

SAS3421-2016
The New SAS® Map Data Sets
Darrell Massengill, SAS Institute Inc., Cary, NC

ABSTRACT

SAS customers have a growing need for quality and accurate SAS maps. In order for SAS to provide these quality maps, we need good, solid, and accurate map data. Because SAS is not in the business of generating map data, we have licensed the maps from a third party to satisfy this need.

This paper explores the new maps by discussing the problems and limitations with the old map data, along with showing features and examples of using the new map data.

INTRODUCTION

In the past, we used various sources of map data - some free, some inexpensive, and others more costly. We spent huge amounts of time processing this data to fit our needs. Besides the time involved, this introduced a lot of problems into the map data. And there were a lot of inconsistencies in the data introduced because the data came from different sources. This approach will no longer meet our needs nor the needs of our customers. The only way to provide up-to-date, accurate, and good quality maps is to have a single standardized source of map data. As a result, we have partnered with a third-party vendor to provide annual updates.

The new maps are not identical to the old maps because of the need to make the new data more consistent and easier to use. In this paper we highlight the major differences between the old and new map data, such as new library names, new filenames, and new variables. We include some examples to show coding changes required to create a new map data program. Some examples also highlight features of the new data. We show how to use some of the new PROC GMAP functions and features with this data.

OLD MAP DATA SETS

The old map data sets (in the MAPS and MAPSAS libraries) were based on free or purchased maps from various sources. Integrating these sources introduced inaccuracies, created problems, and cost us a huge amount of processing time. This approach no longer met our needs because

- inexpensive sources of data have basically dried up. For example, we can no longer get the CIA world data for processing our world maps, and thus, we cannot create any more maps.
- other sources of data had additional problems. For example, the data came with inconsistent coordinates and was frequently out of date. In addition, there was no technical support for the data.

The old maps cannot and will not be updated.

NEW GfK MAP DATA SETS

In order to keep up with the new uses and demands for maps from SAS and our customers, we needed more accurate map data (for creating SAS map data sets), so we partnered with a third-party vendor, GfK GeoMarketing (<http://www.GfK-GeoMarketing.com/sas>).

Before creating the new SAS maps, we set the following goals:

1. Make the maps more consistent and easier to use.
2. Avoid any unnecessary processing of the data, thus allowing for more frequent updates.
3. Avoid breaking compatibility with the old SAS maps, except where absolutely necessary to fix goals 1 and 2.

It was extremely difficult to avoid breaking compatibility with the old SAS maps while maintaining the first and second goals. Therefore, changes were necessary. The following is a summary of the changes in the new maps:

1. **New License** The new maps are licensed from GfK GeoMarketing. The following note will appear in the SAS Log:
NOTE: "The map data sets in library MAPSGFK are based on the digital maps from GfK GeoMarketing and are covered by their Copyright. For additional information, see <http://support.sas.com/maponline/gfkllicense>".
2. **New Libnames** The new maps will be shipped with the libname MAPSGFK. The old maps will be shipped with the libname MAPSSAS. MAPS will point to MAPSSAS in SAS® 9.3 and SAS® 9.4, but it will change in a future release. MAPS can be assigned to point to either libname. Here is an example: `libname MAPS (MAPSGFK) ;`

Note that the old and the new maps will be shipped during a migration period, but the old maps will not be shipped in a future version. (The old maps can be downloaded from the MapsOnline web site here:

<http://support.sas.com/rnd/datavisualization/maponline/html/archivedmaps.html>).

3. **New Files and Filenames**
 - Filenames have changed to make the names more consistent, longer, updated, and more descriptive. For example, AFGHANIS becomes AFGHANISTAN, ZAIRE_CONGO becomes DR_CONGO, and COUNTIES becomes US_COUNTIES.
 - Additional map data sets are available. For example, KOSOVO, CARIBBEAN, and EUROPENUTS0 are new with MAPSGFK. (The NUTS classification is a hierarchical system for dividing up the economic territory of the European Union).
 - A more descriptive _ATTR suffix is used for attribute files (for example, ARGENTINA_ATTR instead of ARGENTI2).
 - The _ALL suffix is added for countries with dependencies and other territories. For example, compare AUSTRALIA_ALL (Australia and dependencies) to AUSTRALIA.
 - Some files were eliminated from the new maps because they were combined with other maps (for example, COUNTY and US2).
 - There is a map of the entire world named WORLD and a map file for world attributes, WORLD_ATTR.
 - The United States has some special maps that are not available for the rest of the world: US, US_STATES, and US_COUNTIES. US_COUNTIES is like the other countries.
 - There are special maps for the seven continents: AFRICA, AFRICA1, AFRICA2, ANTARCTICA, ANTARCTICA1, ASIA, ASIA1, ASIA2, ASIA3, EUROPE, EUROPE1, EUROPE2, EUROPE3, NAMERICA, NAMERICA1, NAMERICA2, OCEANIA,

OCEANIA1, OCEANIA2, SAMERICA, SAMERICA1, and SAMERICA2. Each file has an _ATTR file too.

