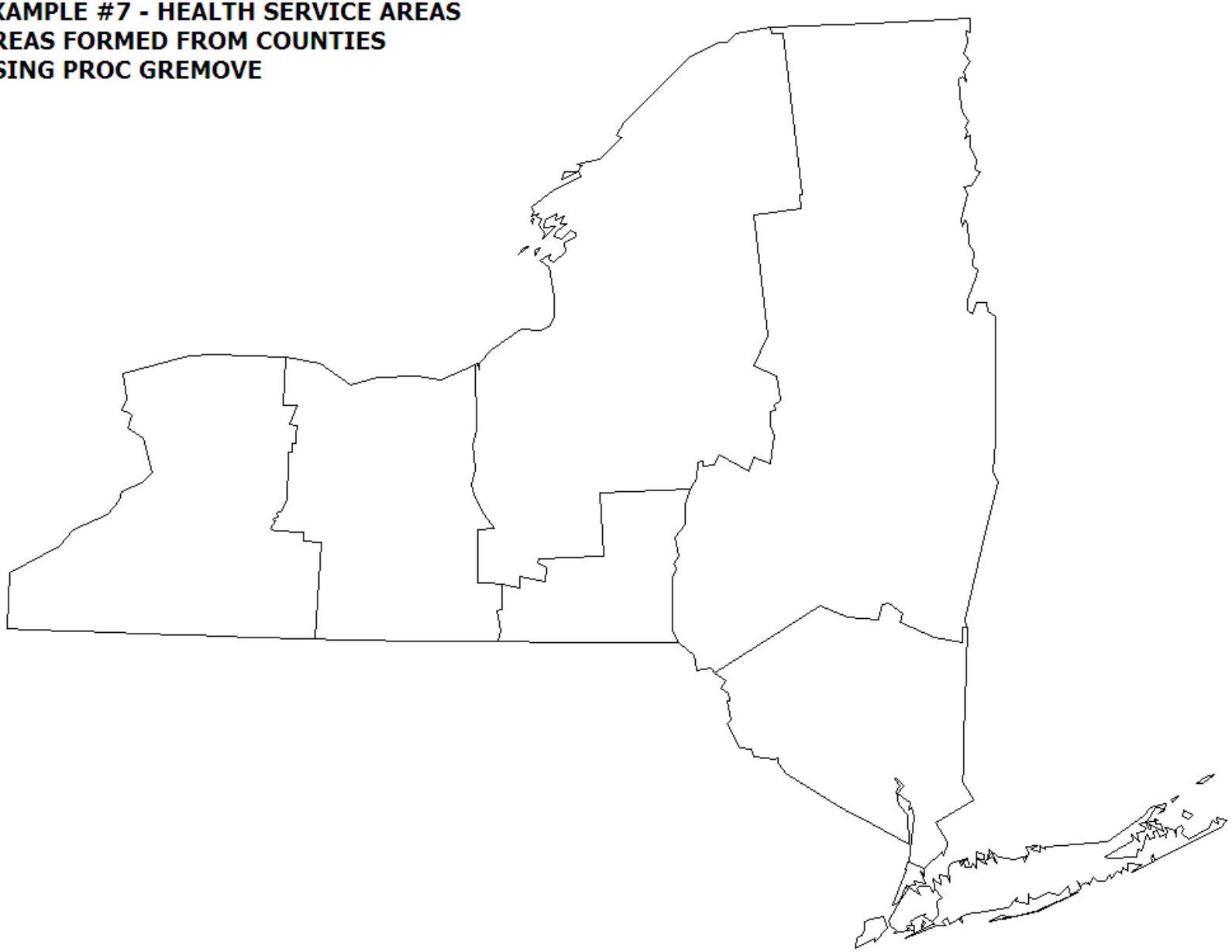
PROC GREMOVE

```
* rules to assign counties to HSAS;
proc format;
value cou2hsa
3,9,13,29,37,63,73,121 = '1'
15,51,55,69,97,99,101,117,123 = '2'
11,23,43,45,49,53,65,67,75,89,109 = '3'
7,17,107 = '4'
1,19,21,25,31,33,35,39,41,57,77,83,91,93,95,113,115 = '5'
27,71,79,87,105,111,119 = '6'
5,47,61,81,85 = '7'
59,103 = '8';
run;

* select New York State counties, add HSA to data set;
data hsatemp;
set maps.counties;
where state eq 36 and density lt 6;
hsa = put(county,cou2hsa.);
run;
```

