Paper 2346-201

Wow! You Did That Map With SAS®?! Round II

Louise S. Hadden, Abt Associates Inc.

ABSTRACT

This paper explores the creation of complex maps with SAS® software. This presentation incorporates explores the wide range of possibilities provided by SAS/GRAPH and polygon plots in the SG procedures, as well as replays, overlays in both SAS/GRAPH and SG procedures, and annotations including Zip Code level processing. The more recent GfK maps now provided by SAS, that underlie newer SAS products such as Visual Analytics as well as traditional SAS products, will be discussed. The pre-production SGMAP procedure released with Version 9.4 Maintenance release 5 will be discussed in context.

INTRODUCTION

The timing of this paper / presentation was fortuitous, since it occurred after the release of SAS® 9.4 M5. A long-awaited new SAS mapping procedure, PROC SGMAP, is pre-production in 9.4 M5. The advent of PROC SGMAP highlights many of the building blocks that have been developed and honed to perfection in the years of mapping with SAS/Graph and then the creation of maps with PROC SGPLOT and the POLYGON statement.

SAS/Graph is a separate SAS product and must be licensed separately or combined with other SAS products. There are a number of procedures, and some SAS provided data, that are part of the SAS/Graph product that are not available to Base SAS users. The SG procedures, on the other hand, are available to BASE SAS users (including SAS University Edition and SAS Studio® users). PROC SGMAP integrates some of the procedures from SAS/Graph into base SAS, with more to come with further version releases.

We will walk through the main resources for creating maps with SAS, and compare and contrast the three primary methods for creating maps in SAS. Examples demonstrating the various procedures and methods in more complex map samples using SAS/Graph, including zip code mapping, will be provided as well as a few examples using PROC SGPLOT and PROC SGMAP. Major topics include PROC MAPIMPORT, GfK maps, and zip code processing. Snippets of code are used for illustrative purposes: full code is available from the author by request.

RESOURCES

SAS has provided myriad resources for intrepid SAS mappers in addition to the actual SAS procedures that can create maps and geocode, and helpful data sets. SAS recently incorporated SAS/Graph mapping documentation into the standard documentation offerings, and the SG Procedures are also well documented. Primary resources include SAS Maps Online (briefly described below), map data sets in the SASMAPS and GFKMAPS folders provided to licensed SAS/Graph users, useful data sets and format catalogs in the SASHELP folder, geographic SAS functions, SAS blog posts created by Sanjay Matange, Ed Odom, and Robert Allison, Robert Allison’s fantastic demo website, SAS books and papers written by SAS staff and users (references included at the end of this paper.)

SAS MAPS ONLINE

SAS MAPS ONLINE is truly a buried treasure. It is a SAS-provided web page with a number of useful tools regarding mapping, including both traditional and the GfK map data sets and updates, related data sets (such as SASHELP.ZIPCODE, SASHELP.MAPFMTS, the new GfK WORLD_CITIES_ALL, and WORLDCTS), and sample programs. SAS MAPS ONLINE archives older map data sets and versions of SASHELP.ZIPCODE so that users can match their response data and/or software version appropriately. The zip code files and some other files do change over time so it is important to visit SAS MAPS ONLINE on a regular basis to download newer versions of files, or, you may wish to download an older file which matches the timing of your response data or software.   Or maybe you’d just like to see the cool new samples that are available. You can subscribe to an RSS feed so that you will be notified of updates on the page. This is essential as files are refreshed periodically (the zip code files are refreshed every quarter, for example.) SAS MAPS ONLINE files available on your SAS installation are NOT automatically updated – you (or your system admin) must perform updates manually. Note that in the current maps page only GFK maps from SAS 9.3M2 (2012) are presented – in order to obtain older versions (non-GFK) maps you must download files from the archive pages. Current GfK maps are provided updated with new installations/versions of SAS (GfK maps are not always updated for maintenance releases.) Older (non-GFK) maps are NOT being updated, and eventually will be phased out. Unlike the older map data sets, you must have a current license for a SAS product that includes SAS/GRAPH in order to download any GfK maps. GfK maps will be discussed further below.

SAS DOCUMENTATION

Recently SAS has integrated information relating to SAS/GRAPH mapping into the technical support documents located at support.sas.com at:

https://support.sas.com/documentation/cdl/en/grmapref/69722/HTML/default/titlepage.htm (HTML) and
https://support.sas.com/documentation/cdl/en/grmapref/69722/PDF/default/grmapref.pdf (PDF)

A sub-page includes list of recommended reading. This has been a long time coming and very exciting (I personally have been asking for it for years)! SAS MAPS ONLINE is still your primary resource for updated map data sets and other important information.

MACROS, FUNCTIONS AND FORMATS, OH MY

SAS provides a collection of macros originally designed for the traditional map data sets called %ANNOMAC. More information on %ANNOMAC can be found at http://support.sas.com/documentation/cdl/en/graphref/63022/HTML/default/viewer.htm#annodict-using-macro.htm. %GEODIST (shown below) provides a method to calculate distance for users with versions below SAS 9.2.

```
/* calculate the distance between zip code centroids */
%macro geodist(lat1,long1,lat2,long2);
%let pi180=0.0174532925199433;
7921.6623*arsin(sqrt((sin((&pi180*&lat2-&pi180*&lat1)/2))**2+
cos(&pi180*&lat1)*cos(&pi180*&lat2)*(sin((&pi180*&long2-
&pi180*&long1)/2))**2));
%mend;
```

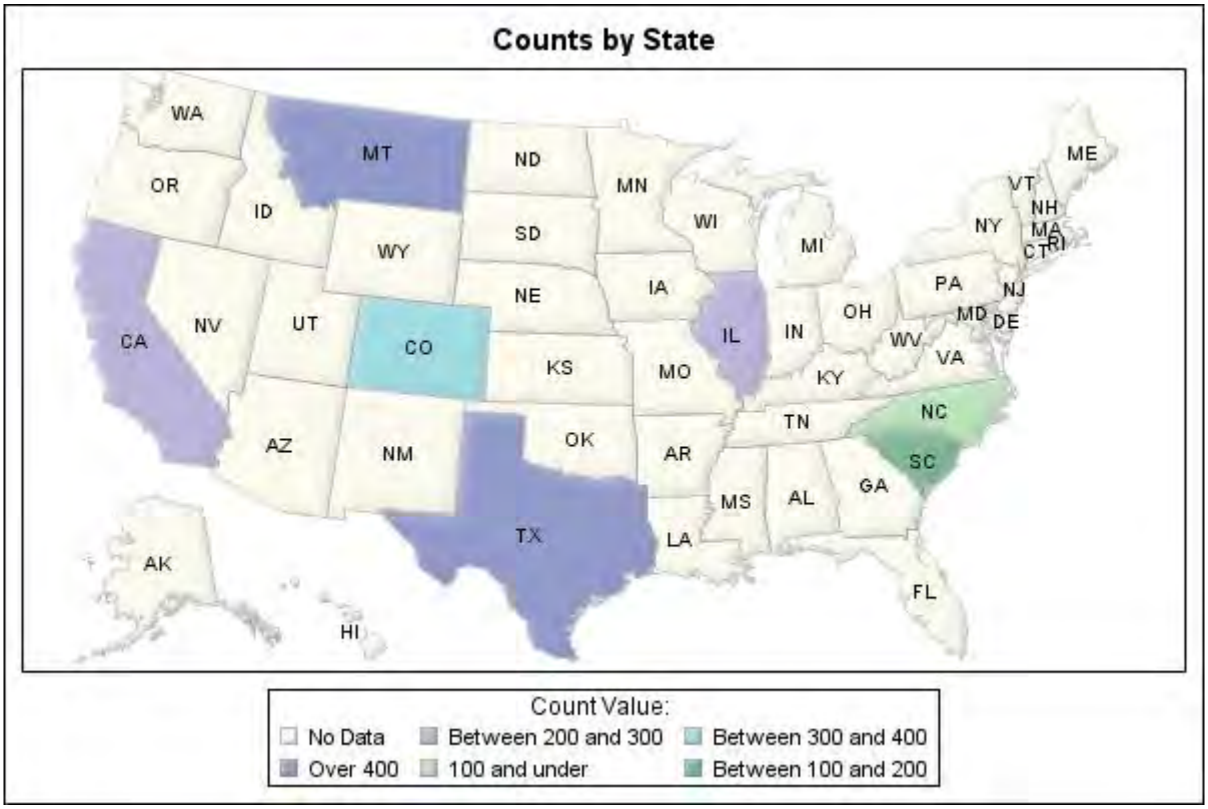
SASHELP.ZIPCODE (the basis of a number of SAS geographic functions) and SASHELP.MAPFMTS are available in the eponymous folder SASHELP.

Example 1: Featuring %ANNOMAC and PROC SGPLOT / POLYGON

This example, from SAS sample 53367, uses the %CENTROID macro within the %ANNOMAC collection of macros, to create the position for state labels on a US map. The map was created using PROC SGPLOT with a polygon statement. A code snippet follows below.

```
/* Make the SAS/GRAPH annotate macros available */
%annomac;

/* Calculate the center of each polygon to be used to place a label */
%centroid(us,centers,id);

/* Define a label to be placed at the center of each polygon */
data all;
    set all centers(rename=(x=xcen y=ycen) in=a);
    /* Define the variable to contain the label for each polygon */
    if a then label=fipstate(substr(id,4,2));
    /* Adjust a few label coordinates */
    if label = 'ID' then ycen + -.025;
    . . .
    if label = 'DE' then do;
        ycen + -.005;
        xcen + .005;
    end;
```

```
  run;
```



Example 2: Featuring GEOGRAPHIC SAS FUNCTIONS

As of SAS version 9.1, SAS-written functions based on data in the SASHELP.ZIPCODE file include ZIPCITY, ZIPSTATE, ZIPNAME, ZIPNAMEL and ZIPFIPS. For example, the ZIPCITY function takes ZIPCODE as its argument and returns a proper case city name and a two character postal state abbreviation.

ZIPCITY(‘02138’) returns ‘Cambridge, MA’

ZIPSTATE(‘02138’) returns ‘MA’

ZIPNAME(‘02138’) returns ‘MASSACHUSETTS’

ZIPNAMEL(‘02138’) returns ‘Massachusetts’

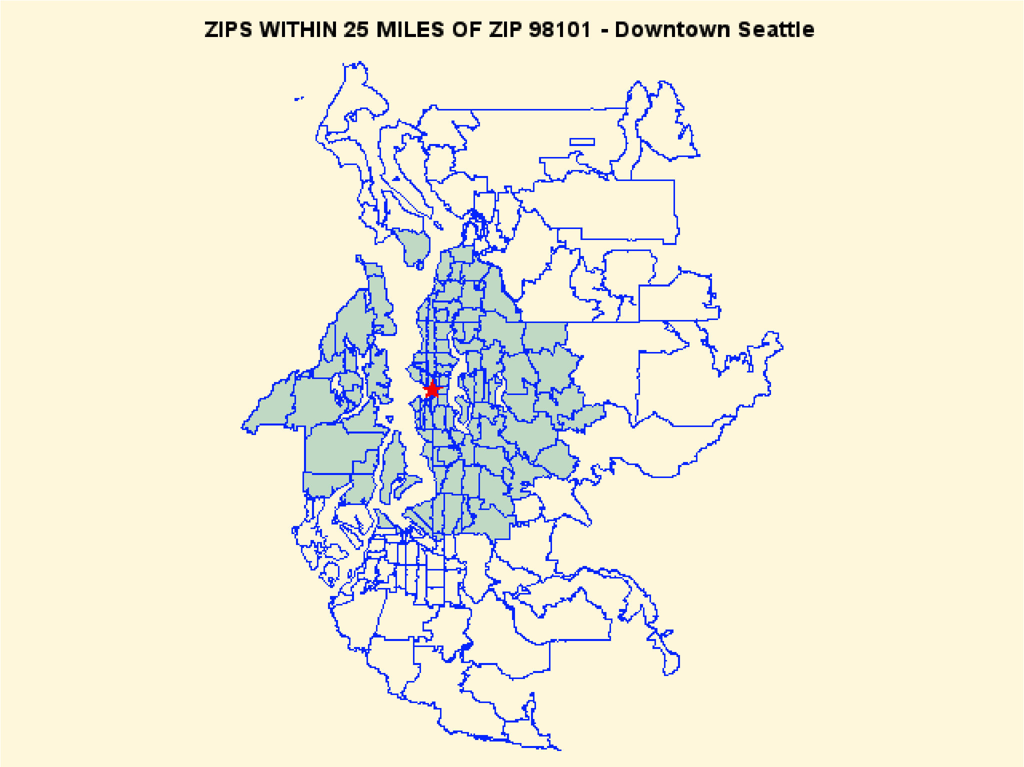
ZIPFIPS(‘02138’) returns 25 (FIPS state code for Massachusetts)