- There are special files used for PROC GEOCODE and annotating maps: USCENTER, USCENTER_ALL, USCITY, USCITY_ALL, and WORLD_CITIES. Note that WORLD_CITIES_ALL is an unabridged data set that can be downloaded from <http://support.sas.com/rnd/datavisualization/maponline/index.html>.
- PROJPARM contains the PROC GPROJECT information for all projected maps in MAPSGFK.

4. New Variables

- There are many changes to variable names:
 - i. All the map data sets have ID, SEGMENT, LAT, LONG, X, Y, RESOLUTION and DENSITY.
 - ii. All the _ATTR data sets have ID, IDNAME, ISO, and ISONAME.
 - iii. Many map data sets have additional variables like, ID1, LAKE.
 - iv. Many _ATTR data sets have additional variables like ID1, ID1NAME, ID1NAMEU, and ID1NAMEU.
 - v. A few map data sets also have ID2, CONT, STATE and COUNTY.
 - vi. A few _ATTR data sets have ID2, ID2NAME, ID2NAMEU, and CONT.
 - vii. Table 1 shows examples in various data sets.
- The RESOLUTION variable is a more practical version of PROC GREDUCE. The levels are based on your display size. The various pixel sizes are shown below. In SAS@9.4, the RES= values in PROC GMAP appear as in the table. For example, RES=4 displays a map using all points with a resolution value less than and equal to 4. Note that RES=AUTO is the default so that the best size will be selected. (In SAS@9.3, the RES= values are high, medium, and low).

10 =	28,800 x	23,040
9 =	14,400 x	11,520
8 =	6,000 x	4,800
7 =	2,400 x	1,800
6 =	1,600 x	1,200
5 =	1,280 x	1,024
4 =	800 x	600
3 =	640 x	480
2 =	400 x	300
1 =	320 x	240

5. Data Changes The content of some variables has changed:

- ID variables are now character instead of numeric and are unique worldwide because they contain the country code.
For example, looking at STATE and ID1 in the US_COUNTIES data set, we have STATE=51, identifying Virginia, and ID1=US-51, identifying the country (US) and the state (51).

Looking at COUNTY and ID in the US_COUNTIES data set, we have COUNTY=15, identifying Augusta County, and ID=US-51015, identifying the country (US), the state code (51), and the county code (015).

- X and Y are always projected.
- LAT and LONG are always unprojected degrees. They are also based on the eastern hemisphere.

Table 1 shows some examples of maps and attribute files and the variables that they contain. The EUROPE files show an example of continent files.

<p>MAPSGFK.US_COUNTIES variables:</p> <ul style="list-style-type: none"> • ID – County code (char) • SEGMENT– ID segment number • X – Projected longitude coordinate versus unprojected radians • Y – Projected longitude coordinate versus unprojected radians • LAT – Unprojected degrees latitude • LONG – Unprojected degrees longitude (East) • DENSITY – PROC GREduce density values • RESOLUTION – Map detail level based on output resolution • ID1 – States code (char) • LAKE – Lake flag: 1-water, 2-citytype • STATE – Same as MAPS data set • COUNTY – Same as MAPS data • STATECODE – Two-letter state abbreviation 	<p>MAPSGFK.US_COUNTIES_ATTR variables:</p> <ul style="list-style-type: none"> • ID – County code (char) • IDNAME – County name • ISO – Country ISO number • ISONAME – Country name • ID1 – State code (char) • ID1NAME – State name • STATE – Same as MAPS data set • COUNTY – Same as MAPS data • STATECODE – Two-letter state abbreviation
<p>AFGHANISTAN variables:</p> <ul style="list-style-type: none"> • ID – Districts code • SEGMENT – Map segment • X – Projected longitude • Y – Projected latitude • LAT – Unprojected latitude (Y) in degrees • LONG – Unprojected longitude (X) in degrees • DENSITY – PROC GREduce density value • RESOLUTION – Similar to Density, but processed for display size • ID1 – Province code • LAKE – Lake flag: 1-water, 2-citytype 	<p>AFGHANISTAN_ATTR variables:</p> <ul style="list-style-type: none"> • ID – District code • IDNAME – District name • ISO – ISO Country code • ISONAME – ISO Country name • ID1 – Province code • ID1NAME – Province name • ID1NAMEU – Unicode version of ID1NAME
<p>BRAZIL variables:</p> <ul style="list-style-type: none"> • ID – Mesorregions code • SEGMENT – ID segment number • X – Projected longitude coordinate • Y – Projected latitude coordinate • LAT – Unprojected degrees latitude • LONG – Unprojected degrees longitude • DENSITY – PROC GREduce density value • RESOLUTION – Similar to Density, but processed for display size • ID1 – State code 	<p>BRAZIL_ATTR variables:</p> <ul style="list-style-type: none"> • ID – Mesorregion code • IDNAME – Mesorregion name • ISO – ISO Country code • ISONAME – ISO Country name • ID1– State code • ID1NAME – State name • ID1NAMEU – Unicode version of ID1NAME • IDNAMEU – Unicode version of IDNAME • IDNAME_ALT – Alternate IDNAME • IDNAMEU_ALT – Alternate IDNAME in Unicode

