MWSUG 2017 - Paper PO02

SAS/GRAPH® and GfK Maps: a Subject Matter Expert Winning Combination

Louise S. Hadden, Abt Associates Inc.

ABSTRACT

SAS® has an amazing arsenal of tools to use and display geographic information that is relatively unknown and underutilized. 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 and "traditional" SAS map data sets with licenses for SAS/GRAPH for some time, recognizing there will need to be an extended transitional period. However, for those of us who have been putting off converting our SAS/GRAPH mapping programs to use the new GfK maps, the time has come, as the "traditional" SAS map data sets are no longer being updated.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. The GfK maps are updated once a year. This presentation will walk through the conversion of a long-standing SAS program to produce multiple US maps for a data compendium to take advantage of GfK maps, as well as several other examples. Products used are Base SAS® and SAS/GRAPH®. SAS programmers of any skill level will benefit from this presentation.

INTRODUCTION

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 on a yearly basis. 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 (more detailed information can be found in the excellent papers GfKMaps2013 by Liz Simon and Darrell Massengill (http://support.sas.com/rnd/papers/sasgf13/GfKMaps2013.pdf) and

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

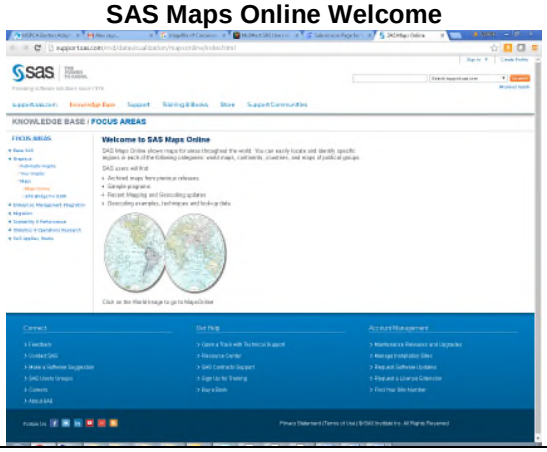
- Many changes to variable names
- Added RESOLUTION variable
- ID variables are now character and unique world wide
- X and Y are ALWAYS projected

- Lat and Long are ALWAYS unprojected, based on the eastern Hemisphere and in degrees (PROC GPROJECT will require the DEGREES and EASTLONG options)

SAS MAPS ONLINE

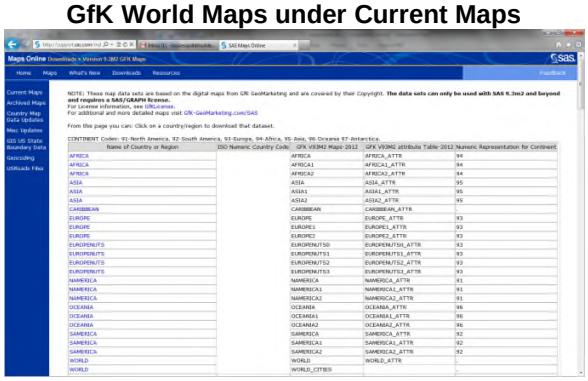
SAS MAPS ONLINE is truly a buried treasure. It is a SAS-provided web page with a number of useful tools regarding mapping, including both traditional and the GfK map data sets and updates, related data sets (such as SASHELP.ZIPCODE, SASHELP.MAPFMTS, the new GfK WORLD_CITIES_ALL, and WORLDCTS), and sample 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.*** Screenshots of the major web pages related to GFK maps under SAS MAPS ONLINE that may help with navigating the site are presented below. 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.) The next planned GfK update is scheduled for the SAS 9.4M5 release. ***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.***