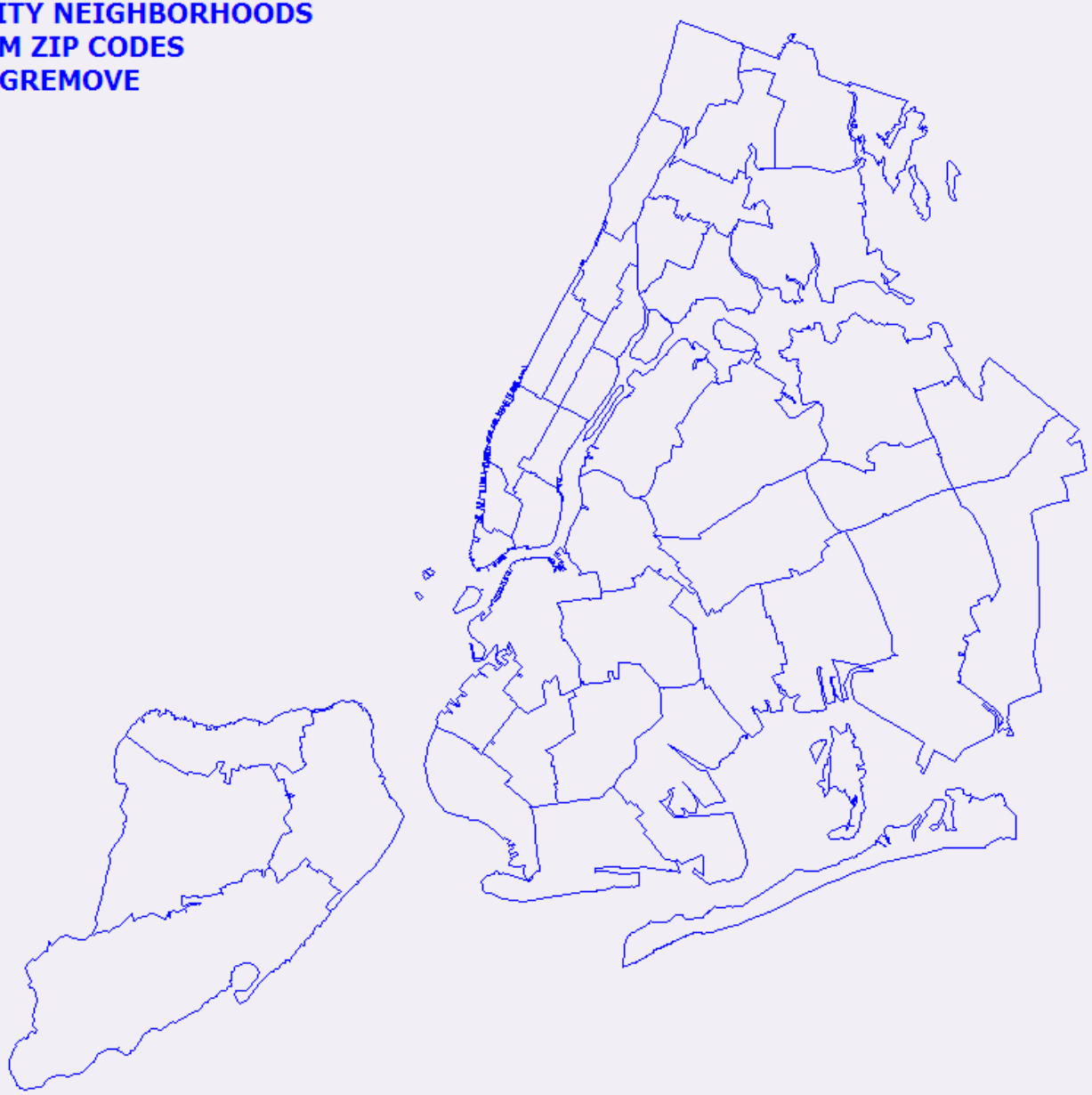
```
* sort map data set in order by new geographic designation;
proc sort data=hsatemp;
by hsa;
run;

* create HSA boundaries using PROC GREMOVE;
proc gremove
data=hsatemp
out=hsamap;
by hsa;
id county;
run;

* project the new map data set;
proc gproject
data=hsamap
out=hsaproj;
id hsa;
run;
```

```
* an empty fill pattern;
pattern v=me;

* draw the map;
proc gmap
data=hsaproj (obs=1)
map=hsaproj
all;
id hsa;
choro hsa / discrete nolegend;
note ' EXAMPLE #7 - HEALTH SERVICE AREAS'
j=l ' AREAS FORMED FROM COUNTIES'
j=l ' USING PROC GREMOVE';
run;
quit;
```