<p>UZBEKISTAN variables:</p> <ul style="list-style-type: none"> • ID – Districts code • SEGMENT – ID segment number • X – Projected longitude coordinate • Y – Projected latitude coordinate • LAT – Unprojected degrees latitude • LONG – Unprojected degrees longitude • DENSITY – PROC GREduce density value • RESOLUTION – Similar to Density, but processed for display size <ul style="list-style-type: none"> • ID1 – Province code • LAKE – Lake flag: 1-water, 2-citytype 	<p>UZBEKISTAN_ATTR variables:</p> <ul style="list-style-type: none"> • ID – Districts code • IDNAME – District name • ISO – ISO Country code • ISONAME – ISO Country name <ul style="list-style-type: none"> • IDNAMEU – Unicode version of IDNAME • ID1– Province code • ID1NAME – Province name • ID1NAMEU – Unicode version of ID1NAME
<p>EUROPE variables:</p> <ul style="list-style-type: none"> • ID – Alpha2 country code • SEGMENT – ID segment number • X – Projected longitude: Albers • Y – Projected latitude: Albers • LAT – Unprojected degrees latitude • LONG – Unprojected degrees longitude • RESOLUTION – Similar to Density, but processed for display size • DENSITY – PROC GREduce density value <ul style="list-style-type: none"> • IDNAME – Country name • LAKE – Lake flag: 1-water, 2-citytype • ISO – ISO Country code 	<p>EUROPE_ATTR variables:</p> <ul style="list-style-type: none"> • ID – Alpha2 country • IDNAME – ISO country name • ISO – ISO Country code • ISONAME – ISO Country name • ISOALPHA2 – ISO Alpha2-code • ISOALPHA3 – ISO Alpha3-code • CONT– Numeric number
<p>EUROPE1 variables:</p> <ul style="list-style-type: none"> • ID – Admin1 code • SEGMENT – ID segment number • X – Projected longitude coordinate • Y – Projected latitude coordinate • LAT – Unprojected degrees latitude • LONG – Unprojected degrees longitude • DENSITY – PROC GREduce density value • RESOLUTION – Similar to Density, but processed for display size <ul style="list-style-type: none"> • IDNAME – Admin1 name • ISO – ISO Country code • ISOALPHA2 – ISO Alpha2-code • LAKE – Lake flag: 1-water, 2-citytype 	<p>EUROPE1_ATTR variables:</p> <ul style="list-style-type: none"> • ID – Admin1 code • IDNAME– Country name • ISO – ISO Country code • ISONAME – ISO Country name • ISOALPHA2– ISO Alpha2-code • ISOALPHA3– ISO Alpha3-code • CONT– Numeric number <ul style="list-style-type: none"> • ID1– Country code • ID1NAME – Country name • IDNAME_ALT – Alternate admin name • ADMINTYPE – Admin1 type

<p>EUROPE2 variables:</p> <ul style="list-style-type: none"> • ID – Admin2 code • SEGMENT – Map segment • X – Projected longitude • Y – Projected latitude • LAT – Unprojected latitude (Y) in degrees • LONG – Unprojected longitude (X) in degrees • DENSITY – PROC GREduce density value • RESOLUTION – Similar to Density, but processed for display size • ISO – ISO Country code • ISOALPHA2 – ISO Alpha2-code • LAKE – Lake flag: 1-water, 2-citytype 	<p>EUROPE2_ATTR variables:</p> <ul style="list-style-type: none"> • ID – “County” code • IDNAME – “County” name • ISO – Country code • ISOName – Country name • ISOALPHA2– ISO Alpha2-code • ISOALPHA3– ISO Alpha3-code • CONT– Numeric number • ID1– Alpha2 Country code • ID1NAME – County name • IDNAME_ALT – Alternate admin name • ID2– Admin1 code • ID2NAME – Admin1 name • ADMINTYPE – Admin1 type • ADMINTYPE2 – Admin2 type
--	--

Table 1 – Variables in the Various Maps

EXAMPLES FOR MIGRATION

The following examples illustrate how to convert existing programs to use the new GfK map data sets. MAPSSAS is the old data set (formerly MAPS), and MAPSGFK is the new data set.

