

# INTRODUCTION TO SAS MAPS

Gur Hoshen, American Institutes for Research, Palo Alto, CA

## ABSTRACT

Visual representation of geographic data can offer insights which a reader or client might otherwise not grasp if presented in text or tabular format. SAS Graph provides some useful ways to display data with maps. This paper will provide a few examples of using maps.

## INTRODUCTION

PROC GMAP can create several kinds of maps. This paper will discuss choropleth maps which represent data as either patterns or colors. They are created using a map dataset and a response data. The map data set defines the map boundaries. For example, these could be state or county. The response data are the fields displayed on the map like costs, income levels, etc.

An ID field identifies the geographic unit. IDs could be the state, or a county ID. Of course, the ID field needs to be found in response dataset. GMAP basically merges records from the response data set to the map dataset and displays the results in a map.

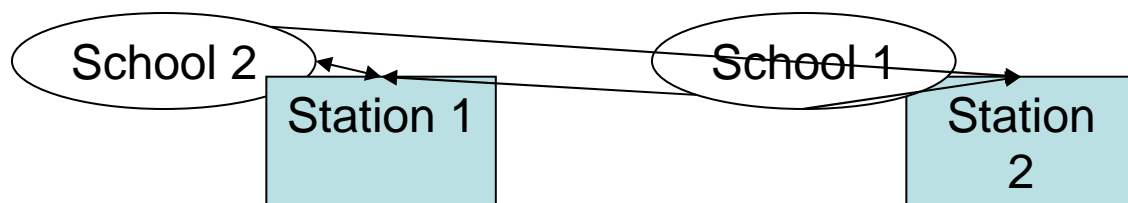
## X, Y COORDINATES

SAS has a number of maps in the MAPS library which is part of SAS GRAPH. The unprojected longitude and latitude information in these files are in radians, not in more familiar x and y coordinates, but the transformation below to radians is simple. Some of the datasets are projected, meaning they can be plotted with their existing position information. The position information, however, if to be merged with other data, needs to come from unprojected data.

One common issue is that the response dataset may need to be linked to another geographic dataset. Perhaps we are interested in merging housing prices from the Census but only have zip codes for a labor market analysis dataset. One could get x and y coordinates if, for example, the zip code, city, or county is known in the response data set from the Census. If the response data only had zip codes, one could use the zip longitude and latitude information in the Census Gazetteer files under <http://www.census.gov/geo/www/gazetteer/places2k.html>.

For fairly large mappings, like at the state level, the Census gazetteer is fairly reasonable. However, if one wants to calculate actual distances within a county between zip codes, results are somewhat unreliable. Census files are provided on an as is basis. Commercial vendors who sell zip code information are likely to have more accurate information.

In this example, a master list of schools with x and y coordinates are joined to a list of weather stations which also have x and y coordinates. Distances between schools and stations are calculated for every pair of schools and stations. The graphic below shows only two weather stations and two schools so only 4 distances are calculated using SQL.



The weather stations contain information about precipitation, and heating/cooling degree days which is important in a state like Alaska if one does a labor market study. Doing this Cartesian join below can be onerous because if, for example, there were 6,000 schools across 4 years of data and 5,000 weather stations, there would be 30,000,000 resulting records from the SQL join, but in this case, there are only a few weather stations. For joins with a large number of resultant records, use of where clause is advised, if it can be used.

```
proc sql;
  create table all_rec as
  select f.schcode, f.fy, f.zip,
         f.long as x1, f.latit as y1,
         c.long as x2, c.latit as y2 , hdd, cdd, precip
```

```

        from imf0 f , facil.climate c
        order by f.schcode, f.fy;
;
quit;

```

Distances for each x1, y1 and x2, y2 pairs are calculated, and the smallest distance for between each weather station and school can be kept. Note the radian conversion of dividing longitude/latitude by 180/pi. Note that these are unprojected coordinates and that the formula is taken from <http://support.sas.com/techsup/unotes/SN/003/003973.html>

```

data all_Rec;
  set all_rec;
  pi=4*atan(1);
  x1=(x1)/(180/pi);
  x2=(x2)/(180/pi);
  y1=(y1)/(180/pi);
  y2=(y2)/(180/pi);;
  distance = 3949.99 * arcos(sin(Y1) * sin(Y2) +cos(Y1) * cos(Y2) * cos(X1 - X2));
  drop pi;
run;

proc sort data=all_rec;
  by schcode fy distance;
run;

data all_rec;
  set all_rec;
  by schcode fy distance;
  if first.fy; * pick smallest distance assumes no missing data;
run;

```

## GMAP EXAMPLES

The reader is strongly advised to look at the maps near then end of this paper, then go back and read about the details of how to create them. They provide a visual context for the code presented here.