EXAMPLE #7 - HEALTH SERVICE AREAS
AREAS FORMED FROM COUNTIES
USING PROC GREMOVE



NEW YORK CITY NEIGHBORHOODS
FORMED FROM ZIP CODES
USING PROC GREMOVE



- NOTICE

  MUST HAVE A RULE TO ASSIGN EACH LOWER LEVEL OF GEOGRAPHY TO A HIGHER LEVEL (E.G. ZIP CODE TO A NEIGHBORHOOD)

- LABEL MAP AREAS

  NO PROC GMAP OPTION TO LABEL MAP AREAS

  USE AN ANNOTATE DATA SET

  AN ANNOTATE DATA SET IS THE EQUIVALENT OF PUTTING INTO TEXT WHAT YOU ARE THINKING IF YOU ARE USING A GRAPHICS EDITING PROGRAM

  WHAT-WHERE-HOW

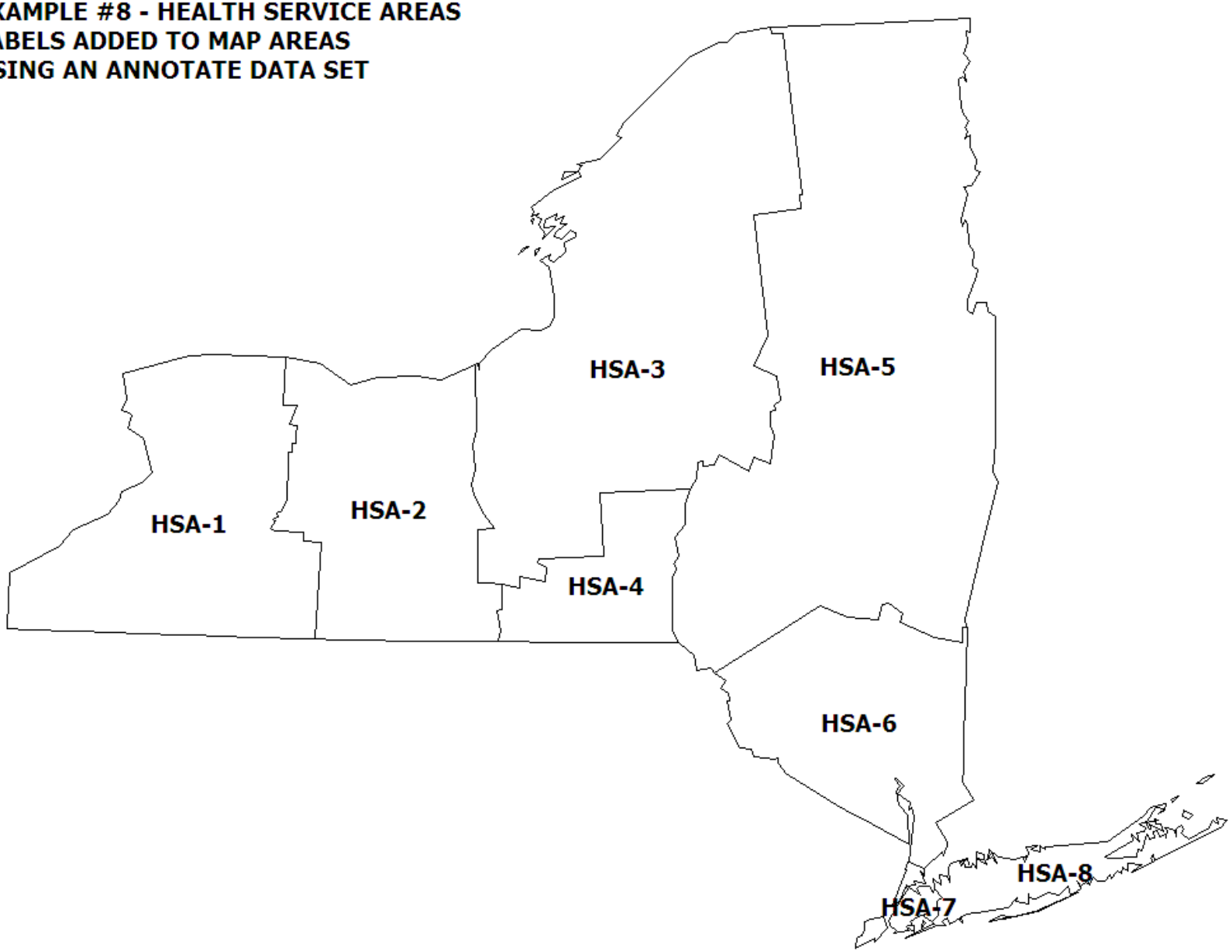
"TYPICAL" ANNOTATE DATA SET

```
* the annotate data set;
data labels;
length text $5;
retain xsys ysys '2' hsys  '3' position  '5'
function  'label' size 3 style '"Tahoma/bo"' when  'a' ;
input text x y;
*the longitude/latitude of HSA area centroids;
datalines;
HSA-1  1.371803  0.742941
HSA-2  1.348815  0.744540
HSA-3  1.321340  0.756707
HSA-4  1.323836  0.738309
HSA-5  1.294420  0.756696
HSA-6  1.295074  0.726438
HSA-7  1.288815  0.710749
HSA-8  1.273433  0.713304