EXAMPLE 1 – Changing the Data Set Name

MAPSSAS data set:

```
proc GMAP map=maps.uzbekist
          data=maps.uzbekist;
  id id;
  choro id / nolegend;
run;
```

MAPSGFK data set:

```
proc GMAP map=mapsgfk.uzbekistan
          data=mapsgfk.uzbekistan;
  id id;
  choro id / nolegend;
run;
```

Note that these examples use the map data for the response data. This ensures that there is a value for each map area, but it isn't useful in the real world. Figure 2 shows the results from using the MAPS data set, and Figure 3 shows the results from using the MAPSGFK data set. As you can see, MAPSGFK has a lower level of polygonal areas.

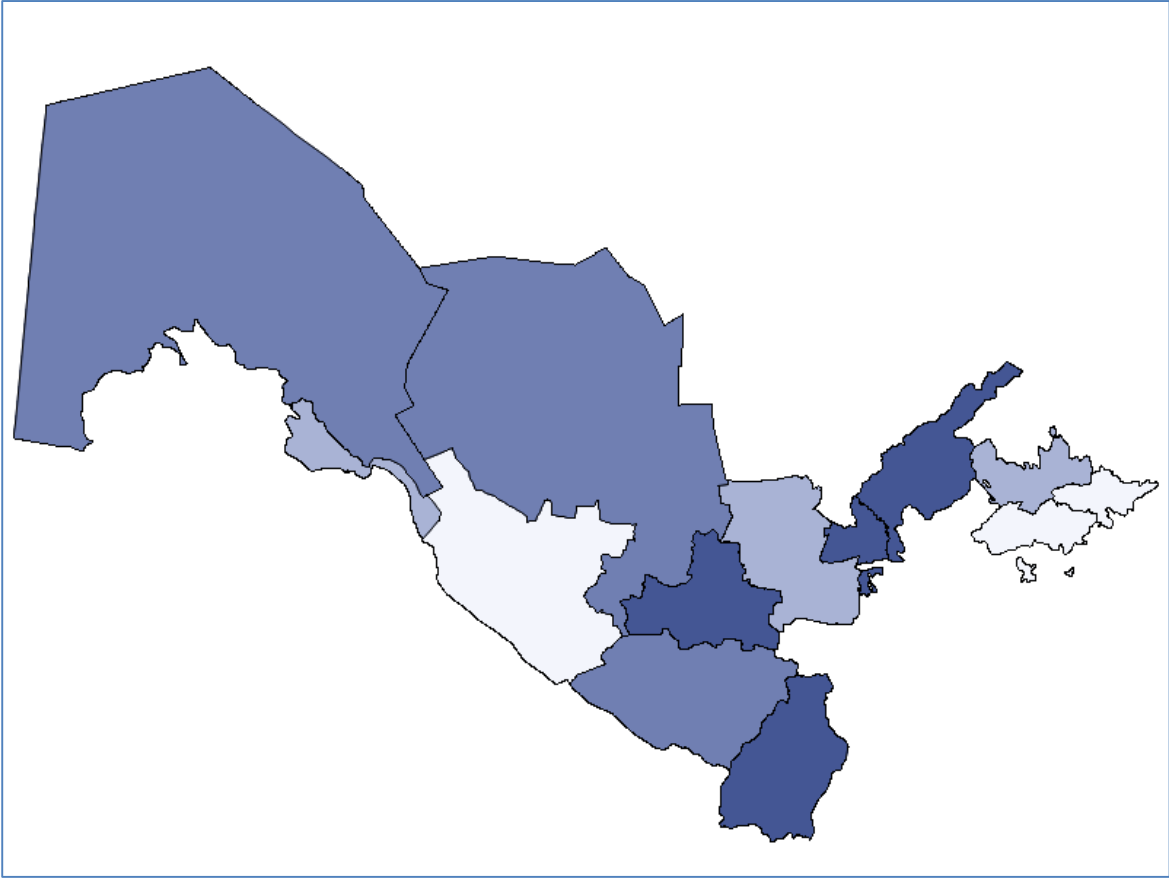


Figure 2 – Output from MAPSSAS Data Set