SAS Maps Online Welcome

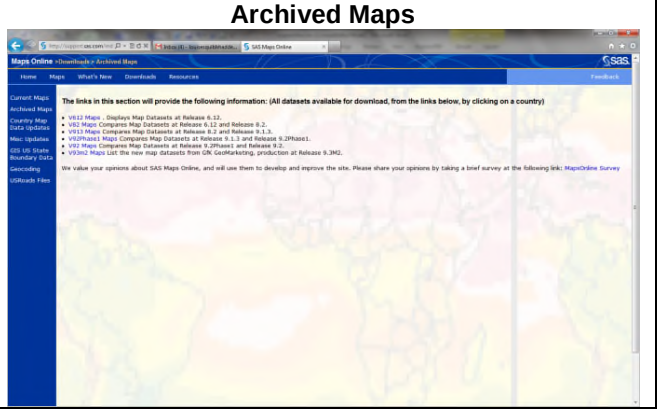


SAS Maps Online Home

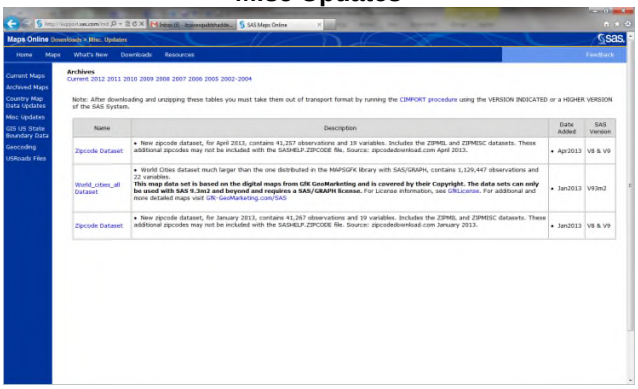
GfK World Maps under Current Maps



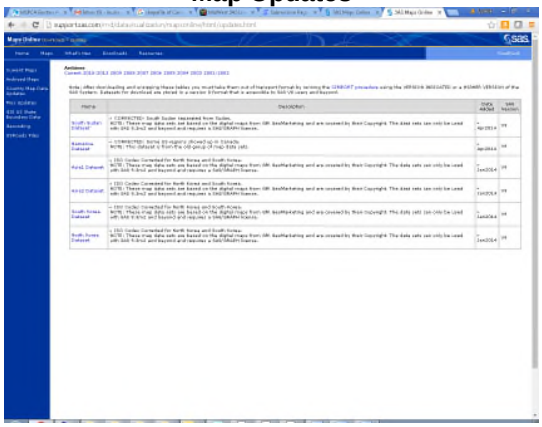
Archived Maps



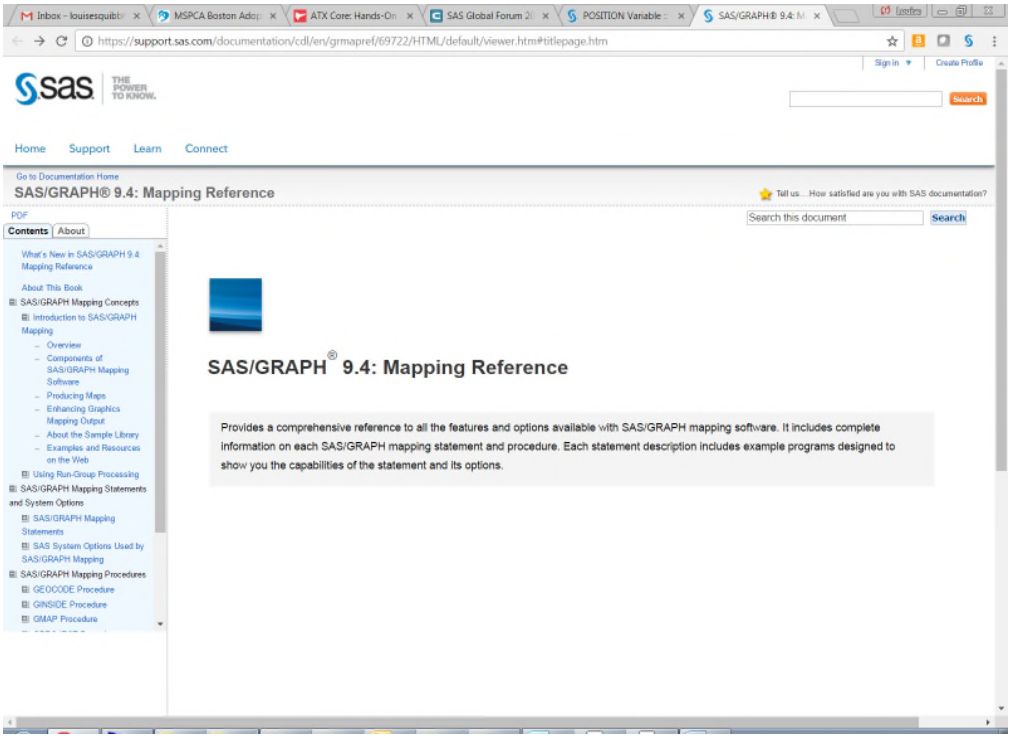
Misc Updates



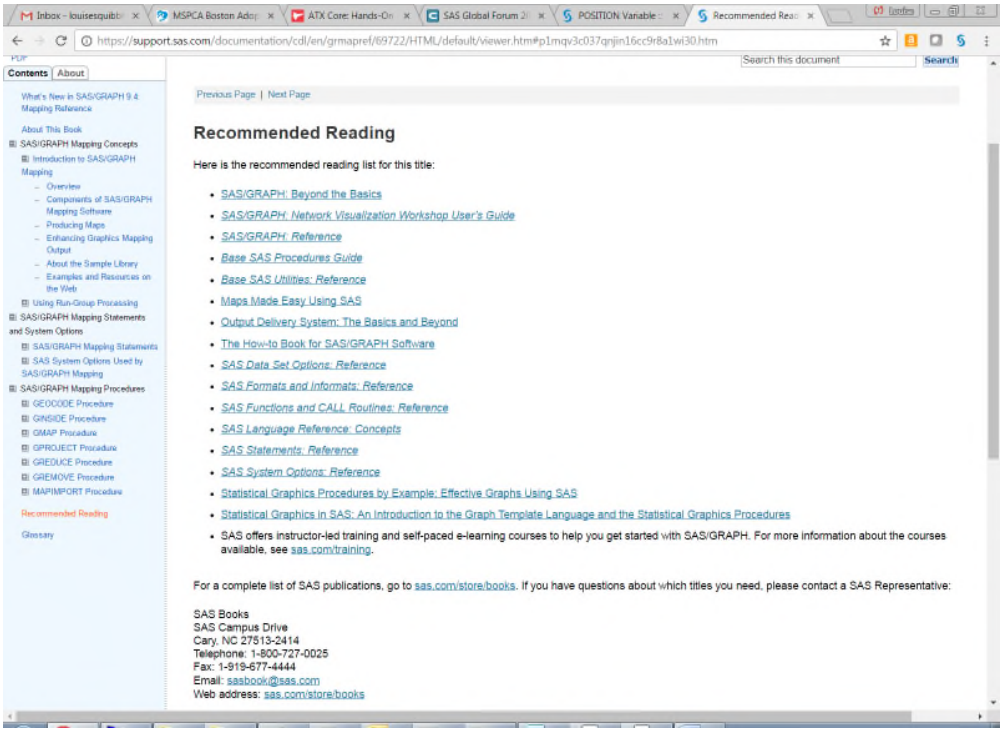
Map Updates



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)



This 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.

EXAMPLE 1: US MAP WITH ANNOTATED BOXES

We need to produce the figures and associated legends for several dozen maps for several dozen different continuous measures and time periods for an annual data compendium … so, the process is automated. A method was devised using SAS® PROC RANK to generate the quintiles, PROC SQL to get the data value ranges within each quintile, and PROC FORMAT (with the CNTLIN= option) to generate and store the legend labels. The resulting data files and format catalogs are then used to generate both the maps (with legends) and associated "alt text" (for Section 508 compliance). Then, these processes were rolled into a macro to apply the method for the many different maps and their legends.