In 9.2, SAS provides functions to calculate distance between longitude and latitude pairs with the addition of the ZIPCITYDIST() and GEODIST(). ZIPCITYDIST() calculates the number of miles between 2 US Zip Codes. GEODIST() allows the user to specify 2 sets of longitude and latitude coordinates, and calculates the distance between the pairs. You can specify the units for the results, such as miles or kilometers. ZIPCITYDIST() only works on US Zip Codes, while GEODIST() can be used for coordinates anywhere around the globe.

```
  %let zip1=27513;
  %let zip2=21202;
  /* latitude and longitude coordinates */
  proc sql;
      create table zip1 as select * from sashelp.zipcode where zip=&zip1;
      create table zip2 as select * from sashelp.zipcode where zip=&zip2;
      select x into :long1 from zip1;
      select y into :lat1 from zip1;
```

```
        select x into :long2 from zip2;
        select y into :lat2 from zip2;
    quit;
    run;
    %let city1=%sysfunc(zipcity(&zip1));
    %let city2=%sysfunc(zipcity(&zip2));
    %let zipdist=%sysfunc(zipcitydistance(&zip1,&zip2));
    %let geodist=%sysfunc(geodist(&lat1, &long1, &lat2,
        &long2, 'M'),comma10.5);
```

The ZIPCITYDISC() function can be used in conjunction with PROC GMAP, PROC MAPIMPORT, Annotate and the SASHELP.ZIPCODE file to highlight zips within 25 miles (or the distance of your choice) from a central point (zip code.) The red star indicates the central zip code centroid. Such maps could be used to graphically demonstrate sales penetration in target areas, map out sales regions or delivery areas, etc.



Example 3: Featuring SASHELP.MAPFMTS and PROC GMAP

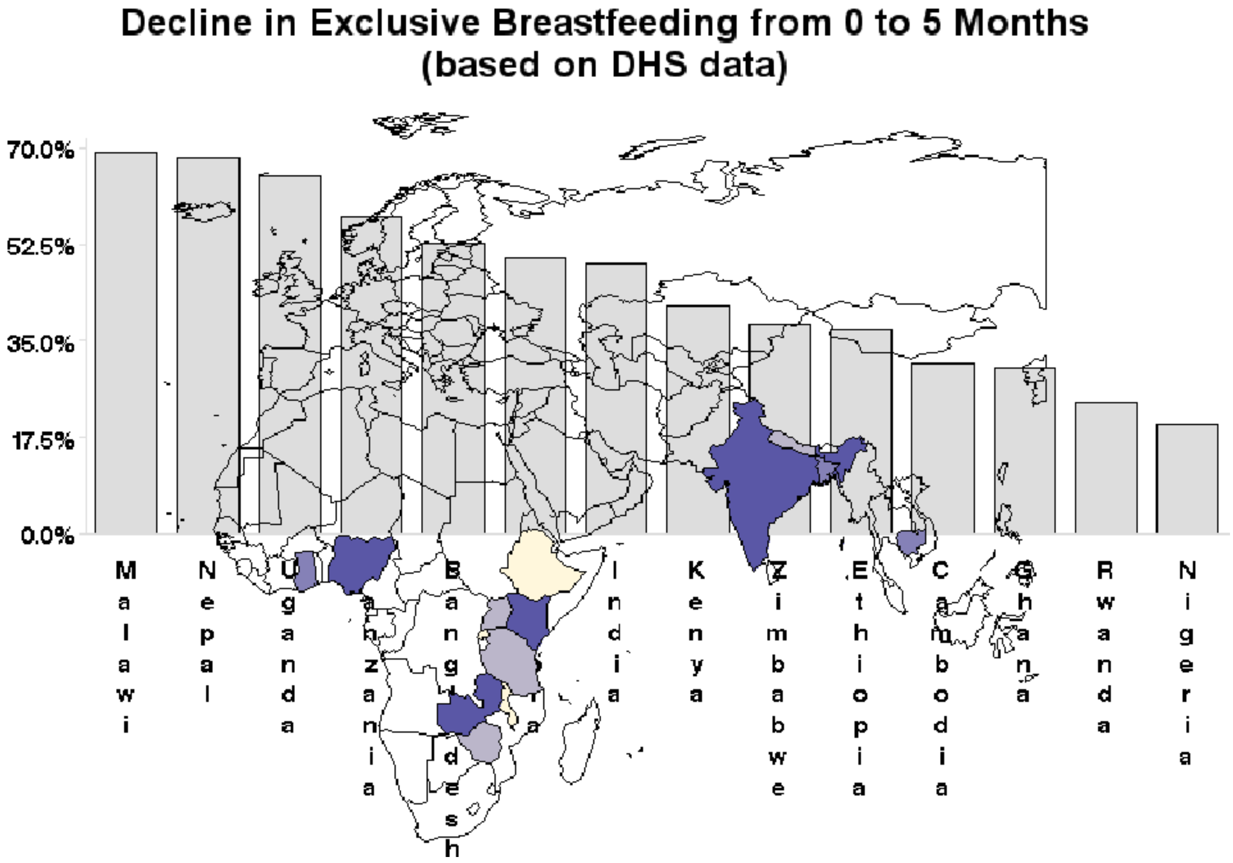
This example uses PROC GREPLAY and transparency to overlay a graph and map. It also utilizes a relatively obscure SAS-provided dataset, SASHELP.MAPFMTS, to produce current world country names matching the world map data set. This is one of many helpful data sets in the SASHELP folder. To use one of the formats for maps, you must specify the SASHELP.MAPFMTS catalog on the FMTSEARCH= option on a SAS OPTIONS statement:

```
    options fmtsearch=(sashelp.mapfmts);
    data work.world;
        length country_name $20;
        set work.world;
        country_name=put(country,glcnsm.);
        country_name=propcase(country_name);
    run;
```

For more information on SASHELP.MAPFMTS, see

http://documentation.sas.com/?docsetId=grmapref&docsetTarget=n0xl3c2yufdagmn1d6u7o5idsemu.htm&docsetVersion=9.4_01&locale=en

Other features of the full program are use of PROC TEMPLATE to define colors used in the GMAP procedure, reduction of the map area and density using PROC SQL; a PROC GMAP (using nodisplay) to create the underlying map; a PROC GCHART to create the overlay bar chart (again using nodisplay), and a PROC GREPLAY to display both images in the same space. The HTML file accompanying the PNG file has additional functionality. Mousing over the colored countries provides additional information on breastfeeding rates and GNI in a tooltip for each country.



SOME SAS MAPPING BASICS

PRIMARY SAS PROCEDURES FOR MAPPING

Below are listed the most commonly used SAS procedures for mapping. A full discussion of all the procedures listed here is out of scope for this paper and PROC GMAP, PROC SGPLOT, PROC SGMAP and PROC MAPIMPORT will be primary subjects of discussion, although other procedures such as PROC GREMOVE may be used incidentally in examples.

SAS Procedure	In SAS/Graph	In Base (9.4M5)	In Base (Future)
PROC GMAP	Yes	No	No
PROC SGPLOT / POLYGON statement	No	Yes	Yes
PROC SGMAP	No	Yes	Yes
PROC MAPIMPORT	As part of Base	Yes	Yes
PROC GEOCODE	As part of Base	Yes	Yes
PROC GREDUCE	Yes	No	Due for 9.5
PROC GREMOVE	Yes	No	Due for 9.5
PROC GPROJECT	Yes	No	Due for 9.5
PROC GINSIDE	Yes	No	Due for 9.5
PROC GREPLAY	Yes	GTL replaced	GTL replaced

Table 1. Availability of SAS Mapping Procedures by Product

MAPS, PLOTS, AND SYNTAX, OH MY

All SAS mapping procedures require a map data set. In general, a SAS map data set is “matched” with a response data set that contains data relating to particular geographies described on the map, with matching geographic identifiers. The map data set and the response data set can be one and the same.

High quality GfK Geocoding maps have been provided by SAS since SAS 9.3 M2, as sources for inexpensive map data dried up. SAS has been including both GfK (found in libname GfKMaps) and "traditional" SAS map data sets (found in libname Maps or SASMaps) with licenses for SAS/GRAPH for some time, recognizing there will need to be an extended transitional period. The "traditional" SAS map data sets are no longer being updated. If you visit SAS MapsOnline, you will find only GfK maps in current maps. You must have a licensed copy of SAS/Graph in order to download GfK maps, as SAS’s agreement with the vendor only allows those sites that have purchased SAS/Graph to access the GfK maps. Most SAS/Graph users have been gradually renovating their code to accommodate the new GfK maps. There are several good papers on GfK maps listed in the references and examples of before and after code renovation are provided below.

The fact that GfK maps are only available to SAS/Graph users becomes an issue as Base SAS users now have mapping procedures, but many do not have access to SAS-provided map data sets. For users of PROC SGPLOT with a POLYGON statement, and PROC SGMAP with a MAPDATA option and a CHOROMAP statement, Base SAS users can download the traditional map data sets from SAS Maps Online, with the caveat that these data sets have not been updated for years. SAS offers other options with the SGMAP procedure: backgrounds created by OPENSTREETMAP and ESRIMAP options (described below). With these two procedural options, all you need is your response data set with appropriate geographic data.

For both SAS/GRAPH and SG mapping procedure users, there are times when a data set at the appropriate geographic level is not available through SAS. A common example is zip code level or census tract level map data sets. The PROC MAPIMPORT procedure will be described in detail below with examples.

PROC GEOCODE and PROC GINSIDE deal more with geolocation than mapping so I won’t discuss here. PROC GREDUCE, PROC GPROJECT, and PROC GREMOVE help to manipulate maps to fit specific purposes or to improve a map’s appearance. Some of the examples provided below utilize these procedures. In the references section, I note several papers that are very informative regarding these additional procedures.

PROC MAPIMPORT

PROC MAPIMPORT has been available since SAS 9.2. This procedure enables you to import ESRI shapefiles and convert them into SAS map data sets that can be used by SAS/GRAPH, PROC SGPLOT and PROC SGMAP. MAPIMPORT in itself does not produce any graphics output, but does aid in the production of output from PROC GMAP and the SG “mapping” procedures.

There are innumerable SHP files available on the internet. SHP files usually come with companion files, two of which are necessary / useful for the PROC MAPIMPORT procedure. Make sure you extract the DBF file (metadata for the “map” file), the SHX file (shape information for the “map” file), and the SHP file (combines the metadata and shape information) from any SHP file zip files into the same directory as the SHP file.

There are a couple of procedure options (CONTENTS and CREATE_ID_, new in later versions of SAS version 9) and some statements (exclude, rename, id and select) available with PROC MAPIMPORT. Make sure you look at the latest version of the SAS documentation in order to find any information on CONTENTS and CREATE_ID_. These options are great time-savers, although there are some cons associated with the CREATE_ID_ option. The CONTENTS options shows information about the SHP files including field names and types in the SAS log. CREATE_ID_ creates a map ID variable named _ID_ with a unique value for each polygon in the map. Unlike SAS map data sets, imported SHP files do not automatically have an ID variable for polygons. This means you have to use an ID statement OR the CREATE_ID_ procedure option in order to have an ID. It is possible to create and use a SAS map data set without an ID statement, but generally output looks nicer with an ID variable. In general this procedure is a little chicken and egg – you need to run it to see what the contents are, explore what ID variables might be available and in play, and then ideally rerun it with an ID statement when you have ascertained what the ID variables are. If you get extraneous lines in your image it is because either you have not used a (correct) ID statement or the image is too detailed, in which case you need to run PROC GREDUCE on your newly created map data set.