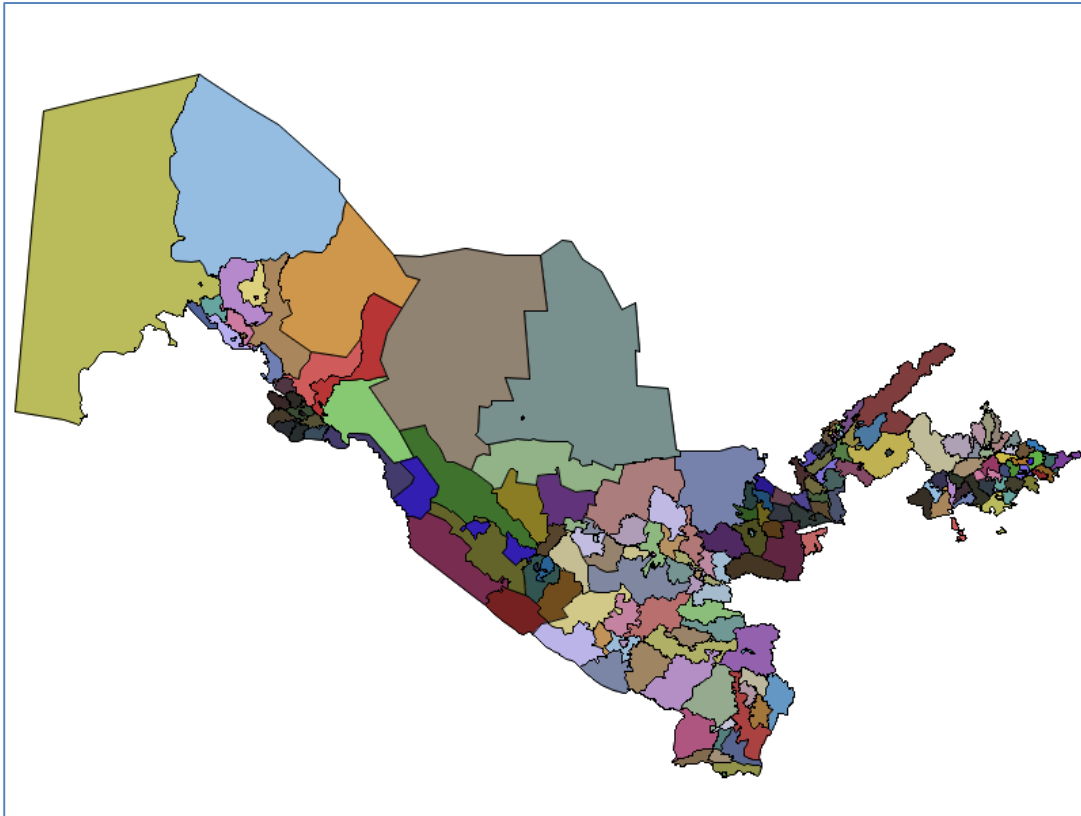


Figure 3 – Output from MAPSGFK Data Set

EXAMPLE 2 – Reducing the Number of Levels

In order to create the same map in Example 1, we need to modify the MAPSGFK program to reduce the number of map levels.

MAPSGFK data set:

```

/* Sort the map data by ID1 */
proc SORT data=mapsgfk.uzbekistam
    out=smymap; by id1id; run;

/* remove the lowest level of the map data set */
proc GREMOVE data=smymap
    out=mymap; by id1; id id; run;

/* rename ID1 to ID */
data mymap (drop=id1); set mymap;
    id=id1; run;

proc GMAP data=mymap map=mymap;
    id id;
    choro id / nolegend;
run;

```


Figure 4 shows the output from this program, which matches Figure 1.