First, a “base” map data set which includes separate boxes for small states in the Northeast is created. All figures we generate are drawn using this map. The code for this map is loosely based on one of Robert Allison's wonderful examples. Note that the map used is from the SAS MAPS library.

```
DATA mymap;
    SET maps.us (where=(state ne 72));
RUN;

DATA extra_squares;
    INPUT statecode $ 1-2;
    state=stfips(statecode);
    DATALINES;
    NH
    VT
    MA
    RI
    CT
    NJ
    DE
    MD
    DC
    ;
RUN;

DATA extra_squares;
    SET extra_squares;
    RETAIN y;
    segment=999;
    IF _n_=1 THEN y=.15;
    ELSE y=y-.010;
    x=.345; OUTPUT;
    x=x+.040; OUTPUT;
    y=y-.020; OUTPUT;
    x=x-.040; OUTPUT;
RUN;

DATA mymap;
    SET mymap extra_squares;
    state_name=FIPNAMEL(state);
RUN;

PROC SQL;
    CREATE table extra_label AS
    SELECT UNIQUE avg(y) AS y, avg(x)+.03 AS x, statecode AS text
    FROM extra_squares
```

```
    GROUP BY statecode;
QUIT;
RUN;

DATA extra_label;
    SET extra_label ;
    xsys='2'; ysys='2'; hsys='3'; when='a';
    function='label'; color='gray55'; position='6'; size=2.25;
RUN;
```

The map (with legend) is generated, with a file written to HTML as well as generating a PNG file. OPTIONS are set to output to a landscape page and suppress the printing of the date and page number; GOPTIONSs are set to determine the type of image (PNG), suppress a border around the graphic, determine the units used in measurements of the image, and assign text formatting; ODS listing is closed to suppress “windowing” of the graphic while running in batch, and ODS destination(s) (HTML) are selected. The result is in Figure 1.

```
. . .
goptions device=png noborder;
ODS HTML path=odsout body="&invar._&year..htm" options(pagebreak='no');
ods escapechar='^';
goptions cback=white;
goptions gunit=pct ftitle="albany amt/bold" ftext="albany amt/bold";
goptions htitle=18pt htext=11pt ctitle=black ctext=gray55;

legend1 label=("&leglab." height=8pt position=top justify=center) across=5
 cframe=gold cborder=navy
 position=(bottom center outside) shape=bar(.15in,.15in) mode=reserve
 offset=(0,0)
value=(j=l "&t1" "&t2" "&t3" "&t4" "&t5");

pattern1 value=msolid color=white;
pattern2 value=msolid color=lightblue ;
pattern3 value=msolid color=CornflowerBlue ;
pattern4 value=msolid color=blue ;
pattern5 value=msolid color=navy;

title1 a=-90 h=5pct " ";
title2 a=90 h=1pct " ";
title3 h=18pt bold "&year";


proc gmap data=my_data map=mymap all anno=extra_label gout=library.excat;;
id state_name;
choro msr_bin / discrete coutline=gold woutline=1
cdefault=yellow
legend=legend1
html=my_html
des='' name="&y2.";
run;

quit;
```

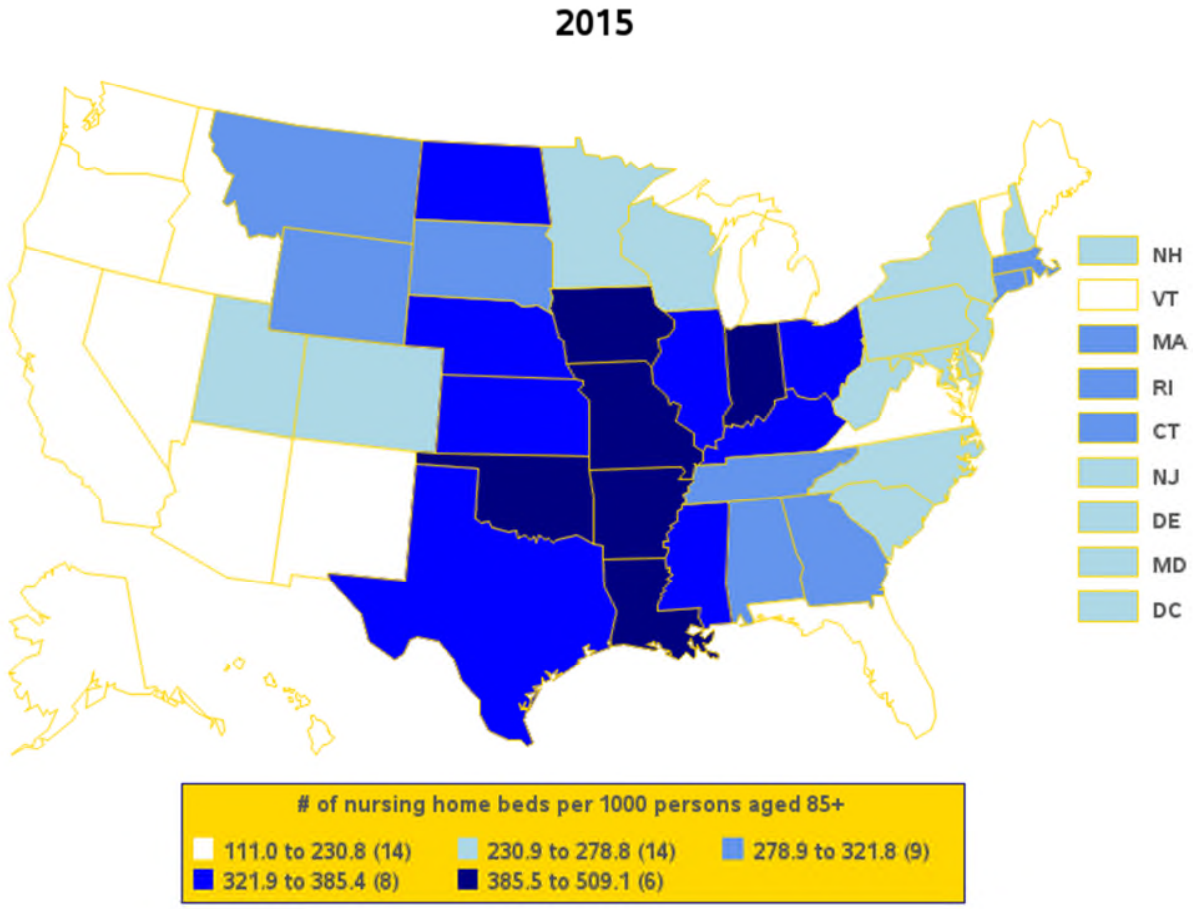


Figure 1: US Map using traditional map data sets

The only change that needs to be made to create this complex set of maps using GfK maps is to change the source library for the US map from MAPS to MAPSGFK.

```
DATA mymap;
    SET mapsgfk.us (where=(state ne 72));
RUN; . . .
```

The result of running the identical code (except for the libname reference) is shown below in Figure 2.