Basic syntax for PROC MAPIMPORT

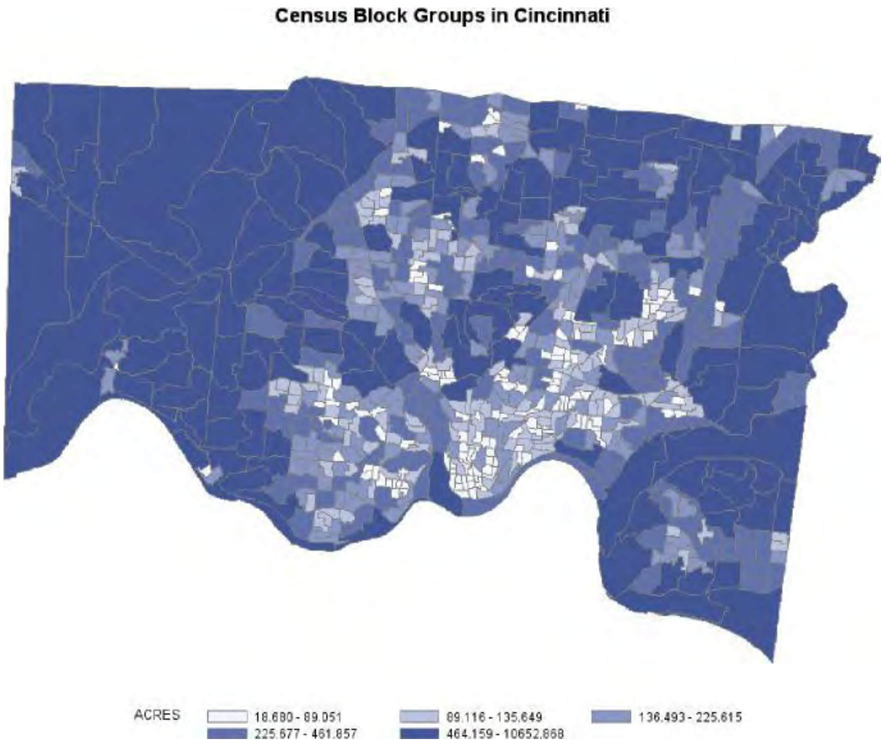
```
PROC MAPIMPORT
     OUT=map-data-set /* required – name of output map data set */
     DATAFILE='path-to-shapefile ' /* required – name and location
                                      of incoming shapefile */
     <CONTENTS> /* optional – produces a contents listing for the
                  shapefile */
     <CREATE_ID_> /* optional – creates a map ID variable uniquely
                    identifying polygons */
     ;
     <SELECT> 'field-identifier(s)'; /* optional, used to keep variables
                                        in the output map data set */
     <EXCLUDE> 'field-identifier(s)'; /* optional, used to drop
                                         variables in the output map
                                         data set */
RUN;
```

Several examples of using PROC MAPIMPORT are shown below. The DBF, SHX and SHP files should all be available in the same logical space.

Example 4: CENSUS BLOCK GROUPS IN CINCINNATI

The source of data used for this example is the Cincinnati Area Geographic Information System (CAGIS), which has a lot of useful SHP files and Cincinnati area information. SHP files often come with “mappable” data included: in this case, I chose acres as the response variable and 5 levels in my PROC GMAP code. I opted to use the CREATE_ID_ procedure option since I knew nothing about the SHP file in advance, and consulted the CONTENTS listing in the log (and looked at a test print).

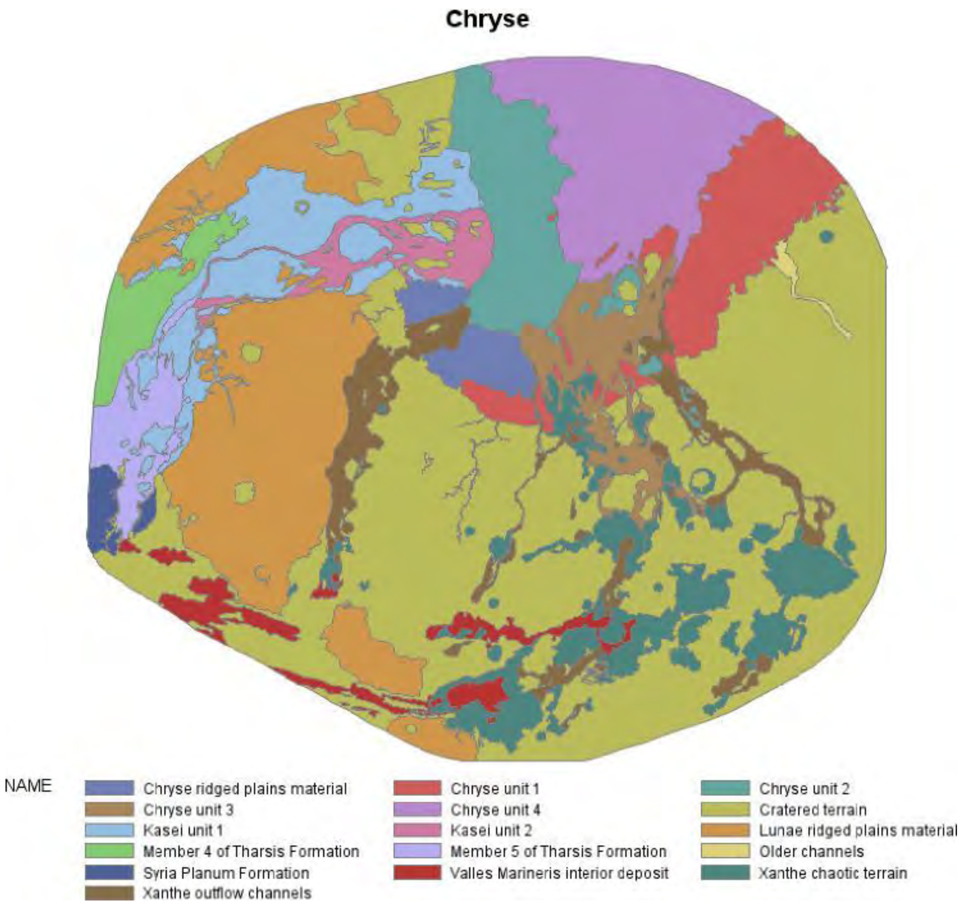
```
PROC MAPIMPORT DATAFILE='.\cenbgr2010.shp' OUT=cenbgr2010 CONTENTS
CREATE_ID_;
RUN;
PROC PRINT DATA=cenbgr2010 (OBS=5) NOOBS;
RUN;
ODS HTML FILE='test_cenbgr2010.html';
ODS PDF FILE='test_cenbgr2010.pdf' NOTOC;
GOPTIONS RESET=all DEVICE=png HSIZE=9in VSIZE=7.5in;
TITLE1 "Census Block Groups in Cincinnati";
PROC GMAP MAP=cenbgr2010 DATA=cenbgr2010 ALL;
    ID _id_;
    CHORO acres / COUTLINE=gray LEVELS=5 NAME='cenblkgr';
RUN;
QUIT;
ODS PDF CLOSE;
```



Example 5: PROC MAPIMPORT and Mars!

The source of the data for the Mars example is U.S.G.S. Planetary GIS Web Server (PIGWAD: Planetary Interactive G.I.S.-on-the-Web Analyzable Database.) The MARS image mapped was the Chryse area. Since I know nothing about this data, I used both CONTENTS and CREATE_ID_, and reviewed the log and the list. I used the SAS created _ID_ variable as the response variable and the name to identify the shapes.

```
PROC MAPIMPORT DATAFILE='.\chryse_dd.shp' OUT=chryse
    CONTENTS CREATE_ID;
RUN;
. . .
GOPTIONS RESET=all DEVICE=png HSIZE=9in VSIZE=7.5in;
TITLE1 "Chryse";
PROC GMAP MAP=chryse DATA=chryse all;
    ID id_;
    CHORO name / COUTLINE=gray NAME='Chryse';
RUN;
QUIT;
```



ZIP LINING

The SASHELP.ZIPCODE file, briefly mentioned above in the resources section, is a SAS data set containing ZIPCODE level information for the United States including ZIPCODE centroids (x, y coordinates), Area Codes, city names, FIPS codes, and more. The file is indexed on ZIPCODE to facilitate processing, and is updated on a regular basis by SAS. It is provided in transport format so that installations with dissimilar releases of SAS and operating systems can make use of the file. The source of the zip code data in the file is http://www.zipcodedownload.com/, which is a subscription service that users can purchase. It is available for free to licensed SAS users as a SAS data set through SAS MAPS Online.

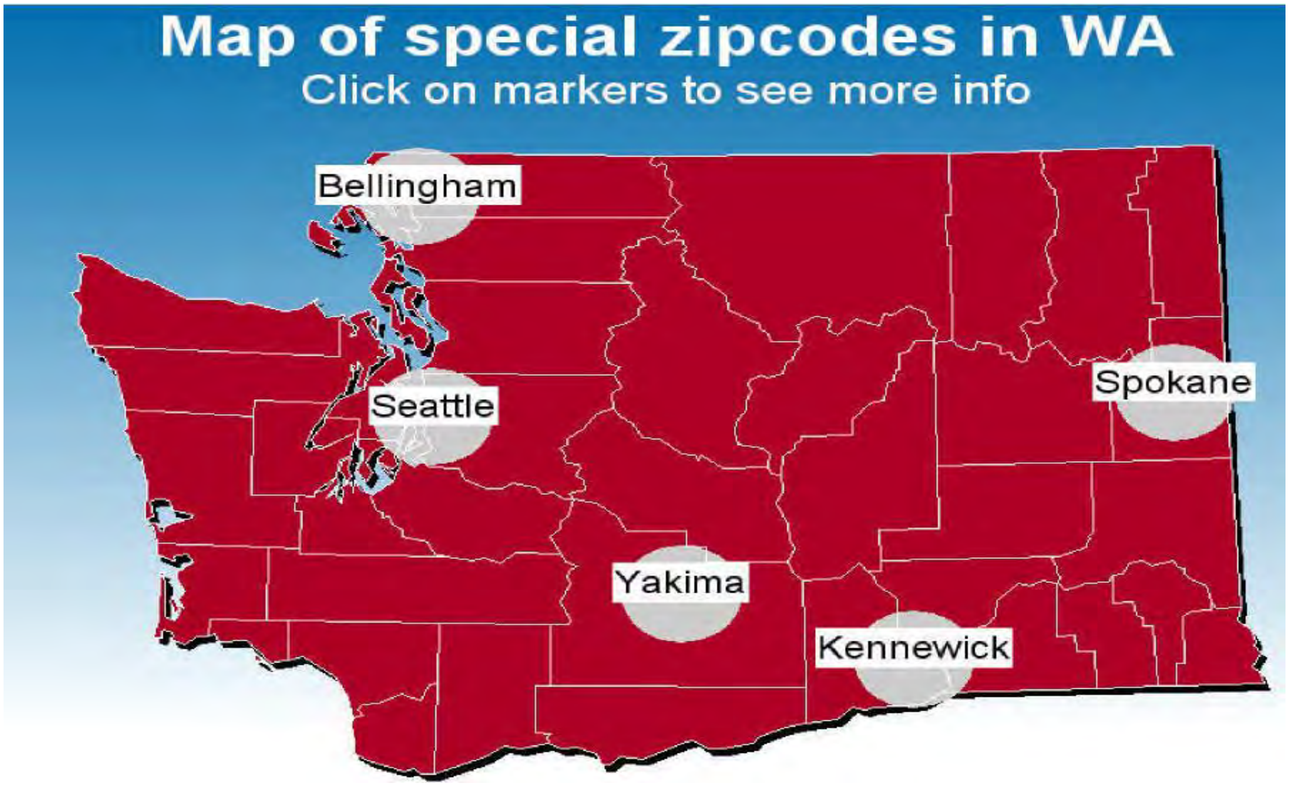
For users with SAS version 9 and above, the SASHELP.ZIPCODE file is located in your SASHELP folder, and may be accessed by using SASHELP as the library reference, just as you would access SAS map data sets by using MAPS as the library reference. Updated versions of SASHELP.ZIPCODE and SAS map data sets may also be obtained from SAS MAPS ONLINE for versions 8 and above.

POTENTIAL USES OF SASHELP.ZIPCODE

As shown above, the SASHELP.ZIPCODE FILE can be used to enable SAS-written functions and calculate distances between zip codes. It can also be used to annotate SAS/GRAPH maps with information in the file, to develop user-defined formats, and much, much more.