We are interested in presenting the proportion of women employed by local government at the state and county level. Perhaps a lawsuit alleges that in Cook County, Illinois, where Chicago is, women are under-represented in local city government. Maps can provide insight into this question.

The response datasets, which had male/female counts at local government, were downloaded from the Census Web site and read into SAS. The Census site used was <http://factfinder.census.gov>

```

data state_cen;
  set outdata.states (keep=geo_name P051008 P051040);
  pct=round(100*P051008/sum(P051008, P051040),1);
  format pct 2.;
  statel=upcase(geo_name);
  if statel =:"PUERTO RICO" then delete;
run;

proc sort data=states;
  by statel;
run;

proc sort data=state_cen;
  by statel;
run;
proc means data=state_cen nway noprint;
  var P051040 P051008;
  output out=s sum=;
run;

```

The macro variable below will be used in the title of the USA Map since displaying the overall percentage across the USA is a useful benchmark:

```
data s;
  set s;
  pct=round(100*P051008/sum(P051008, P051040),1.);
  tot=sum(of P051008, P051040);
  call symput ('pct',put(pct,4.1));
run;

%let pct=%cmpres(&pct);;
```

At this point we have a response dataset for the states plus a summary overall percentage.

We are now ready to look go at Illinois. Occasionally, you will have to use TRANWRD or other more cumbersome recoding in order to get the county names to match up between the SAS MAPS.COUNTIES and the response data. If the ID is not the same between the datasets, then one is forced to use county name.

```
data il;
  set outdata.county (keep=geo_name P051008 P051040);
  state=17;
  countynm=upcase(tranwrld(geo_name,'County, Illinois',''));
  pct=round(100*P051008/sum(P051008, P051040),1);
  format pct 2.;
run;
```

Now, let us do a proc print to see what the numbers look like for Illinois:

County	% female	% range
CASS	41%	40%- <45%
CHAMPAIGN	42%	40%- <45%
CHRISTIAN	42%	40%- <45%
CLARK	55%	50%+
CLAY	37%	35%- <40%
CLINTON	50%	50%+
COLES	44%	40%- <45%
<b>COOK</b>	<b>46%</b>	<b>45%- &lt;50%</b>
CRAWFORD	44%	40%- <45%
CUMBERLAND	40%	40%- <45%
deleted rows	deleted rows	
<b>ILLINOIS, OVERALL</b>	<b>44%</b>	

This table, while descriptive, is not particularly illustrative. Instead, let us look as some samples of maps under: <ftp.sas.com/techsup/download/sample/graph/gmap-examples-list.html> The USA Map example used here is based on one of the sample maps.

Prior to version 9, one had to download and use a maplabel macro to annotate maps, like overlay them with cities or values of the fields in addition to the colors. In version 9, these macros are no longer necessary. <ftp.sas.com/techsup/download/graph/maplabelmacros.txt>

Before we actually start creating graphs, some graphics options ought to be set. For example one should use color patterns that go from lighter to darker for readability since otherwise colors scheme will not be meaningful. Also, we

could increase the number of pixels in order to get maps with better resolution. Note below MAPS.USCENTER and MAPS.COUNTIES are existing SAS supplied datasets. For that matter, the various countries and a world map can be found on the MAPS library.

```

options ftext=swiss ftitle=swiss xpixels = 1600 ypixels = 1200 ;

pattern1 value=solid color=grayff; /* white */
pattern2 value=solid color=grayee;
pattern3 value=solid color=graycc;
pattern4 value=solid color=gray99;
pattern5 value=solid color=gray77;
pattern6 value=solid color=gray44;
pattern7 value=solid color=black; /* dark gray */

data states;
  set maps.uscenter(where=(fipstate(state) not in ( 'GU','PR')) drop=long lat);
  statel=stname(fipstate(state));
run;