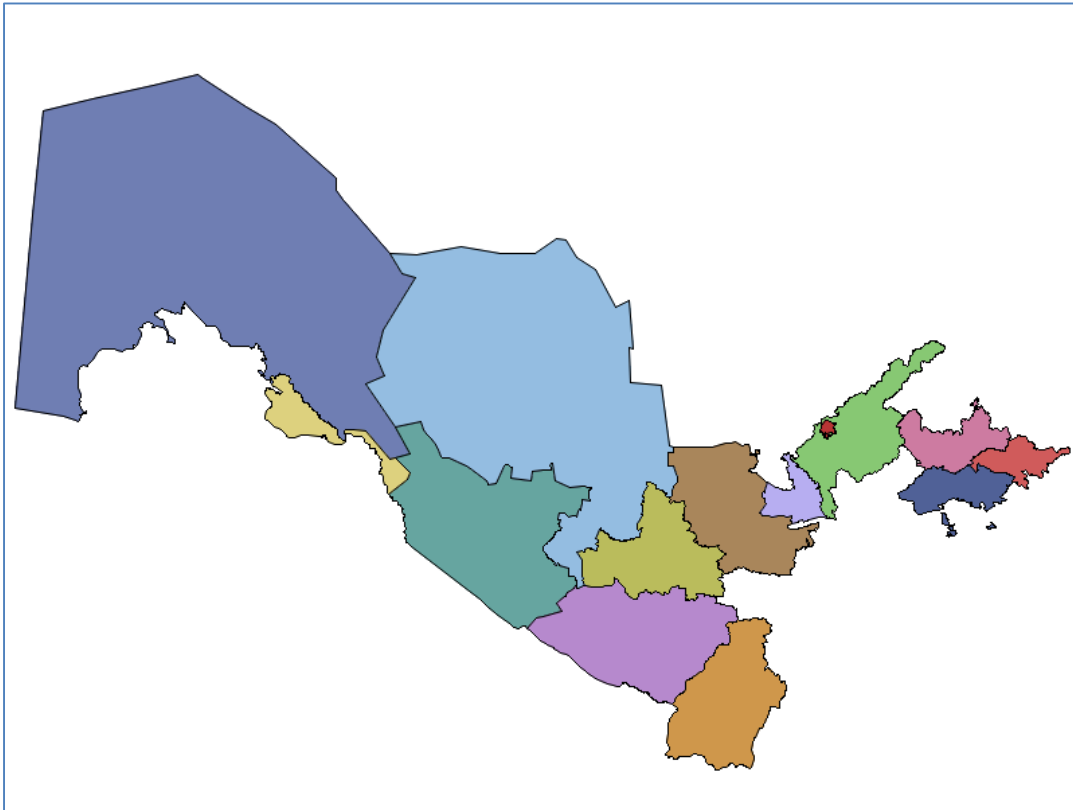


Figure 4 – PROC GREduced Output from MAPSGFK Data Set

EXAMPLE 3 – Fixing the Response Data ID Values

Examples 1 and 2 use the map data set for the response data. This ensures that the ID values always match. But real programs need to use real response data. In order to do this, you need to make the response data match the ID values used by the map.

MAPSGFK data set:

```
%let path=C:\SGF2016\NewMaps\inputdata;

/* read in the population and ensure that IDNAME is UPPERCASED */
data pop;
  length idname $55.;
  infile "&path.\uzbek.dat" delimiter=',';
  input idname population;
  idname=UPCASE(idname);
run;

/*extract one level up */
proc SORT data=mapsgfk.uzbekistan
  out=smymap; by id1 id; run;
proc GREMOVE data=smymap
  out=mymap; by id1; id id; run;
data mymap (drop=id1); set mymap;
```

```

    id=id1; run;

/* now go up a level in the _attr file */
data mymap_attr(keep=ID IDNAME);
    length lastID1 $15; retain lastID1 '';
    set mapsgfk.uzbekistan_attr;
    if (id1 ne lastID1) then do;
        lastID1=id1; id=id1; idname=upcase(idname); output;
    end;
run;

/* sort both files by IDNAME and the merge to match IDNAME to get ID
*/
proc sort data=pop; by idname; run;
proc sort data=mymap_attr; by idname; run;
data mydata; merge pop mymap_attr; by idname; run;

/* Display the map */
proc GMAP data=mydata map=mymap;
    id id;
    choro population;
run;

```

Figure 5 shows the results from this program.