Example 6: Highlight Special Zips

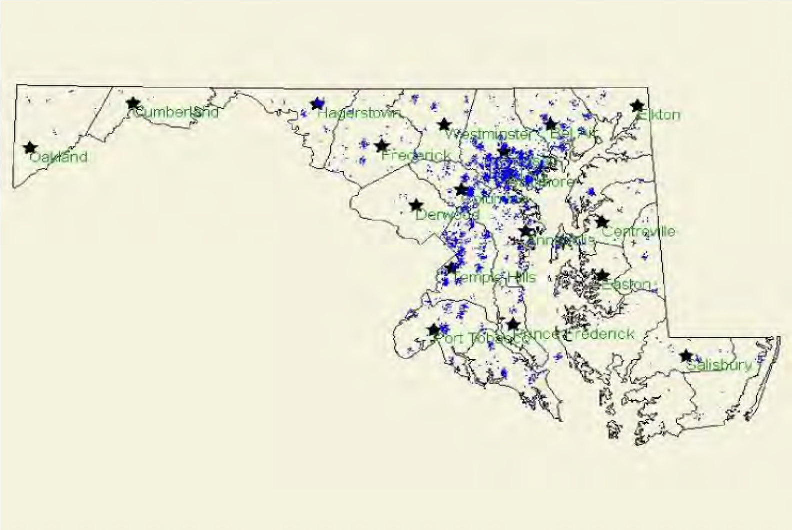
In this sample, PROC GMAP is used in combination with the Annotate facility to create a map of Washington with: fake transparent effects; 'shadows' behind the map; and a gradient shaded background. The example starts with a fairly basic map, a county map, with Annotate markers and labels, based on longitude/latitude coordinates from SASHELP.ZIPCODE. This map is based on a sample presented by Mike Zdeb and Robert Allison (for the state of Pennsylvania) at a Hands-On Workshop at SUGI 30. Then, the all of the 'extras' listed above are added.

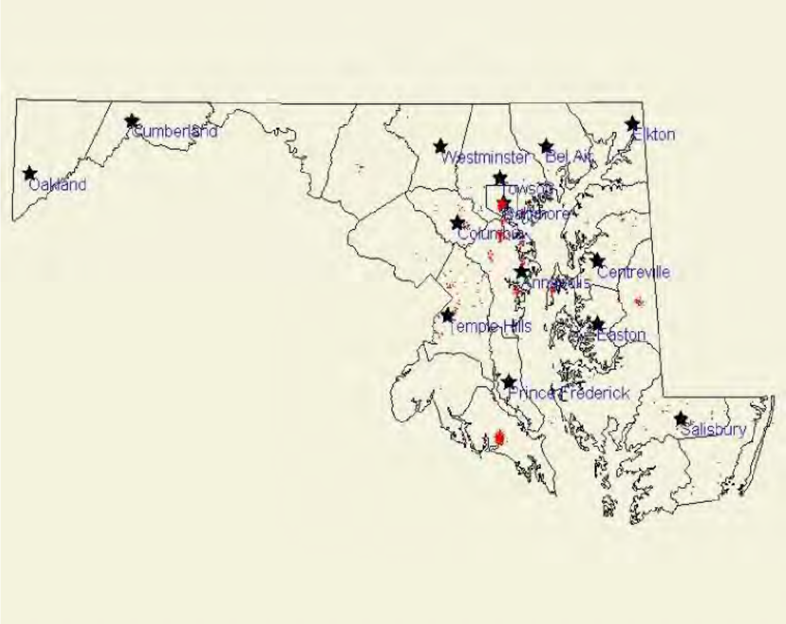

Map of special zipcodes in WA
Click on markers to see more info
Bellingham
Seattle
Spokane
Yakima
Kennewick

Example 7: Dot Density Maps

In 2006, Mike Zdeb and I collaborated on a paper, “ZIP Code 411: A Well-Kept SAS® Secret”, which won “Best Paper” at SUGI 31. In the years that have passed, there’s never been a month that I haven’t gotten a request for the code for the following examples. Mike and I ended up putting a zip file of code and sample data up on SASCOMMUNITY.ORG where it is still available today (read-only). My project involved using the zip code centroids provided in SASHELP.ZIPCODE in conjunction with a “dot density” mapping program supplied on the support.sas.com website written by Darrell Massengill of SAS (retired) to annotate county maps in Maryland. The blue dots represent Medicare beneficiaries

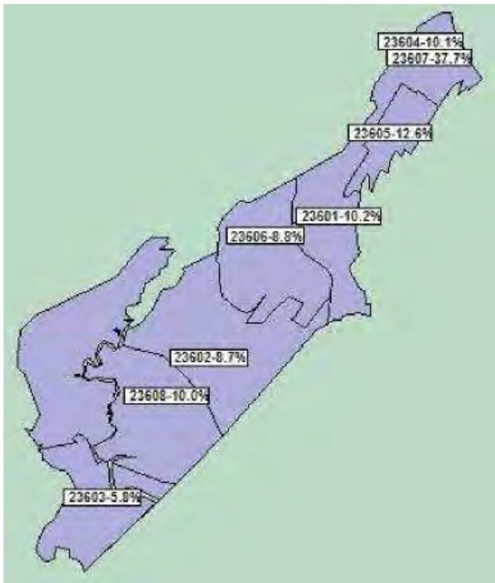
This type of map would be ideal to show sales penetration, sales, etc. The annotation (dots) can be color-coded and sized to match response data. The “dot density” effect is achieved by randomly distributing multiple records per zip code centroid with small adjustments.





Example 8: Annotate Response Data on a Zip Code Level Map

SAS's PROC MAPIMPORT was used to import the zip code boundary maps for selected communities for a study of beneficiaries in Medicare's Part D program. Poverty data from the Bureau of the Census were collected for the same communities and combined with the SASHELP.ZIPCODE file to obtain zip code centroids. The poverty data were used to annotate zip codes in the various communities with poverty rates attached to the zip code centroids.



The project team was then able to select zip codes with predominantly low incomes close to the focus group facility for sampling. This type of map could be used to inform locations of residential, industrial and commercial developments, as well as target sales, etc.

Example 9: Annotate Response Data on a Zip Code Level Map

The variables in the SASHELP.ZIPCODE file can also be used to create user-defined formats. Some potential user-written formats include: conversion of zip code to MSA code (and by extension, urban/rural status); conversion of zip code to FIPS county code; conversion of zip code to county name; conversion of zip code to latitude and longitude; conversion of zip code to telephone area codes; conversion of zip code to daylight savings time flag; and conversion of zip code to time zones. An example of creating a format to transform zip codes to MSA codes follows below.

```
DATA zip2msa (RENAME=(zipcode=start msacode=label));
    RETAIN fmtname '$zip2msa' type 'c' hlo=' ';
    SET sashelp.zipcode END=last;
    Msacode=PUT(msa,z4.);
    Zipcode=PUT(zip,z5.);
    IF last THEN hlo='h';
RUN;

PROC FORMAT LIBRARY=library CNTLIN=zip2msa;
RUN;

DATA x;
LENGTH msacode $ 4;
    Zipcode='02138';
```

```
      Msacode=put(zipcode,$zip2msa.);
  RUN;
PROC PRINT DATA=x;
RUN;
```

Result:

```
Obs    msacode      Zipcode

1      1120         02138
```

SAS PROVIDED GFK MAPS

GfK maps were new as of SAS 9.3 M2. Sources for inexpensive map data have become rare, leading SAS to enter into a partnership with a 3rd party source, GfK GeoMarketing, which specializes in map data. The vendor will update the map data sets periodically. The GfK maps require a lot of changes, but also have their benefits (not the least of which is greater accuracy!) For now, SAS releases are shipping with both the new GfK maps and the traditional map data sets, recognizing that there will need to be an extended transition period. Changes and usage tips are briefly discussed below.

LIBNAME CHANGES

The traditional SAS maps are currently the default maps. You can point to either map library if desired using the following code:

```
  LIBNAME MAPS(MAPSSAS);
  LIBNAME MAPS(MAPSGFK);
```

Or, you can simply refer to the MAPS or MAPSGFK library when using maps in your program.

FILE AND FILENAME CHANGES

- Added longer, more informative and consistent names (i.e. AFGHANISTAN vs AFGHANIS)
- Added NUTS level files for Europe
- Added _ATTR suffix for attribute files (ARGENTINA_ATTR vs ARGENTI2)
- Added PROJPARM (GPROJECT information for all projected maps)
- Eliminated some data sets (such as COUNTY and US2)
- Some maps will be at a different levels of political data (for example COUNTY instead of STATE)

VARIABLE CHANGES

- Many changes to variable names
- Added RESOLUTION variable
- ID variables are now character and unique world wide
- X and Y are ALWAYS projected
- Lat and Long are ALWAYS unprojected, based on the eastern Hemisphere and in degrees (PROC GPROJECT will require the DEGREES and EASTLONG options)

Example 10: Otter Country – Differences between GfK and Traditional SAS Maps

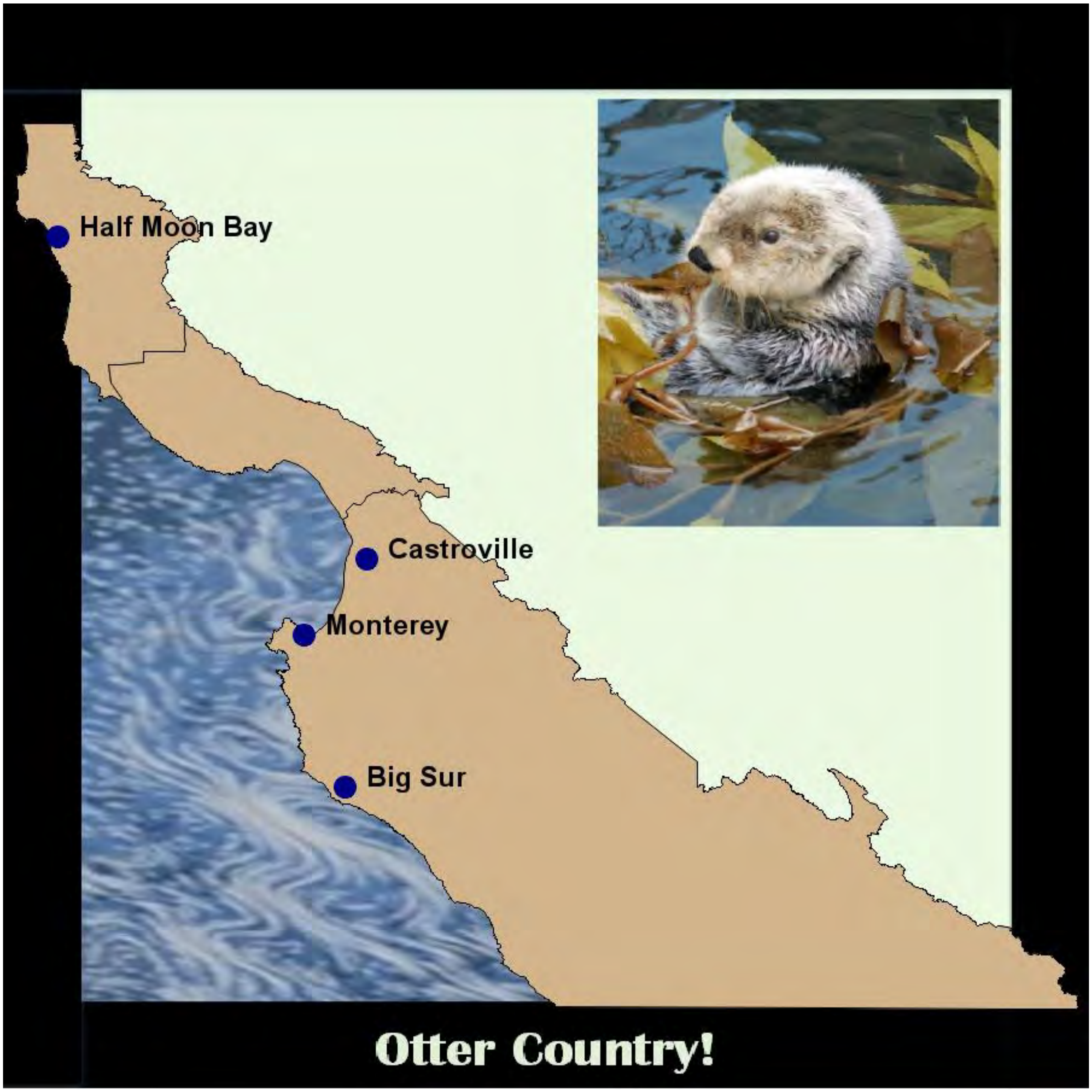
In this example inspired by Barbara Okerson, we work with a subset of counties in California. We'll start with the GfK map. It's a relatively simple task with the GfK maps to subset specific counties and annotate city names, etc. in your programs. X and Y are always projected, and LAT and LONG are always unprojected. You have the option of using the more robust ID variable used in the example below, or the customary ID variables that you used with the traditional map data sets.