;
run;
```

```
*combine the map data set with map labels;
data map_labl;
set hsamap labels;
run;

*project all observations (map and labels);
proc gproject
data=map_labl
out=hsaproj;
id hsa;
run;

*separate the labels from the map dataset;
data hsaproj labels;
set hsaproj;
if function eq 'label' then output labels;
else output hsaproj;
run;
```

```
* an empty pattern;
pattern v=me;

* draw the map with labels;
proc gmap
data=hsaproj (obs=1)
map=hsaproj
all;
id hsa;
choro hsa/
discrete
nolegend
annotate=labels;
note ' EXAMPLE #8 - HEALTH SERVICE AREAS'
j=l ' LABELS ADDED TO MAP AREAS'
j=l ' USING AN ANNOTATE DATA SET';
run;
quit;
```

EXAMPLE #8 - HEALTH SERVICE AREAS
LABELS ADDED TO MAP AREAS
USING AN ANNOTATE DATA SET



- NOTICE

  THIS IS "NOT EASY"

  %MAPLABEL ---NEW ANNOTATE MACRO AVAILABLE IN V9 TO MAKE THIS "EASIER"

  YOU NEED THE X-Y COORDINATES OF THE POSITION FOR THE LABEL

- MAPS FOR THE WEB

  GIF (ALL EXAMPLES SHOWN THUS FAR WERE CREATED AS GIF FILES)

  PDF

  MORE...