```

We want to overlay the state percentage of local workers who are female on each state so we merge this census percentage to the SAS supplied datasets.

```

data states;
  merge states state_cen;
  by statel;
run;

```

This following code creates annotation for the states. It is taken from <http://ftp.sas.com/techsup/download/sample/graph/gmap-pop.html>

```

data maplabel;
  length function $ 8;
  retain flag 0 xsys ysys '2' hsys '3' when 'a' style 'swissb';
  set states ;
  by state;
  if state_app=. then state_app=0;
  function='label';
  text=put(pct,4.1); *<== this is the place were we place the annotated text;
  size=2.5; position='5';

  /* The FIPSTATE function creates the label */
  /* text by converting the FIPS codes from */
  /* MAPS.USCENTER to two-letter postal codes. */

  if ocean='Y' then
  do;
    position='6'; output;
    function='move';
    flag=1;
  end;

  /* If the labeling coordinates are outside the state (OCEAN='Y'), Annotate */
  /* adds the label and prepares to draw the leader line. Note: OCEAN is a */
  /* character variable and is therefore case sensitive. OCEAN='Y' must specify */
  /* an uppercase Y. */

  /* When external labeling is in effect, Annotate */
  /* draws the leader line and resets the flag. */
  else if flag=1 then
  do;
    function='draw'; size=.5;

```

```

        flag=0;
    end;
    output;
run; * state percentages;

```

We are now ready to plot a US Map. Before we do that, it might be a good idea to create a format for the percentage field rather than rely on SAS's ranges.

```

proc format;
    value pct
    0-<35='<35%'
    35-<40='35%<40%'
    40-<45='40%<45%'
    45-<50='45%<50%'
    50-high='50%+';
Run;

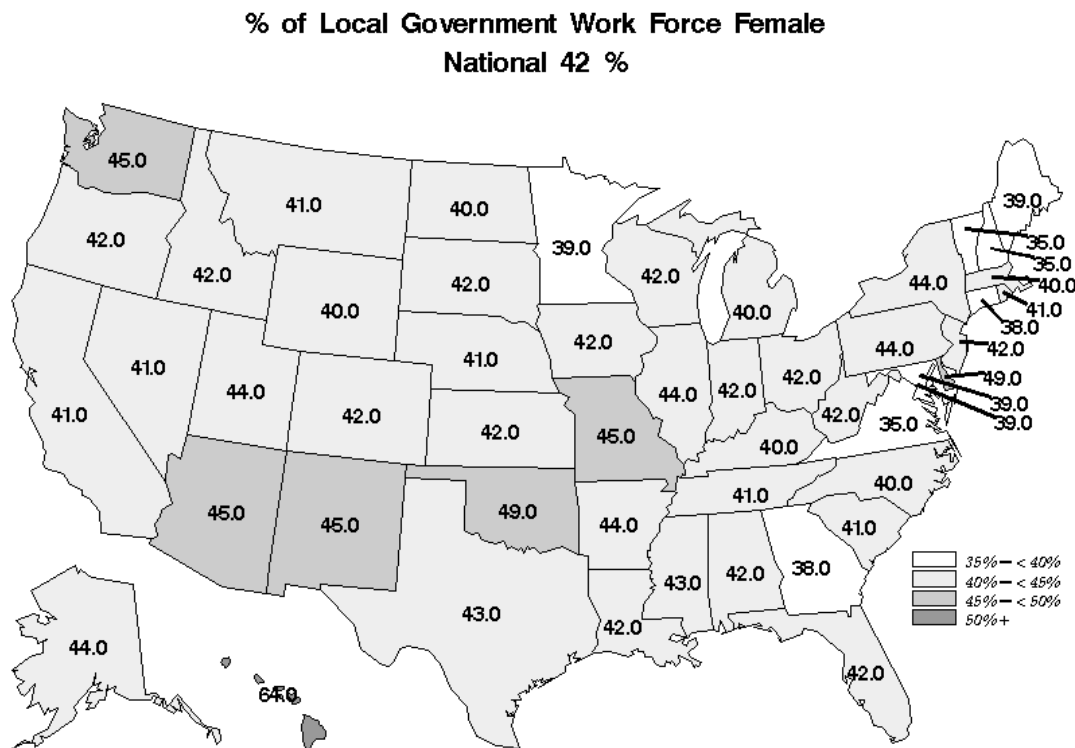
```