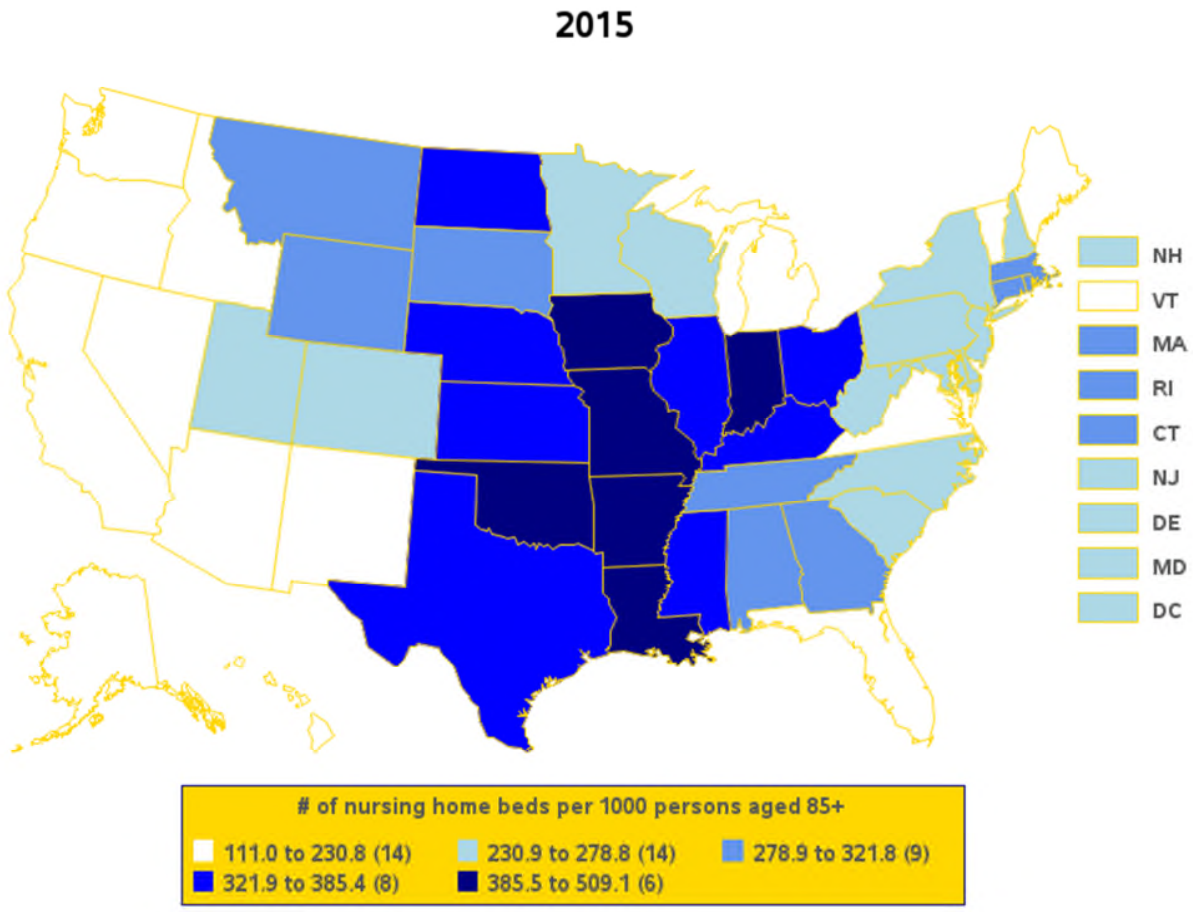


Figure 2: US map using MAPSGFK as map source library

We won't always be that lucky. As noted above, the ID variables for GfK maps are different, AND the traditional map data sets have not been updated for a long time, causing differences in some maps (for example, US county maps). Annotation also becomes an issue, as all GfK maps are projected so you may need to retrieve and reproject the unprojected coordinates.

EXAMPLE 2: OTTER COUNTRY

In this example inspired by Barbara Okerson, we work with a subset of counties in California. We'll reverse directions here and 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;
```

As for Figures 1 and 2, 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 in Figure 3.

```
/*County Map GMAP code*/

FILENAME odsout '.';
ODS LISTING CLOSE;

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;

ODS LISTING;
```

Figure 3: Map of selected counties using MAPSGFK as map source library

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 Figure 3. 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;
```

Figure 4: Map of selected counties using traditional SAS maps (MAPSSAS) as map source library

EXAMPLE 3: OH, CANADA

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. The code to create such a province level map with the traditional map data set CANADA2 is presented below, and the results shown in Figure 6.

```
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;
```