- CREATE A MAP WITH HYPERLINKS

  MAP OF NEW JERSEY

  EACH COUNTY LINKED

  EACH LEGEND BOX LINKED

```
* group counties based on population;
proc format;
value pop2000_
 64285 - 144166 =  '64,285-144,166'
146438 - 350761 =  '146,438-350,761'
423394 - 522541 =  '423,394-522,541'
608975 - 884118 =  '608,975-884,118'
;
run;
```

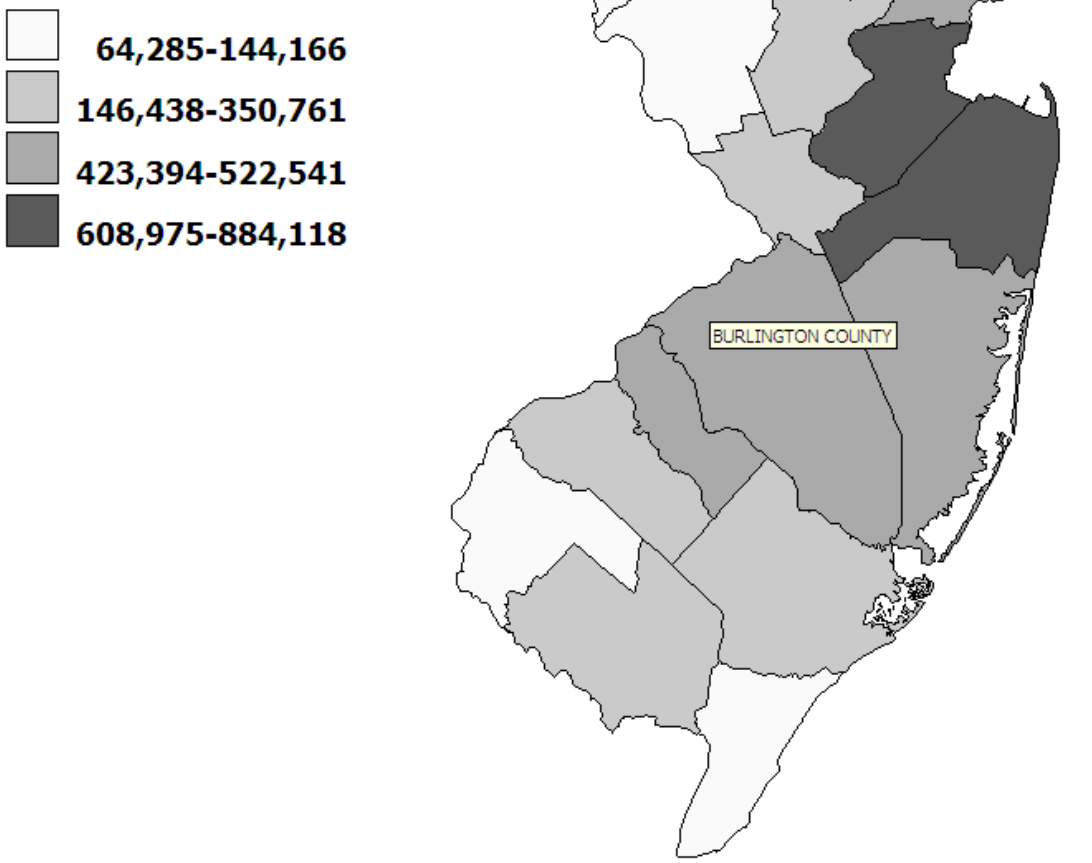
```
* input data - create county and legend links;
data njpop;
input county county_name & : $upcase20.
      (pop1990 pop2000) (: comma8.);
* create a link for each county;
county_link = 'ALT="' || trim(county_name) ||
                  '" href=county' || put(county,z3.) || '.htm';

* create a link for the legend;
if pop2000 le 144166 then
legend_link = 'ALT="1ST QUARTILE" href=nj2000_q1.htm';
else
if pop2000 le 350761 then
legend_link = 'ALT="2ND QUARTILE" href=nj2000_q2.htm';
else
if pop2000 le 522541 then
legend_link = 'ALT="3RD QUARTILE" href=nj2000_q3.htm';
else
legend_link = 'ALT="4TH QUARTILE" href=nj2000_q4.htm';
```

```
datalines;
001 Atlantic County         224,327         252,552
.
.
.
041 Warren County           91,607          102,437
;
run;

* use the GIF device driver - select GIF attributes;
goptions reset=all dev=gif  xpixels=1024 ypixels=768
ftext='Tahoma/bo' htext=4 gunit=pct ;

pattern1 v=ms c=grayfa;
pattern2 v=ms c=grayca;
pattern3 v=ms c=grayaa;
pattern4 v=ms c=gray5a;
```

```
* define a legend;
legend1 mode=share origin=(5,50) label=none value=(h=3 j=r)
shape=bar(3,4) across=1
;

* select New Jersey (state number 34) from the
   MAPS.COUNTIES map data set;
proc gproject data=maps.counties out=nj asis;
where state eq 34;
id county;
run;
```

```
* use ODS to create an HTML file
  it will link to the GIF file with the actual map;
* use linking variables defined earlier;
ods listing close;
ods html path='z:\' (url=none) file='nj2000.htm';
proc gmap map=nj data=njpop ;
id county;
choro  pop2000 / discrete coutline=black legend=legend1
                 html=county_link html_legend=legend_link;
format pop2000 pop2000_.;
note  j=l ' EXAMPLE #9'
j=l ' NEW JERSEY COUNTIES'
j=l ' 2000 CENSUS POPULATION'
h=3  j=l ' (CLICK ON COUNTY OR LEGEND TO'
j=l ' DRILL DOWN TO ADDITIONAL DATA)'
;
run;
quit;
ods html close;
ods listing;
```