First, three contiguous California coastal counties are pulled in from the GfK US_COUNTIES data set. We want to annotate several cities' names onto the output map, so we access the GfK USCITY data file. Because the X, Y variables have already been projected for the entire US in both files, we reset X, Y to LONG, LAT respectively in BOTH files so that they will match.

```
DATA camonterey;
    SET mapsgfk.us_counties (WHERE=(FIPSTATE(state)="CA"
        and county IN(053,081,087)));
    x=long;
    y=lat;
RUN;

DATA anno;
    SET mapsgfk.uscity (WHERE=(statecode="CA"
        and county_name IN("Monterey","San Mateo","Santa Cruz")
        and city IN('Big Sur','Half Moon Bay','Monterey',
            'Castroville')));
    LENGTH function $ 8;
        x=long; y=lat;
        xsys='2'; ysys='2'; hsys='3';
        when='a';
        function='pie';
        style='psolid';
        rotate=360;
        size=1;
        color='navy';
    OUTPUT;
        function='label';
        position='C';
        style=' ';
        rotate=.;
        size=2.5;
        color=' ';
        text=catt("   ",city);
    OUTPUT;
RUN;
```

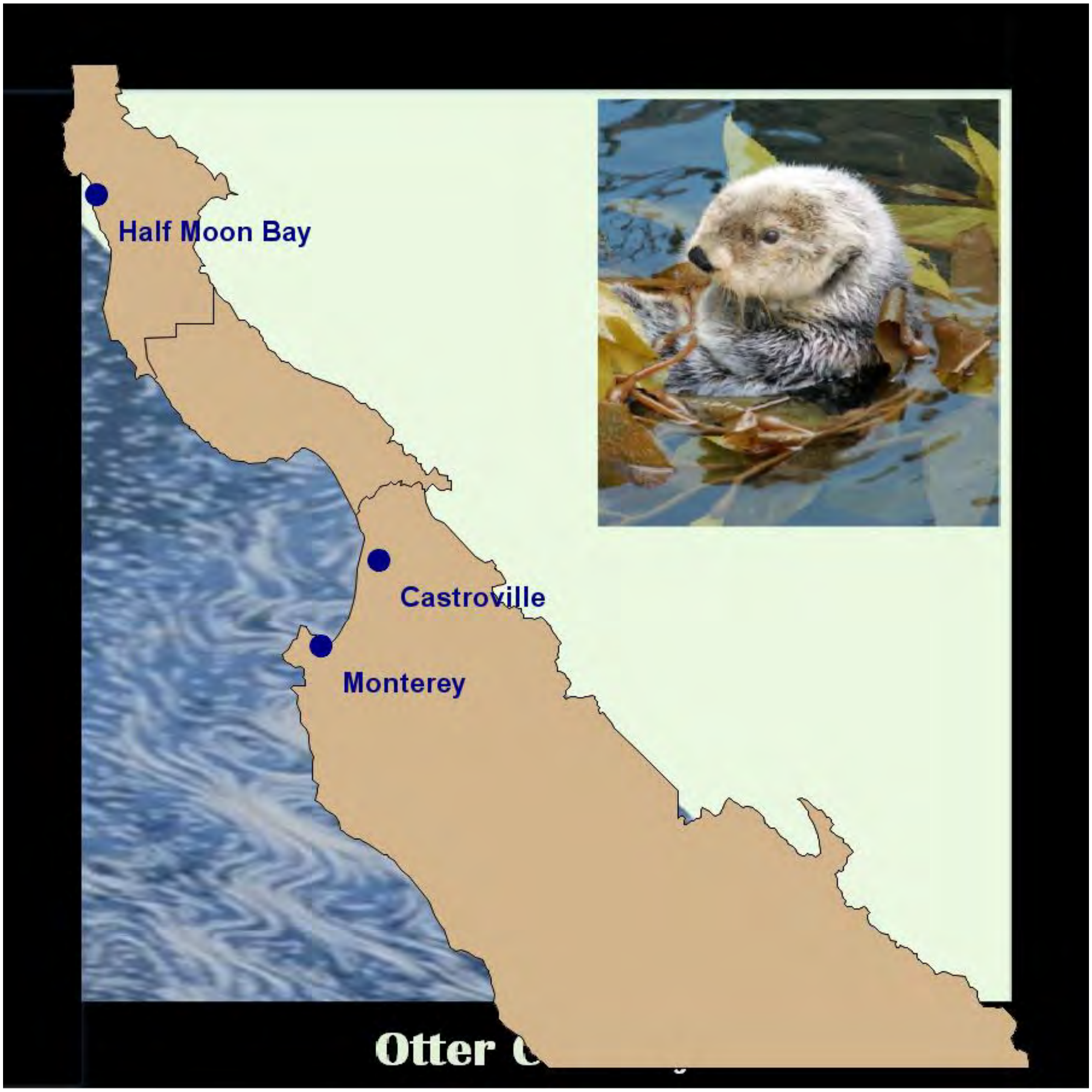
A map is generated. OPTIONS are set to output to a landscape page and suppress the printing of the date and page number; GOPTIONS are set to determine the type of image (PNG), insert a background image and format it, determine the units used in measurements of the image, and produce a border; the ODS listing destination is closed to suppress "windowing" of the graphic while running in batch, and ODS destination(s) (HTML) are selected. The result is shown below.

```
/*County Map GMAP code*/

GOPTIONS DEVICE=png IBACK='otterbackground.jpg' IMAGESTYLE=fit;
GOPTIONS XPIXELS=900 YPIXELS=900;
GOPTIONS border;

ODS HTML PATH=odsout BODY="gfkmap.htm" STYLE=sasweb;
PATTERN1 V=s C=tan R=3;

PROC GMAP MAP=CAmonterey DATA=CAmonterey ANNO=anno;
    ID id;
    CHORO state / DISCRETE NOLEGEND NAME="gfkmap";
RUN;
QUIT;

ODS HTML CLOSE;
```



Now, let's see how producing a similar map with traditional SAS maps works. In order to annotate the city names with traditional SAS map files, the x and y variables have to match, and the map and uscity data files must be combined and projected together. You may need to delve into the originating files to find out exactly what x, y, lat and long represent, and if/how any values are projected. Unlike the GfK map data sets which are consistent, there are no guarantees with the traditional map data sets. Once the combined data set is re-projected, the two input data sets are separated again and used as input to a PROC GMAP statement. As you can see, it will take some additional tweaking to get an output map that matches the GfK figure. The point here is that unless you are an expert who is familiar with traditional SAS map data sets, it will take you longer to achieve your goals than if you use the more consistent GfK maps.

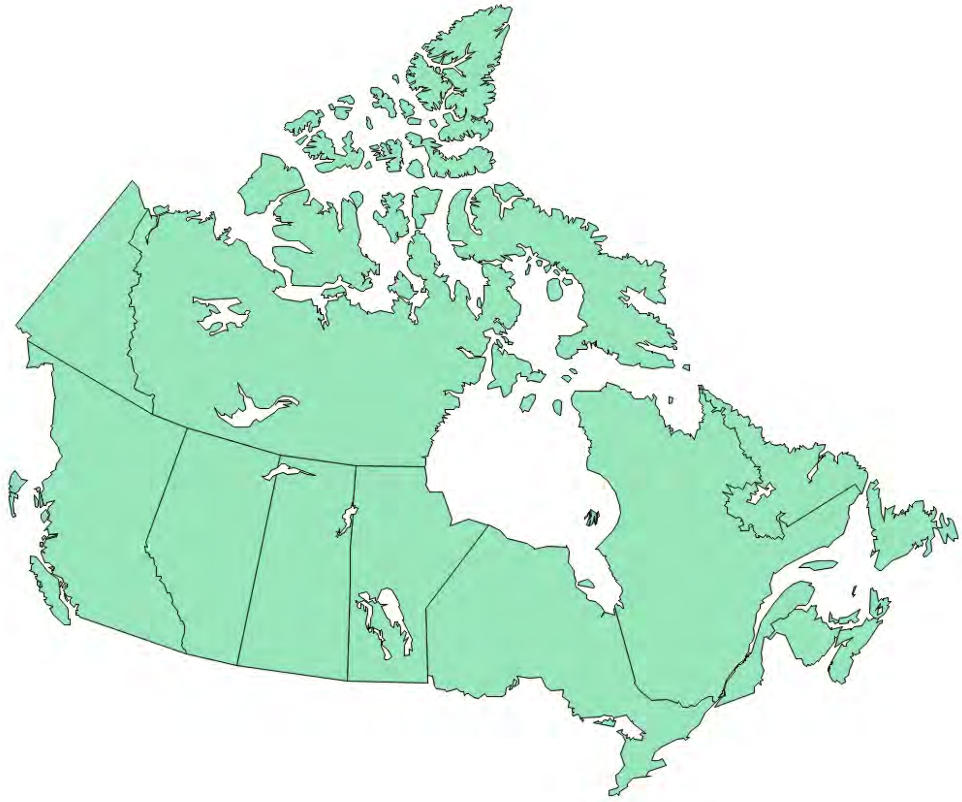
```
DATA annoc;
    SET maps.uscity (DROP=state WHERE=(statecode="CA" and county
        in(53,81,87) and city in('Big Sur','Half Moon Bay',
        'Monterey','Castroville')));
    LENGTH function $ 8;
    x=long*arcos(-1)/180;
    y=lat*arcos(-1)/180;
    county=1000;
    xsys='2';
    ysys='2';
    hsys='3';
    when='a';
    function='pie';
    style='psolid';
    rotate=360;
    size=1;
    color='navy';
    OUTPUT;
    county+1;
    function='label';
    position='9';
    style=' ';
    rotate=.;
    size=2.5;
    color='navy';
    text=catt("   ",city);
    OUTPUT;
RUN;

DATA all;
    SET maps.counties (WHERE=(state=6 and county IN(53,81,87))) annoc;
run;

PROC GPROJECT DATA=all OUT=allp PROJECT=gnomon;
    ID county;
RUN;
QUIT;

DATA citiesp montereyp;
    SET allp;
    IF county >999 THEN OUTPUT citiesp;
    ELSE OUTPUT montereyp;
RUN;

GOPTIONS DEVICE=png IBACK='otterbackground.jpg' IMAGESTYLE=fit;
GOPTIONS XPIXELS=900 YPIXELS=900;
GOPTIONS border;

ODS HTML PATH=odsout BODY="mont_std.htm" STYLE=sasweb;
PATTER1 v=s c=tan r=3;
PROC GMAP MAP=montereyp DATA=montereyp ANNO=citiesp;
    ID county;
    CHORO county / NOLEGEND NAME="mont_std";
RUN;
QUIT;

ODS HTML CLOSE;
```



Example 11: Oh Canada – Using PROC GREMOVE to Remove Inner Boundaries

In this example we head north to Canada, to map provinces. We begin with a review of the data sets available for Canada in both the MAPSGFK and MAPSSAS directories. We find that the MAPSGFK directory contains CANADA and CANADA_ATTR data sets. The MAPSSAS folder contains a number of data sets related to CANADA, called CANADA, CANADA2, CANADA3, CANADA4, and CANCENS. On the plus side for GFK maps, you only need to investigate one map and attribute data set. On the negative side, the one map that is available is designed to produce a census district level map, not a province level map. On the plus side for traditional map data sets, there's a map (CANADA2) which is designed to produce a province level map.

```
GOPTIONS DEVICE=png XPIXELS=1200 YPIXELS=900 BORDER;

ODS HTML PATH=odsout BODY="sasprov.htm" STYLE=sasweb;
PATTERN1 V=s C=vlig R=5000;
PROC GMAP MAP=mapssas.canada2 DATA=mapssas.canada2;
    ID province;
    CHORO province / DISCRETE NOLEGEND NAME="sasprov";
RUN;
QUIT;
ODS HTML CLOSE;
```



The code to create a province level map using GfK map data is slightly more complicated. The GfK CANADA map contains additional ID variables. With some research, we discovered that the ID1 variable refers to provinces, while the ID variable refers to census districts. In order to produce a map without census district lines, we need to use the GREMOVE procedure, shown below.

```
PROC SORT DATA=mapsgfk.canada OUT=provcanada;
   BY id1;
RUN;

PROC GREMOVE DATA=provcanada OUT=remdiv;
   BY id1;
   ID id;
RUN;
```

Then PROC GMAP code can be run on the new map data set REMDIV, producing the results shown below.