Note that we use DISCRETE as option below because we have applied an format to the percent field. Running it without the discrete option will result in SAS assigning the ranges, even if they are formatted, which will create a confusing formatted legend.

```

ods html file      = 'maps.htm';
proc gmap data=states map=maps.us;
    format pct pct.;
    title f=swissb height=15 pt "% of Local Government Work Force Female";
    title2 f=swissb height=15 pt "National &pct %";
    id state;
    choro pct /      outline=black discrete annotate=maplabel name="usa";
run;

```



Note that from the &pct macro field displayed in the title, we can see that nationwide, women comprise 42% of the local government work force. This percentage is a little lower on the East Coast. Hawaii is quite an exception with 64%.

Now we can concentrate on Illinois. As a first step, we need to overlay the percentages to the county level MAPS dataset, then highlight Cook County, where Chicago is, and finally plot the map.

```

data pct;

```

```

merge pcts (in=p) maps.counties ;
by state county;
if p;
run;

proc sort data=il;
by countynm;
run;

proc sort data=pcts;
by countynm;
run;

data pcts;
merge il pcts;
by countynm;
if county=31 then cook=1; else cook=0;
run; * highlight Cook (Chicago area) county;

proc sort data=pcts;
by state county;
run;

proc gmap data=pcts map=pcts all;
title1 f=swissb height=15 pt "Illinois";
id state county;
choro cook / name="il_1" nolegend coutline=black ;
run;

```



However, the map is unprojected so it first needs to be projected.

```

proc gproject data=pcts out=pcts;
id state county;
run;

```

Illinois, Cook County Highlighted

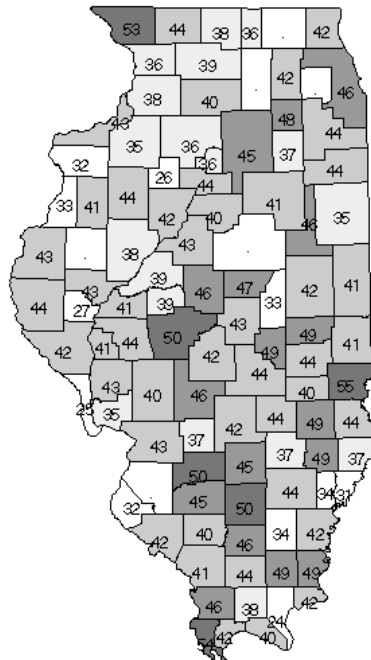


Now we annotate our map of Illinois as seen below. The ranges for the percentages could be changed using a proc format statement.

```
%annomac; * run this before maplabel;
%maplabel(pcts,pcts,anno,pct,state county,font='swiss',color=black);

proc gmap data=pcts map=pcts all;
  format pct pctc.;
  title1 f=swissb height=15 pt "% of Local Government Work Force Female";
  id state county;
  choro pct / annotate=anno name="il_2" discrete coutline=black ;
run;
```

% of Local Government Work Force Female



pct     <35%     35%-<40%     40%-<45%     45%-<50%     50%+

One can see that Cook County female work force in the local government is 46%, higher than for other many other counties in Illinois and higher than the national rate of 42%.

## **CONCLUSION**

SAS Maps provide a powerful visual tool to displaying results. One downside with maps is that while they might look good in color, they may lose some of their luster in black and white or vice versa. The same could be said about printing them out versus seeing them on screen. Nonetheless, they provide a powerful visualization tool for geographic data.

## **CONTACT INFORMATION**

Your comments and questions are encouraged. Contact the author for the program and data presented in this paper at:

Gur Hoshen  
American Institutes for Research  
1791 Arasratadero Road  
Palo Alto, CA 94304  
Work Phone: (650)843-8113  
Fax: (650)493-3550  
Email: [ghoshen@yahoo.com](mailto:ghoshen@yahoo.com), [ghoshen@air.org](mailto:ghoshen@air.org)  
Web:

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.