EXAMPLE #9
NEW JERSEY COUNTIES
2000 CENSUS POPULATION
(CLICK ON COUNTY OR LEGEND TO DRILL DOWN TO ADDITIONAL DATA)



- CREATE A MAP WITH "POP-UPS"

  MAP OF NEW JERSEY

  EACH COUNTY LINKED TO A POP-UP BOX THAT CONTAINS INFORMATION ABOUT THAT COUNTY

```
* (same format and data set used as in previous map)

* text to put in floating text box (after line with TIP)
  calculate %-change in population 1990 to 2000;
data popup;
length popvar $200;
set njpop;
pct    = 100*(pop2000 - pop1990) / pop1990;
popvar = 'tip=['||
         quote(couname) || ' ' ||
         quote('POPULATION')  || ' ' ||
         quote('1990 :  ' || put(pop1990,comma10.)) || ' ' ||
         quote('2000 :  ' || put(pop2000,comma10.)) || ' ' ||
         quote('% CHANGE :  ' || put(pct,10.1))
         || ']';
run;
```

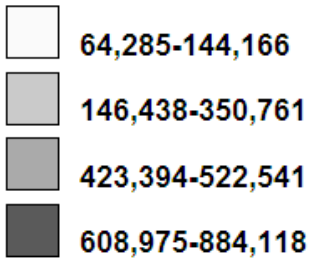
```
* use the JAVAMETA device driver - specify attributes;
goptions reset=all device=javameta ftext='HelveticaBold'
                xpixels=1024 ypixels=768 htext=4 gunit=pct ;

* select patterns;
pattern1 v=ms c=grayfa; pattern2 v=ms c=grayca;
pattern3 v=ms c=grayaa; pattern4 v=ms c=gray5a;

* define a legend;
legend1 mode=share origin=(5,50) label=none shape=bar(3,4)
across=1 value=(h=3.5) ;

* select New Jersey (state number 34) from the
   MAPS.COUNTIES map data set;
proc gproject data=maps.counties out=nj asis;
where state eq 34;
id county;
run;
```