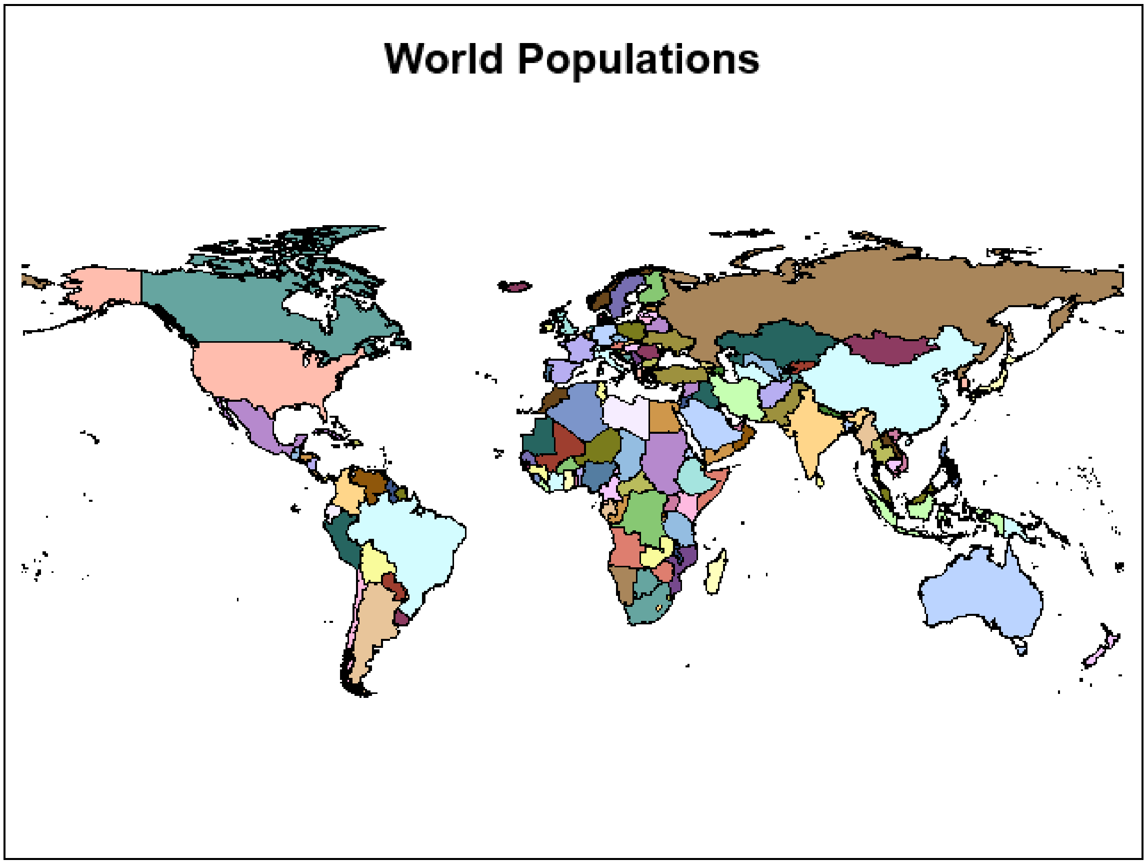
```
GOPTIONS DEVICE=png XPIXELS=1200 YPIXELS=900 BORDER;

ODS HTML PATH=odsout BODY="gfkprov.htm" STYLE=sasweb;
PATTERN1 V=s C=vlig R=5000;
PROC GMAP MAP=remdiv DATA=remdiv ;
   ID id1;
   CHORO id1 / DISCRETE NOLEGEND NAME="gfkprovd";
RUN;
QUIT;
ODS HTML CLOSE;
```

NEW AND COMPLEX MAPS

We are all accustomed to seeing chloropleth maps, with geographic entities colored according to a given metric. All the examples shown so far are chloropleth maps, some with embellishments created by annotation. In SAS/GRAPH, there are other map variations (BLOCK, PRISM and SURFACE), and PROC SGMAP follows suit. Like PROC GMAP, PROC SGMAP has a chloropleth variant, with the CHOROMAP statement. Other current (PROC SGMAP is still pre-production) types of SGMAPs are created using the OPENSTREETMAP and ESRIMAP statements. Both these variations use background maps provisioned by OSM and ESRI. SAS has a very helpful tip sheet on the PROC SGMAP procedure with useful information and code for three samples. Users with SAS 9.4 M5 are encouraged to try PROC SGMAP out and provide feedback to SAS to improve the procedure. The results of running the sample code follow below.

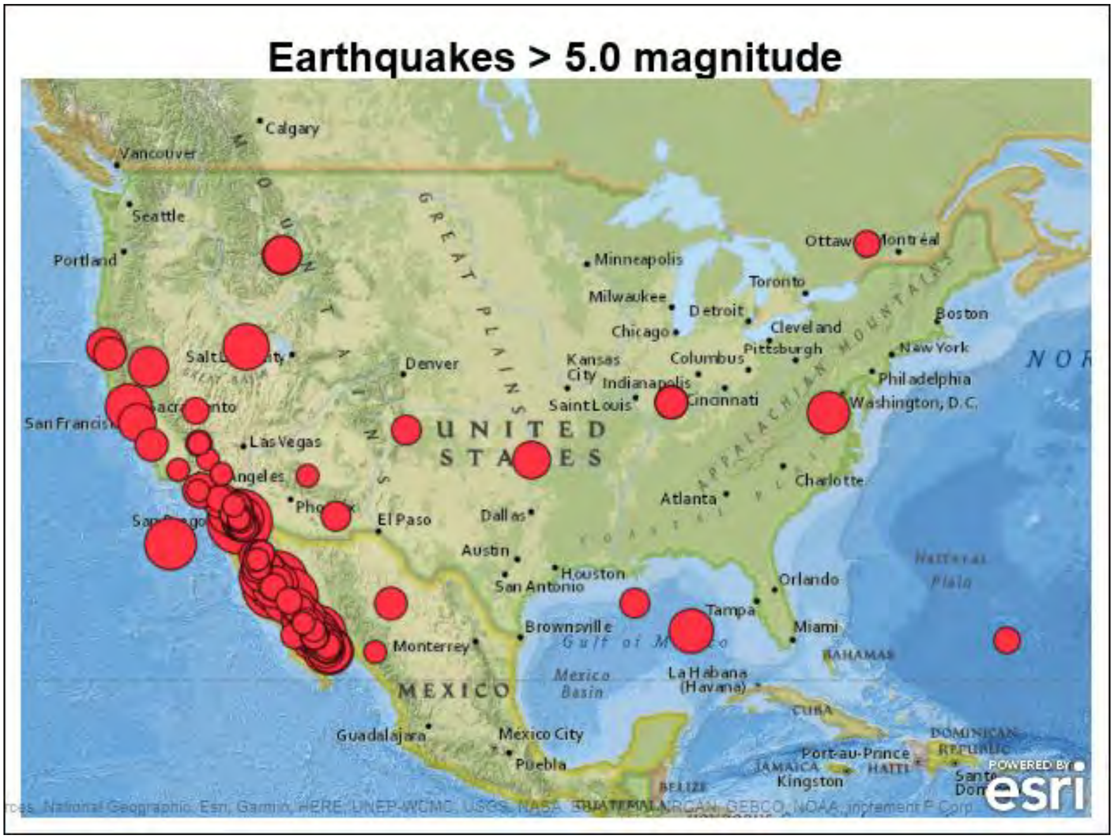
Example 12: PROC SGMAP with the CHOROMAP statement

```
PROC SGMAP MAPDATA=maps.world MAPRESPDATA=sashelp.demographics;
TITLE H=2 'World Populations';
    CHOROMAP popUrban / MAPID=id ID=id;
RUN;
```



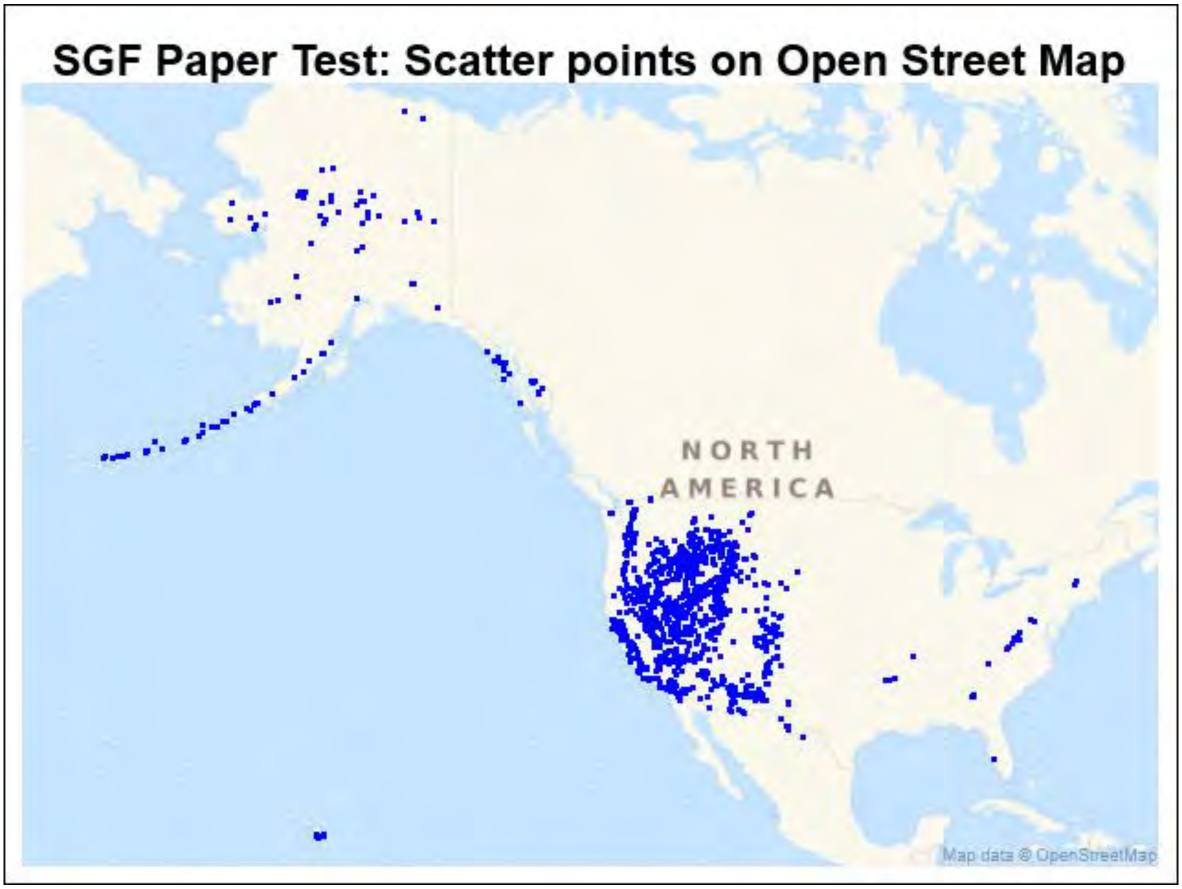
Example 13: PROC SGMAP with the ESRIMAP statement

```
DATA major_quakes;
    SET sashelp.quakes (WHERE=(magnitude>5.0));
RUN;

PROC SGMAP PLOTDATA=major_quakes;
    ESRIMAP URL='http://services.arcgisonline.com/
                 arcgis/rest/services/NatGeo_World_Map';
TITLE H=2 'Earthquakes > 5.0 magnitude';
    BUBBLE X=longitude Y=latitude SIZE=magnitude /
    FILLATTRS=(COLOR=cxff3344);
RUN;
```



Example 14: PROC SGMAP with the OPENSTREETMAP statement

```
PROC SGMAP PLOTDATA=sashelp.springs;
TITLE H=2 'SGF Paper Test: Scatter points on Open Street Map';
    OPENSTREETMAP;
    SCATTER X=longitude Y=latitude / MARKERATTRS=(COLOR=blue SIZE=3
     SYMBOL=circlefilled);
RUN;
```