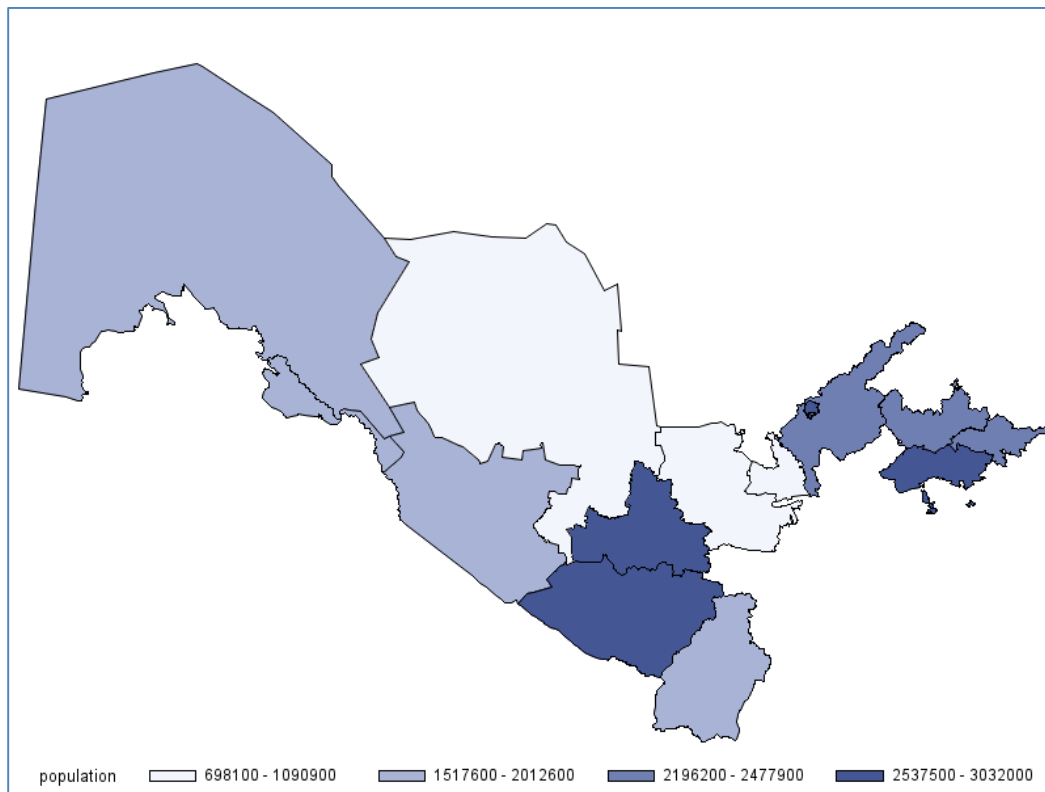


Figure 5 – Fixing the Response Data

EXAMPLE 4 – Character versus Numeric ID and Using the LEVELS= Option

The ID variables are numeric in MAPSSAS and character in MAPSGFK. If you use `LEVELS=1` with ID, it will not work the same with MAPSGFK data sets; it will be ignored. Another numeric variable, such as `SEGMENT`, must be used with `LEVELS=1` instead.

MAPSSAS data set:

```
/* want a single-colored map */  
proc gmap data=maps.europe map=maps.europe;  
  id id;  
  choro id / levels=1 nolegend ;  
run;
```

MAPSGFK data set:

```
/* need a numeric variable*/  
proc gmap data= mapsgfk.europe map=mapsgfk.europe;  
  id id segment;  
  choro segment / levels=1;  
run;
```

Figure 6 shows the output from MAPS, and Figure 7 shows the output from MAPSGFK. Notice that the map data contains different areas in MAPS and MAPSGFK as indicated by the arrows.