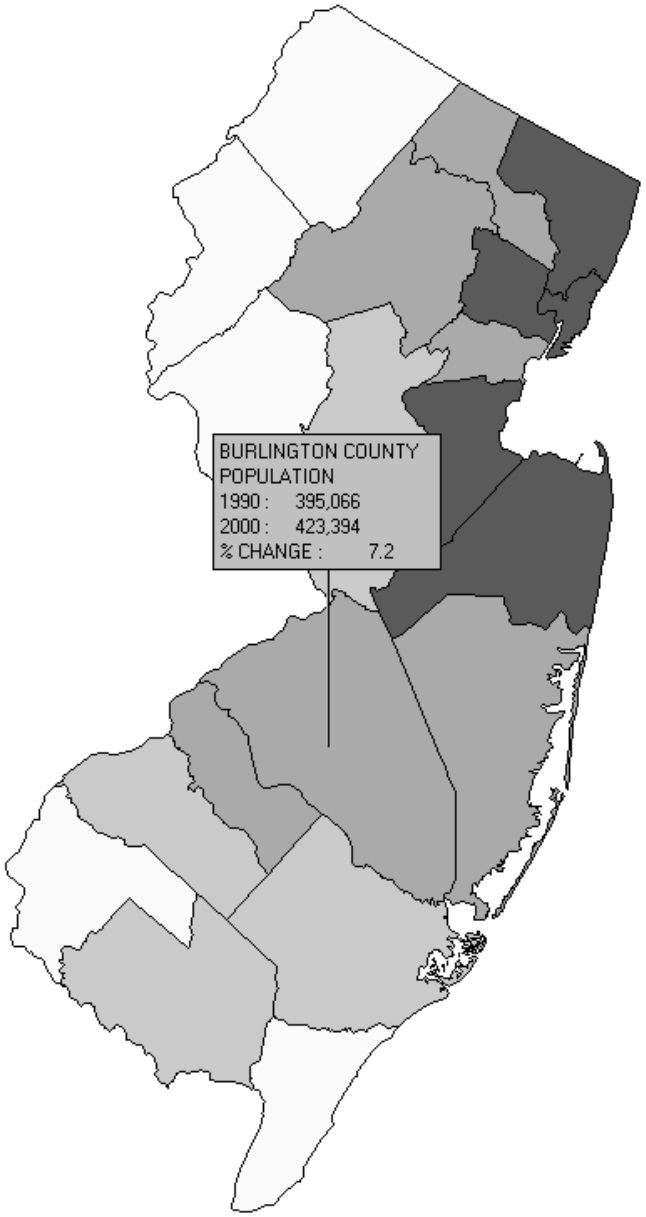
```
* create the map - select display options;
ods listing close;
ods html body="z:\nj_pop.htm"
            parameters=("BackgroundColor"="0xFFFFFF"
            "DataTipStyle"="Stick_Fixed"
            "ZoomControlEnabled"="False");

proc gmap
map=nj
data=popup;
id county;
choro  pop2000 / discrete coutline=black legend=legend1
html=popvar;
format pop2000 pop.;
note j=l ' EXAMPLE #11'
j=l ' NEW JERSEY COUNTIES'
j=l ' 2000 CENSUS POPULATION';
run;
quit;

ods pdf close;
ods listing;
```

EXAMPLE #11

NEW JERSEY COUNTIES

2000 CENSUS POPULATION





- RESOURCES

  MAPS ON LINE

  COLOR BREWER

  SEE PAPER IN PROCEEDINGS

- CONCLUSION

  EASY

  NOT EASY

  LIMITATIONS - CURIOSITY/ENTHUSIASM