Figure 6: Map of Provinces in Canada using traditional SAS map data set CANADA2

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 in Figure 7 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;
```

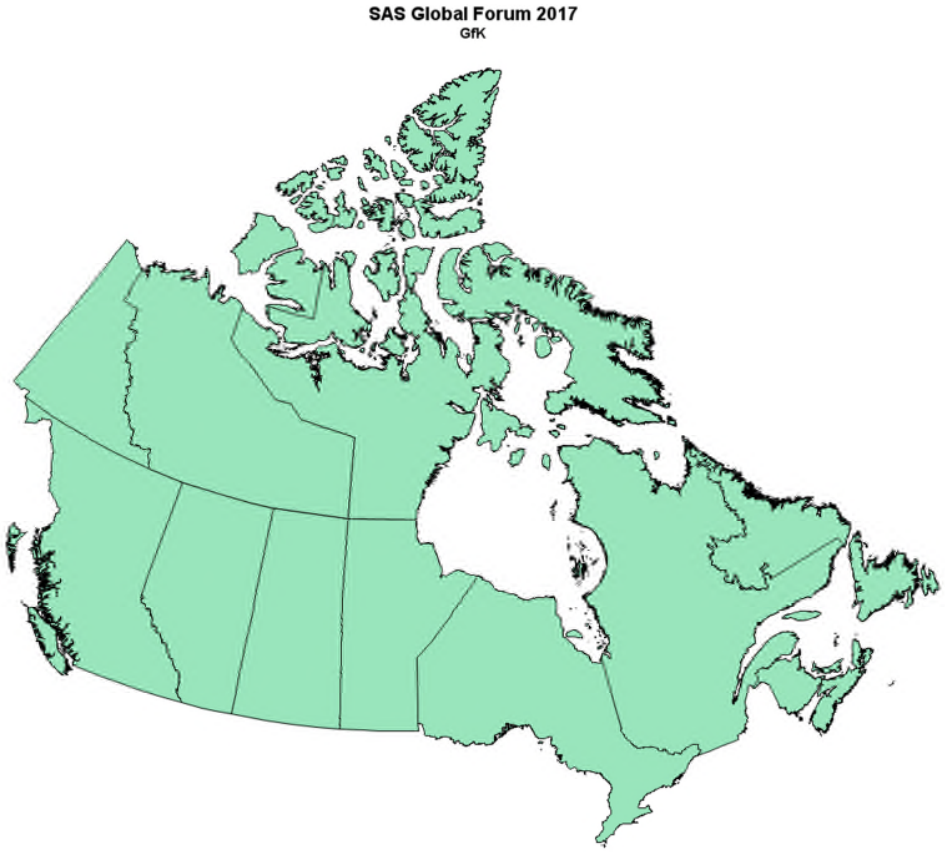


Figure 7: Map of Provinces in Canada using PROC GREMOVE and GfK CANADA map data set

Annotating province names in both the GfK and traditional SAS map data sets is a further step, most easily accomplished by using the SAS supplied %CENTROID macro (obtained by running %ANNOMAC in your SAS session), then building an annotate data set to use while mapping.

EXAMPLE 4: THE WORLD AT NIGHT

In this example (also based on an example from Robert Allison), we create an annotate data base with light dots representing world cities sized according to population overlaid on a world map (excluding Antarctica.) In the original code run in 2009, the world map is created from the traditional SAS-supplied world map data set WORLD, while the latitude, longitude, and population of world cities comes from the

traditional SAS-supplied WORLDCTS (world cities) data set. The code is presented below and the results shown in Figure 8.

```
. . .