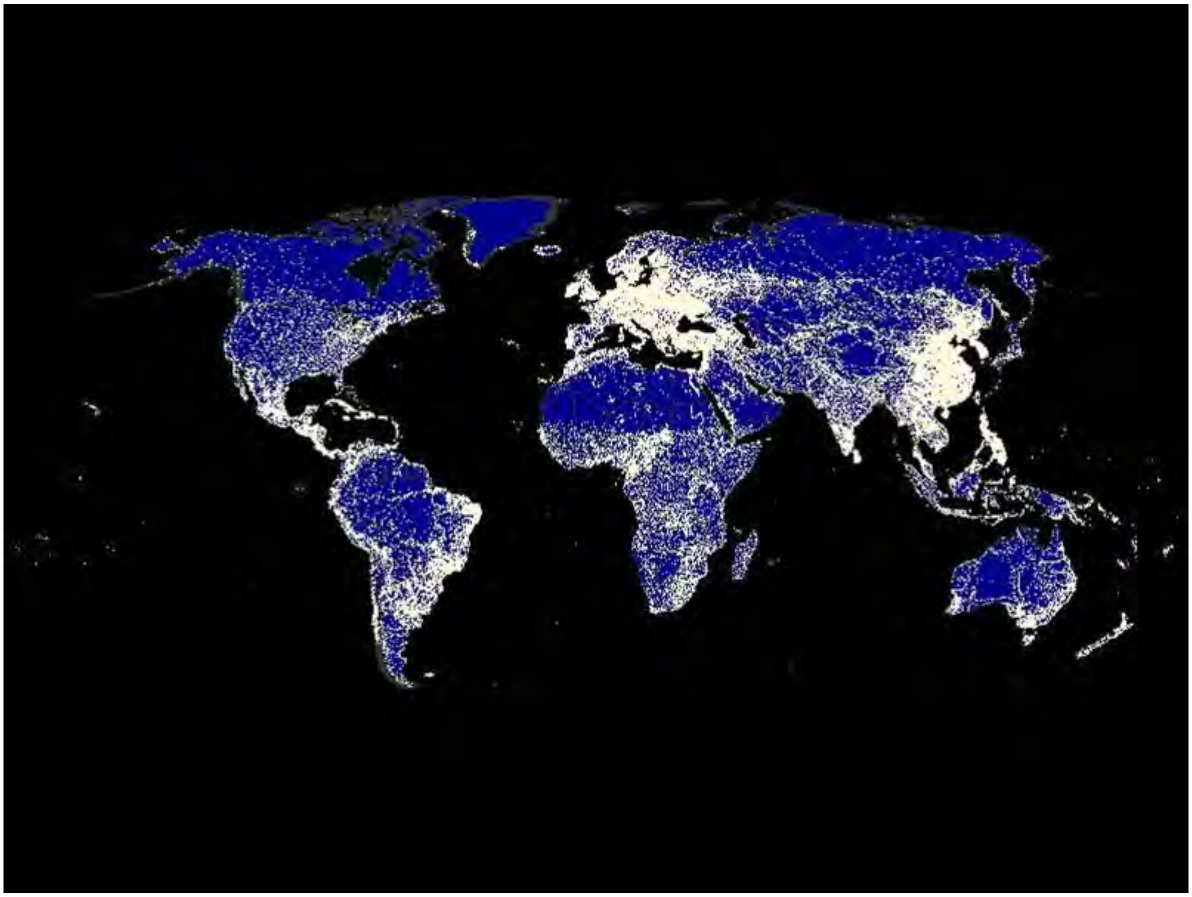


Example 15: The World at Night

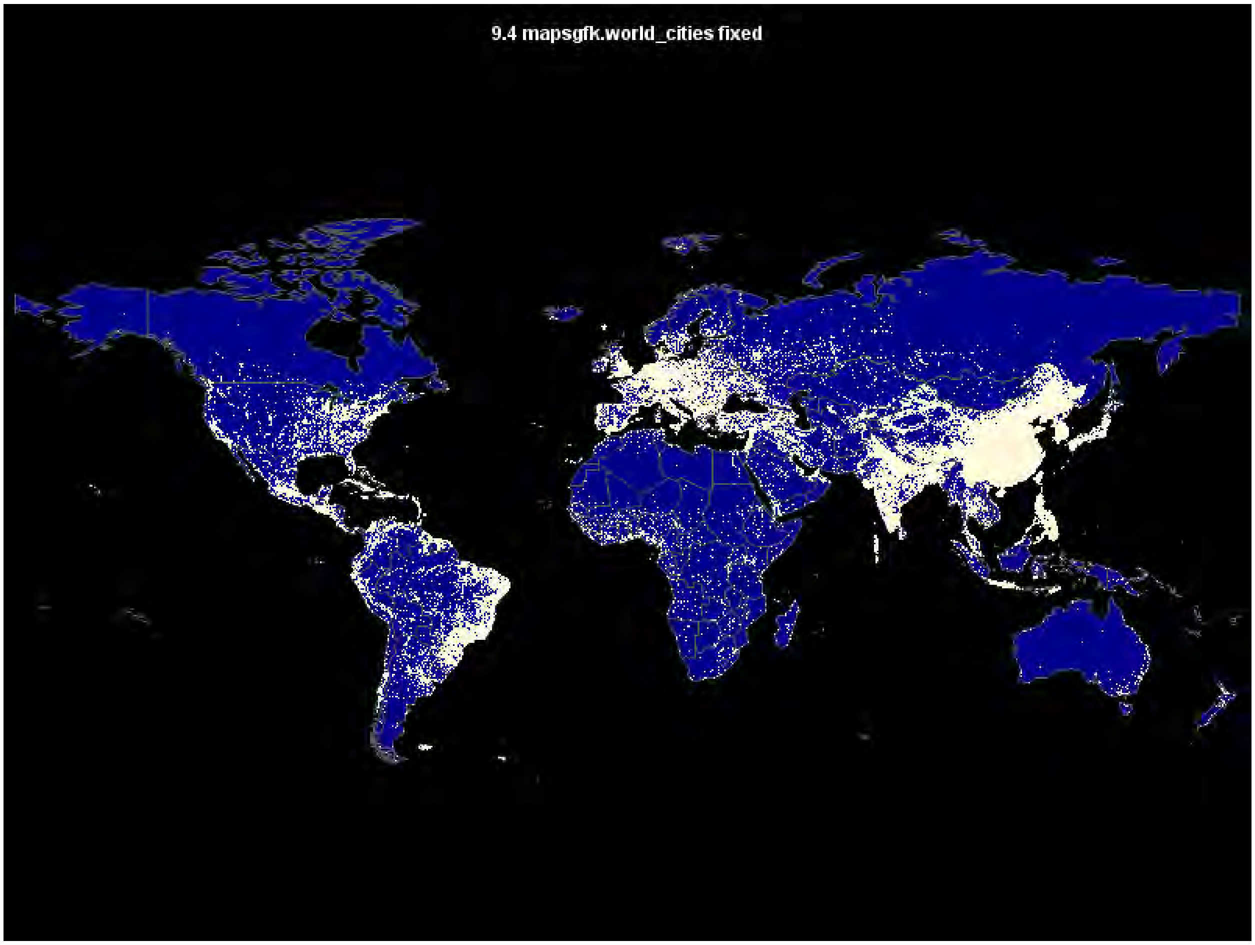
The traditional WORLDCTS and the GfK WORLD_CITIES[_ALL] data sets are highly useful. The examples below are created in SAS/GRAPH by creating an annotate data base with light dots representing world cities (sized according to population in the traditional example) overlaid on a world map (excluding Antarctica.) For the traditional example, the world map is created from the traditional SAS-supplied world map data set, while the latitude, longitude, and population of world cities comes from the traditional SAS-supplied WORLDCTS (world cities) data set. In the GfK example, GfK WORLD_CITIES has now replaced WORLDCTS, and traditional MAPS.WORLD has been replaced by the more up-to-date MAPSGFK.WORLD. This technique offers an excellent way to represent various statistics and is no doubt the precursor of the SCATTER technique shown in the PROC SGMAP example above.

```
PROC SQL;
CREATE TABLE mymap AS
SELECT long AS x, lat AS y, cont, id, segment
FROM maps.world
WHERE (density<=2) and (id ^= 143) and (id ^=405);
CREATE TABLE worldcts AS
SELECT *, (atan(1)/45 * -1*long) as x, (atan(1)/45 * lat) as y
FROM maps.worldcts;
QUIT;
RUN;

DATA dot_anno;
```

```
   LENGTH function style color $ 8 position $ 1;
   RETAIN xsys ysys '2' hsys '3' when 'a';
   SET worldcts;
   anno_flag=1;
   function='pie';
   rotate=360;
   style='psolid';
   color='cornsilk';
   position='5';
   IF (pop > 1000) THEN size=.05;
   ELSE IF (pop > 500) THEN size=.01;
   ELSE size=.0001;
   OUTPUT;
RUN;

DATA combined;
    SET mymap dot_anno;
RUN;

PROC GPROJECT DATA=combined OUT=combined DUPOK PROJECT=robinson;
    ID id;
RUN;

DATA mymap dot_anno;
    SET combined;
    IF anno_flag=1 THEN OUTPUT dot_anno;
    ELSE OUTPUT mymap;
RUN;

GOPTIONS XPIXELS=800 YPIXELS=600;
GOPTIONS GBACK=black;
GOPTIONS DEVICE=jpeg;
ODS LISTING CLOSE;
ODS HTML PATH=odsout BODY="&name..htm" (TITLE="SAS/Graph worldcts (world
cities) locations") NOGTITLE STYLE=minimal;

TITLE1 F="arial/bo" H=2 "Simulation of Earth's Lights at Night";
TITLE2 F="arial" H=2 "Using SAS-Provided World Cities data";

PATTERN1 v=msolid c=darkblue;

PROC GMAP MAP=mymap DATA=mymap ANNO=dot_anno;
    ID id;
    CHORO id / LEVELS=1 COUTLINE=gray33 NOLEGEND DES="" NAME="&name";
RUN;
```

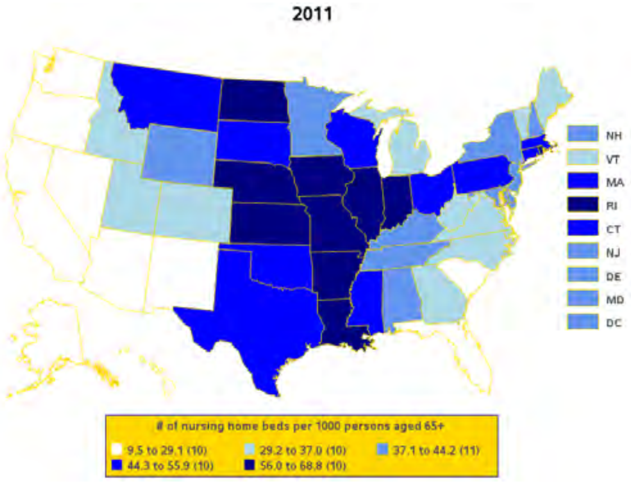


```
DATA my_map1;
    SET mapsgfk.world;
    WHERE (density<=2) AND (id NOT IN('AQ','GL'));
RUN;

DATA dot_anno1 ;
    SETmapsgfk.world_cities;
    WHERE (id NOT IN('AQ','GL'));
    anno_flag=1;
RUN;

DATA combined1;
    SET my_map1 dot_anno1;
 RUN;

PROC GPROJECT DATA=combined1 OUT=combined DUPOK PROJECT=robinson WESTLONG
LATLONG LONGMAX=140 LONGMIN=55 LATMAX=75 LATMIN=0;
    ID id;
RUN;

DATA my_map dot_anno;
SEY combined;
     IF anno_flag=1 THEN OUTPUT dot_anno;
     ELSE OUTPUT my_map;
RUN;
. . .
```