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;
```

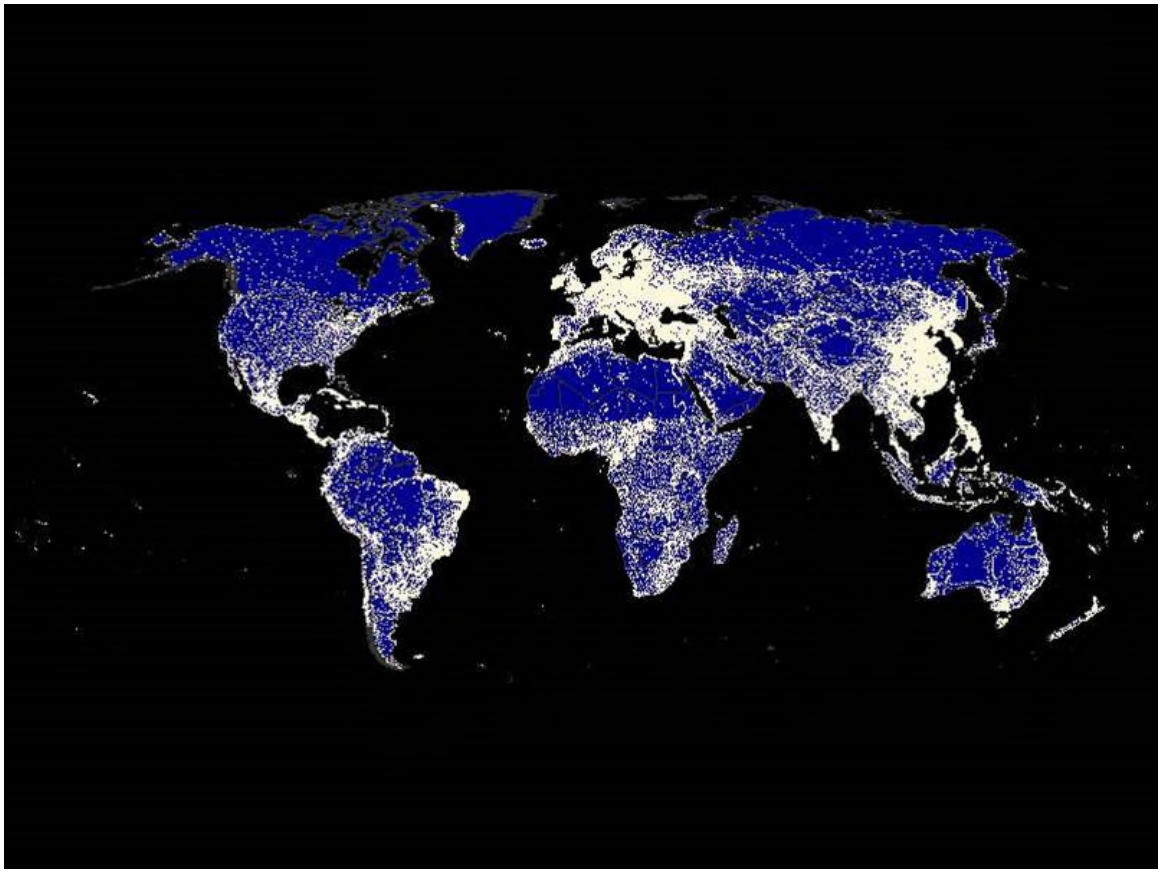


Figure 8: World at night using MAPS.WORLD and MAPS.WORLDCT

We attempted to update this map using MAPSGFK.WORLD and MAPSGFK.WORLD_CITIES. WORLDSCT is no longer available from SAS – it was extremely out of date and incomplete. WORLD_CITIES, which shipped / ships with versions of SAS beyond 9.3 M2, is ALSO not complete – you should download WORLD_CITIES_ALL, which is updated annually, from SAS MAPSONLINE (requires a SAS/GRAPH license.) For the purpose of this demonstration, we'll use WORLD CITIES. WORLD_CITIES has data on around 200,000 cities worldwide – WORLD_CITIES_ALL has data on over 1,000,000 cities. Both MAPSGFK.WORLD and MAPSGFK.WORLD_CITIES have a different ID variable than MAPS.WORLD and MAPS.WORLDCT. Below follows the image created by the revised code (just replacing the library names) in Figure 9.

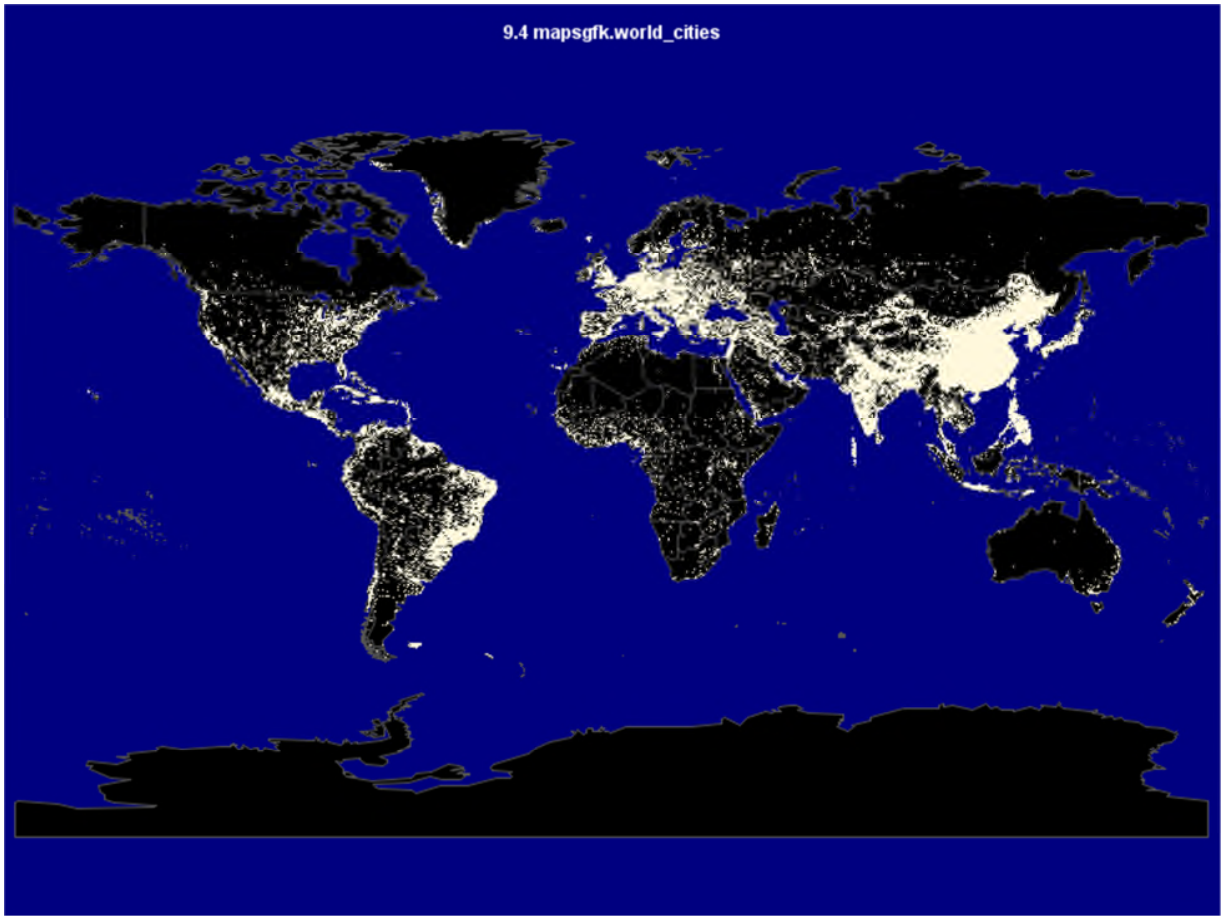


Figure 9: World at night using MAPSGFK.WORLD and MAPSGFK.WORLD_CITIES

What happened here? First of all, Antarctica and Greenland not excluded using the ID variable as a screen.. In 2009, in MAPS.WORLDCT the ID variable was numeric so our where clause did not work. Secondly, MAPS.WORLDCT contained actual population numbers while MAPSGFK.WORLD_CITIES contains a variable with conflicting population levels. In general there is not the curved look evident in the first map. How do we fix this map (and in doing so, give ourselves a blueprint for retrofitting our PROC GMAP code)?

First, check your old and new data sets with PROC CONTENTS and a PROC PRINT test, following up with a PROC FREQ cross-tabulation with relevant variables using the LIST MISSING OPTION. In this case, it was not possible to check the old WORLDCT data set because it is no longer provided, but we have the code we used in the past to provide clues. Below follow snippets of the tests in the SAS list.

Contents listing of MAPSGFK.WORLD

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat	Label
12	CONT	Num	5			Numeric number for continent
9	DENSITY	Num	5			Greduce density values
1	ID	Char	15			Alpha2 Country code
3	IDNAME	Char	55			Country name
8	ISO	Char	3	$3.	$3.	ISO Country code
13	ISOALPHA2	Char	2	$2.	$2.	ISO Alpha2-code for country
11	LAKE	Num	5			Lake Flag:1-water:2-citytype
5	LAT	Num	8			Unprojected degrees latitude
4	LONG	Num	8			Unprojected degrees longitude (East)
10	RESOLUTION	Num	5			Map detail level based on . . .

```
    2    SEGMENT       Num       5                        ID segment number
    6    X             Num       8                        Projected Longitude: Cylindrical
    7    Y             Num       8                        Projected Latitude: Cylindrical


    Sort Information
    Sortedby       ID
```

Contents listing of MAPSGFK.WORLD_CITIES

```
                          Alphabetic List of Variables and Attributes


    #    Variable      Type    Len    Label

    8    CITY          Char     55    World Cities
    5    CITY2         Char     55    Clean CITY name for geocoding
    6    CONT          Num       6    Numeric number for continent
    9    CtType        Char     25    POP categories for cities where appl.
    3    ID            Char      2    Country Alpha code
   20    IDNAME        Char     65    Country name
   14    ISO           Char      3    ISO Country code
   15    ISOALPHA2     Char      2    ISO Alpha2-code for country
   16    ISOALPHA3     Char      3    ISO alpha3 country code
    7    ISONAME       Char     55    ISO Country name
   22    ISONAME2      Char     55    Clean ISO Country Name for Geocoding
   13    LAT           Num       8    Unprojected degrees latitude
   12    LONG          Num       8    Unprojected degrees longitude
   17    MapID         Char     15    ID value from MAP dataset
   18    MapIDName     Char     65    IDNAME from MAP dataset
   21    MapIDName1    Char     55    ID1NAME from MAP dataset
    4    MapIDName2    Char     65    Clean state/province name for geocoding
   19    MapLevel      Char     25    MAP Level
   10    Rank          Num       6    Grouping of CtType high-low
```

Test print of MAPSGFK.WORLD

```
ID     SEGMENT     IDNAME        LONG        LAT          X           Y

AD        1        Andorra     1.72563     42.5052    -17904.61    5442.34    . . .
```

Test print of MAPSGFK.WORLD_CITIES

```
      X          Y      ID   MapIDName2     CITY2            CONT   ISONAME

 -17910.84   5442.56   AD   ANDORRA        ANDORRALAVELLA    93    Andorra

   LONG      LAT     ISO   ISOALPHA2   ISOALPHA3   MapID   MapIDName       MapLevel

  1.5451   42.5107   020      AD          AND      AD-AD   Andorra         provinces
```

We opt to cross-tabulate id and idname to identify what id values to use to drop Antartica and Greenland.

Test cross-tabulation on MAPSGFK.WORLD

```
ID       IDNAME                                              Frequency     Percent
-------------------------------------------------------------------------------
AD       Andorra                                                23        0.01
AE       United Arab Emirates                                  485        0.20
AF       Afghanistan                                           618        0.25
AG       Antigua and Barbuda                                    70        0.03
AI       Anguilla                                               35        0.01
AL       Albania                                               207        0.08
AM       Armenia                                               245        0.10
AO       Angola                                                545        0.22
AQ       Antarctica                                           6145        2.49
 . . .
DZ       Algeria                                               483        0.20
```

```
GB       United Kingdom                                3313        1.34
GD       Grenada                                         37        0.02
GE       Georgia                                        327        0.13
GF       French Guiana                                  152        0.06
GG       Guernsey                                        38        0.02
GH       Ghana                                          264        0.11
GI       Gibraltar                                        4        0.00
GL       Greenland                                     9721        3.95
. . .
```

We now know which ID values to eliminate from our map, AND we know that ID is character, not numeric as it was in 2009. Another change to be made is to simply represent each city with the same sized dot, as the population data in MAPSGFK.WORLD_CITIES is available in overlapping population categories. Furthermore, we know that LAT and LONG are unprojected coordinates, while X and Y are projected coordinates. We edit the program to reflect the desired changes, and rerun. Snippets of the revised code and the revised map (Figure 10) are shown below.