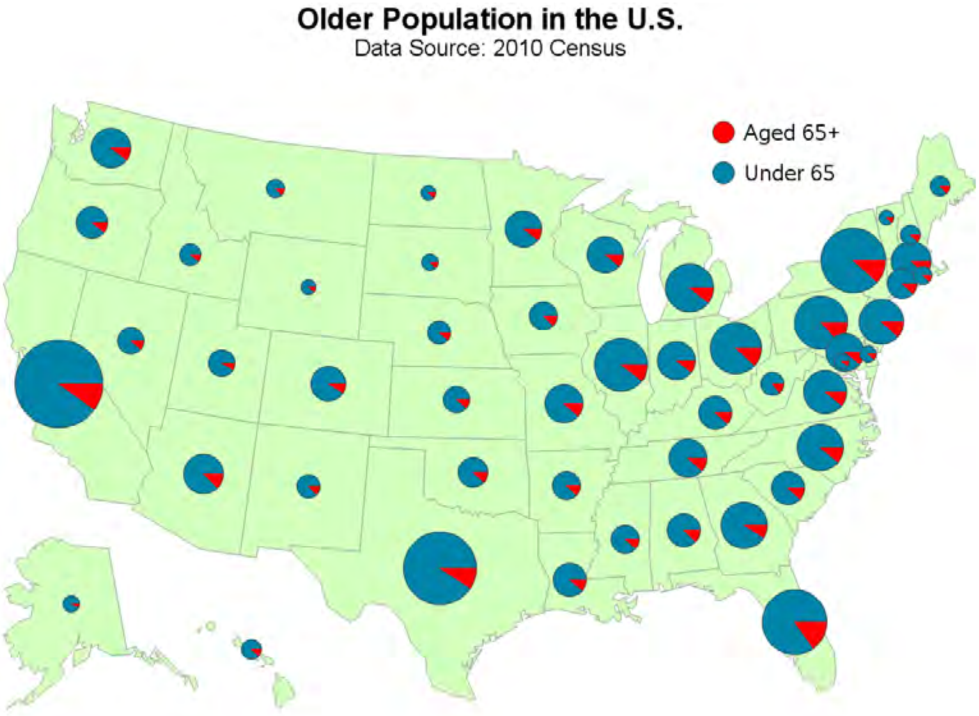
Example 16: Complex Annotated Maps

Below follow several examples of custom maps enhanced with annotate features. The map below (excluding Puerto Rico, Guam and the U.S. Virgin Islands) is one of a series of maps with various measures coded into state quintiles. This annotation (based on one of Robert Allison's amazing examples) features "boxes" for smaller eastern seaboard states color-coded based on the state's quintile for a measure, and a customized legend.



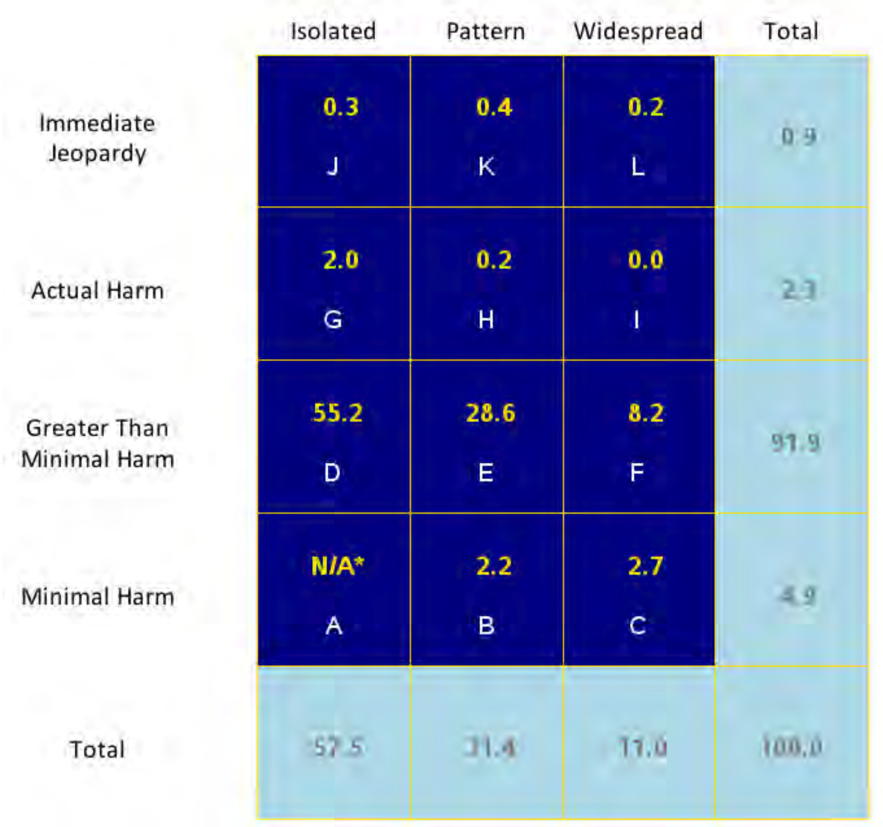
The example below uses the annotate pie function to annotate a tiny pie graph containing a binary statistic on each state.

```
/* Create annotate dataset containing the pie slices */
  data anno_pies; set anno_pies;
  length function color $8 html $500;
      state_name=fipnamel(state);
      non_pct=nonage65p_2010/total_2010;
      age65p_pct=age65p_2010/total_2010;
      xsys='2'; ysys='2'; hsys='3'; when='A';  html='';
      function='PIE'; line=0; style='psolid';
/* Draw the age65+ slice last - otherwise if it's a small slice,
   the larger slice will basically obscure it. */
/* non-65+ pie slice */
      rotate=non_pct*360;
      color="&c_non";
      output;
/* 65+ pie slice */
      rotate=age65p_pct*360;
      color="&c_age65p";
      output;
/* circle around whole pie (with html mouse-over text) */
      html='title='. . .'
      href="aged_population_info.htm"';
      style='pempty'; color='gray44';
      angle=0; rotate=360;
      output;
  run;
```



The example actually draws a “map” of boxes, and annotates statistics, identifiers and labels. It features the use of the %ANNOMAC macro suite provided by SAS, and the %CENTROID macro that is part of %ANNOMAC. Note that PROC SGMAP has a similar suite of annotation macros called %SGANNO.

```
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 2 0 3 3 0 3 3 1 3 2 1
4 3 0 4 4 0 4 4 1 4 3 1

5 0 1 5 1 1 5 1 2 5 0 2
6 1 1 6 2 1 6 2 2 6 1 2
7 2 1 7 3 1 7 3 2 7 2 2
8 3 1 8 4 1 8 4 2 8 3 2

9 0 2 9 1 2 9 1 3 9 0 3
10 1 2 10 2 2 10 2 3 10 1 3
11 2 2 11 3 2 11 3 3 11 2 3
12 3 2 12 4 2 12 4 3 12 3 3

13 0 3 13 1 3 13 1 4 13 0 4
14 1 3 14 2 3 14 2 4 14 1 4
15 2 3 15 3 3 15 3 4 15 2 4
16 3 3 16 4 3 16 4 4 16 3 4

17 0 4 17 1 4 17 1 5 17 0 5
18 1 4 18 2 4 18 2 5 18 1 5
19 2 4 19 3 4 19 3 5 19 2 5
20 3 4 20 4 4 20 4 5 20 3 5
;
run;

data poly_map;
    set poly_map;
    marginal=(id in(1,2,3,4,8,12,16,20));
    if marginal=0 then marginal=2;
run;

%ANNOMAC;

%CENTROID (poly_map, centroid_poly_map, id); . . .

pattern1 color=lightblue;
pattern2 color=navy;
title1 h=48pt;
footnote1;

proc gmap data=poly_map map=poly_map annotate=letters;
id id;
choro marginal / discrete coutline=gold nolegend anno=percents
name="figure2_2_&year.";
run;
```



	Isolated	Pattern	Widespread	Total
Immediate Jeopardy	0.3 J	0.4 K	0.2 L	0.9
Actual Harm	2.0 G	0.2 H	0.0 I	2.3
Greater Than Minimal Harm	55.2 D	28.6 E	8.2 F	91.9
Minimal Harm	N/A* A	2.2 B	2.7 C	4.9
Total	57.5	31.4	11.0	100.0

CONCLUSION

SAS has always been at the forefront of data visualization, from the days of SAS/GIS, ESRI bridge to SAS, SAS/GRAPH, and now with the SG procedures, SAS Visual Analytics, and SAS Viya. I feel privileged to have taken this journey with co-authors and my mentors. SAS has provided programmers with a vast array of tools to make beautiful maps – and those tools continue to evolve. I hope some of the examples in this paper inspire others to explore the unlocked potential of SAS/Graph and PROC SGMAP along with me.

REFERENCES

Code to Project Annotate Data with a GfK Map Data Set ("Chapter 37: GMAP Procedure") in SAS/GRAPH® 9.4: Reference, Fourth Edition.

Hadden, Louise. “What’s in a Map? A Macro-Driven Drill-Down Geo-Graphical Representation System”.

Hadden, Louise, Murphy, Mike, and White, Alan. “From 50,000,000 Claims to One Analytic File”. Proceedings of SUGI 22, April 1997.

Hadden, Louise. “Where in the World Are SAS/GRAPH® Maps? An Exploration of the Old and New SAS® Mapping Capacities.” Proceedings of SAS Global Forum 2014 Conference, April 2014.

http://support.sas.com/rnd/datavisualization/mapsonline/index.html http://www.sascommunity.org/wiki/ZIP_Code_411:_Decoding_SASHELP.ZIPCODE_and_Other_SAS%C2%AE_Maps_Online_Mysteries#Online_Materials

http://www2.sas.com/proceedings/sugi31/143-31.pdf

http://www.gfk-geomarketing.com/en/digital_maps/softwarespecific_maps/maps_for_sas_graph.html

http://support.sas.com/rnd/papers/sasgf13/GfKMaps2013.pdf

http://robslink.com/SAS/Home.htm

https://support.sas.com/documentation/cdl/en/grmapref/69722/HTML/default/titlepage.htm **(HTML)**

https://support.sas.com/documentation/cdl/en/grmapref/69722/PDF/default/grmapref.pdf **(PDF)**

Kelley, Francis J. "Using SAS/Graph® to Generate State Metropolitan Area (MA) Maps. Proceedings of SESUG 1994. October 1994.

Massengill, Darrell. ""Google-Like" Maps in SAS®". Proceedings of SAS Global Forum 2013. April 2013.

Massengill, Darrell. "The New SAS® Map Data Sets." Proceedings of SAS Global Forum 2016 Conference. April 2016.

Okerson, Barbara. "Creating ZIP Code-Level Maps with SAS®." Proceedings of SESUG 2013. October 2013.

Okerson, Barbara. "Distance Mapping in Health and Health Care: SAS® as a Tool for Health Geomatics." Proceedings of SAS Global Forum 2011. April 2011.

Okerson, Barbara. "Mapping Roanoke Island: from 1585 to present." Proceedings of SouthEast SAS Users Group 2016 conference. October 2016.

SAS/GRAPH Concepts section of SAS/GRAPH® 9.4: Reference, Fourth Edition.

Thompson, Stephanie R. "Using Big Data to Visualize People Movement Using SAS® Basics." Proceedings of MWSUG 2016. October 2016.

Using GfK Map Data Sets with Existing Code ("Chapter 37: GMAP Procedure") in SAS/GRAPH® 9.4: Reference, Fourth Edition.

Zdeb, Mike and Allison, Robert. "SAS/GRAPH® 101". Proceedings of SUGI 31. April 2005.

Zdeb, Mike and Allison, Robert. "Stretching the Bounds of SAS/GRAPH® Software". Proceedings of SUGI 30. April 2004.

Zdeb, Mike, Allison, Robert and Hadden, Louise. "Wow! You Did That Map with SAS/GRAPH®?" Proceedings of SAS Global Forum 2009 Conference. April 2009.

Zdeb, Mike. 2002. Maps Made Easy Using SAS. Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

The author wishes to acknowledge Darrell Massengill (retired), Liz Simon and Robert Allison of SAS who work tirelessly to improve and facilitate the use of SAS/GRAPH and mapping with SAS and maintain the SAS MAPS ONLINE site; Ed Odom, Dan Heath, and Sanjay Matange who are ushering in a new era of mapping with base SAS and the SG procedures; Mike Zdeb and Barbara Okerson, extraordinary mapping gurus. Robert Allison of SAS also provides wonderful examples of using SAS/GRAPH and mapping tools, including some of the examples shown in this paper, in his amazing web site.

I'd also like to thank the many readers of my mapping papers over the years for their constant inspiration!

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Louise S. Hadden
Abt Associates Inc.
617-349-2385
Louise_hadden@abtassoc.com
abtassociates.com