```
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;
. . .
```

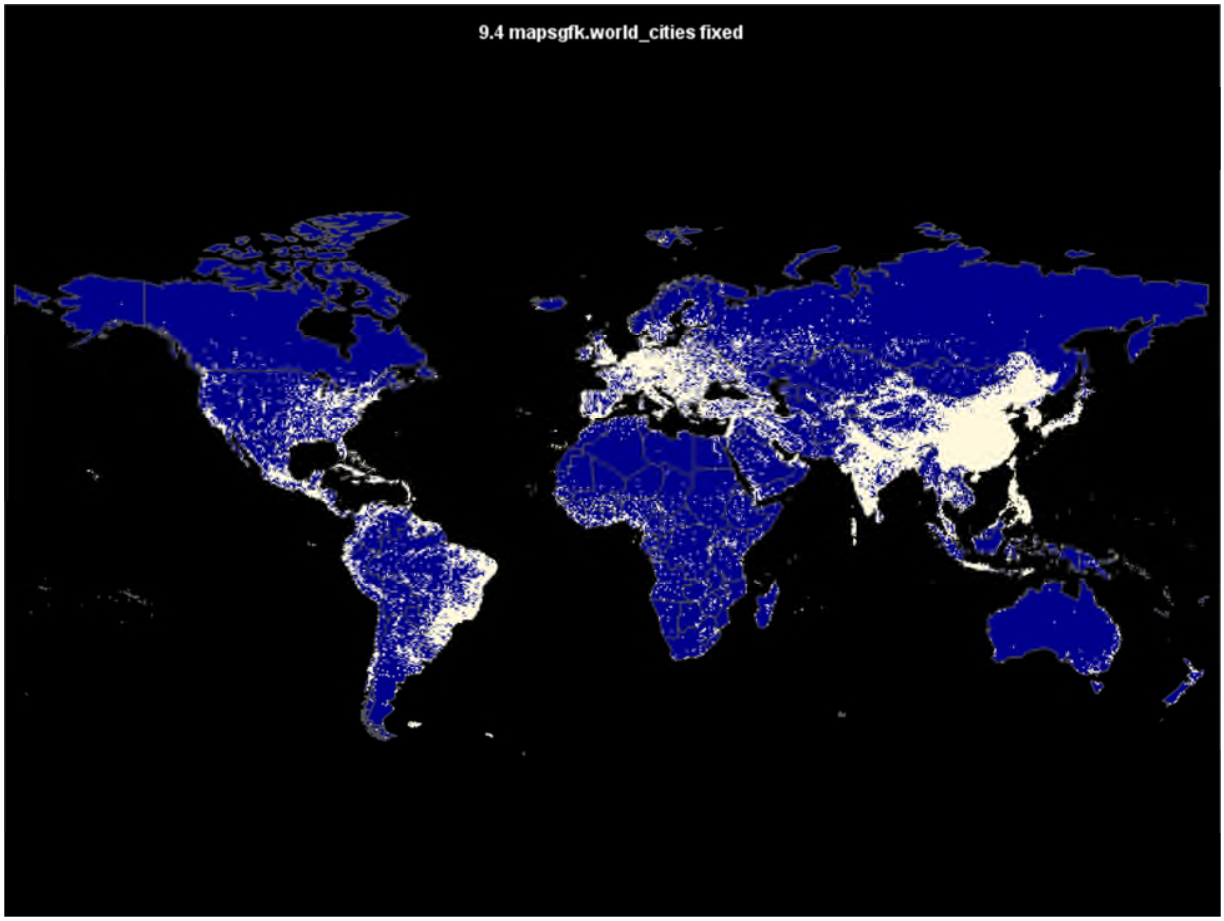


Figure 10: Corrected World at Night using MAPSGFK.WORLD and MAPSGFK.WORLD_CITIES

CODE RETROFITTING CHECKLIST

We have seen one example of retrofitting SAS code to accommodate the new GfK maps that is very easy to accomplish, one example of creating similar images that is easier to accomplish with the new GfK maps than with traditional SAS map data sets, one example of creating similar images that is harder to accomplish with the new GfK maps than with traditional SAS map data sets, and one relatively straightforward example that demonstrates many of the techniques needed to update SAS code for the GfK maps. Note that many of these tasks should be done when using traditional map data sets as well – but the changes in the MAPSGFK data sets are a good excuse to institute these practices as a general rule.

BEFORE YOU START

Make sure you know all the details about the traditional and MAPSGFK data sets, including map, annotate and response data sets. Ensure that you have downloaded the most recent relevant MAPSGFK data sets from SAS MAPSONLINE before starting out. Make sure you know all the differences betweeb traditional and MAPSGFK data sets and understand the implications of those differences. PROC COMPARE is a good tool, as are PROC CONTENTS, PROC PRINT and PROC FREQ. In example 2, the population variable in the traditional annotate data set was a number indicating the actual population, while the same variable in the MAPSGFK annotate data set was a overlapping population category. This would not have been obvious without using PROC FREQ.

Ensure that your annotate and response data sets match up with your map data set with regard to the ID variable. MAPSGFK data sets have a very different id structure and you may need to use multiple ID variables, and you may need to create these variables on your response data set.

Be aware that your map, annotate and response data sets may need to be reprojected together in order to work. Example 2 demonstrates the technique. The contents of MAPSGFK.WORLD show that the data set has both projected and unprojected coordinates, but the contents of MAPSGFK.WORLD_CITIES show that there are only unprojected coordinates available. Therefore, the combined data set must be reprojected in order to match the map and annotate data sets together for the final product. MAPSGFK map data sets have both projected and unprojected coordinates, but this isn't necessary true of data sets commonly used for annotation. The status of traditional map data set regarding projection is not always consistent, and you may need to carefully look at programs that use annotate and/or GPROJECT.

IN PROCESS

Check your interim results carefully. In example 2, some country exclusions that occurred in the traditional map version did not happen in the MAPSGFK version due to changes in the ID variable. Variables with the same name can change in both content and variable type dimensions in between the traditional and MAPSGFK version of a data set. Once the evidence is clearly written on the map, return to your input file review and explore what might be causing the discrepancy and correct it.

REVIEWING THE FINAL PRODUCT

Look at your two maps side by side. Even if you correct any issues detected in earlier reviews, the maps might not look exactly the same. For example, MAPSGFK map files offer the possibility of more control over the map resolution, and the default projection used isn't exactly the same in all cases. You will need to evaluate if the differences matter to the final product. In example 4, we treated the population variable differently, but differences in the final product are pretty transparent to the end user. Bear in mind, too, that the underlying data may have changed. Population increases and decreases over time in given areas, new counties and cities emerge, etc.

CONCLUSION

MAPSGFK map and annotate data sets deliver more up to date and accurate data for SAS mapping than the traditional map data sets. However, using these new data sets with existing code is not always "plug and play". The examples shown will provide helpful suggestions for SAS users embarking on the task of retrofitting existing SAS/GRAPH code. In some cases (for example US county maps or maps for volatile regions in the world) it is vital that SAS users move toward using more up-to-date maps. In other cases, such as mapping US states, it is less urgent; however, the traditional map data sets are being phased out, and it has been demonstrated that code using the US map data set does not require extensive changes. Users who are nervous about losing the traditional map data sets once they stop shipping with new releases of SAS should save the folder outside of their SAS installation.

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. "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)**

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

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.

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

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 and Liz Simon 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; Robert Allison of SAS who provides wonderful examples of using SAS/GRAPH and mapping tools, including some of the examples shown in this paper; and Mike Zdeb, extraordinary mapping guru. Thanks also to Barbara Okerson and Troy Martin Hughes for inspiring the Monterey examples.

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