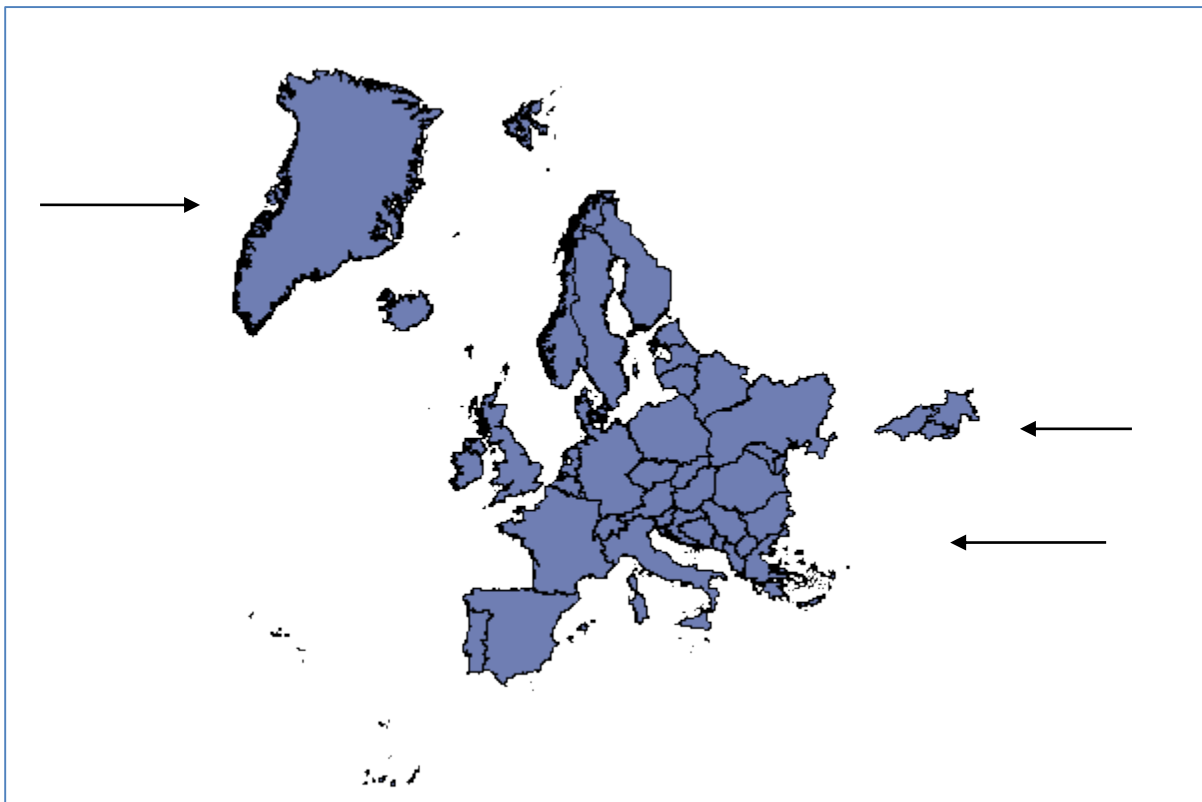


Figure 6 – LEVELS=1 Output with MAPSSAS Data Set

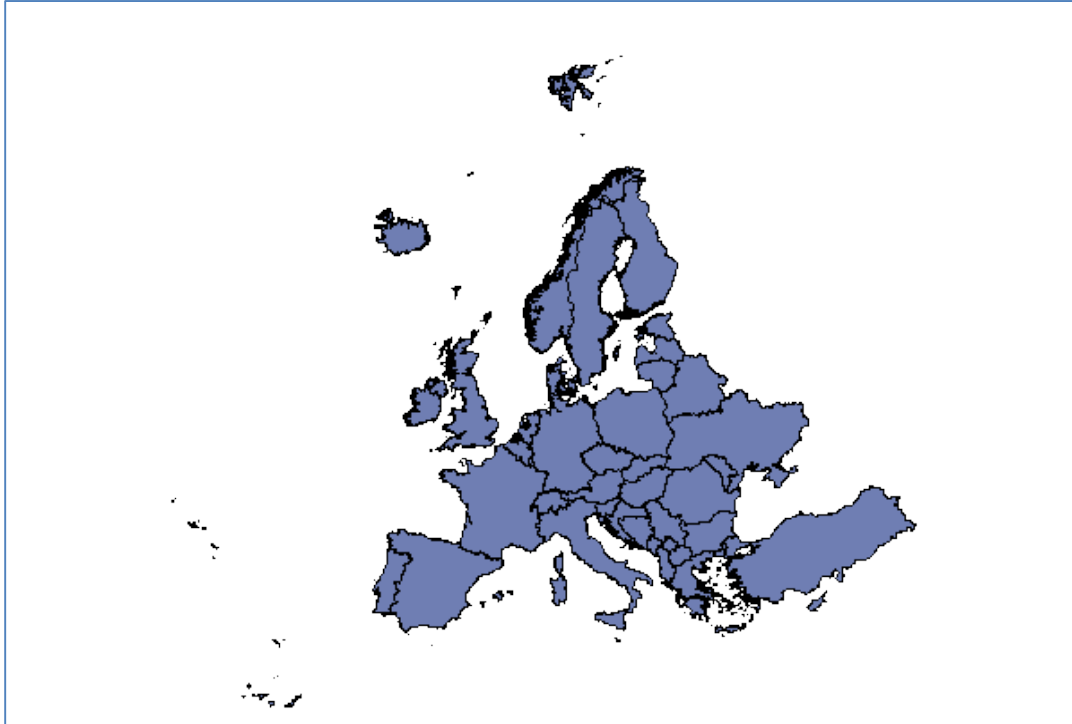


Figure 7 – LEVELS=1 Output with MAPSGFK and Modified Program

EXAMPLE 5 – Projection Differences

In the MAPSSAS data set, there was inconsistency regarding which values were projected. In the US data, X and Y were unprojected in the past. In MAPSGFK, X and Y are always projected. In this example, changing the data set name creates a difference in projection.

MAPSSAS data set:

```
proc GMAP map=maps.states data=maps.states;
  id state; /* no ID variable */
  choro state / nolegend;
run;
```

MAPSGFK data set:

```
proc GMAP map=mapsgfk.us_states data=mapsgfk.us_states;
  id id;
  choro id / nolegend;
run;
```

Figure 8 shows that MAPSSAS data appears unprojected and backward. Figure 9 shows that the X and Y values give you a projected map. Notice that there are also some differences in the polygonal areas contained in the maps, as indicated by the arrow.

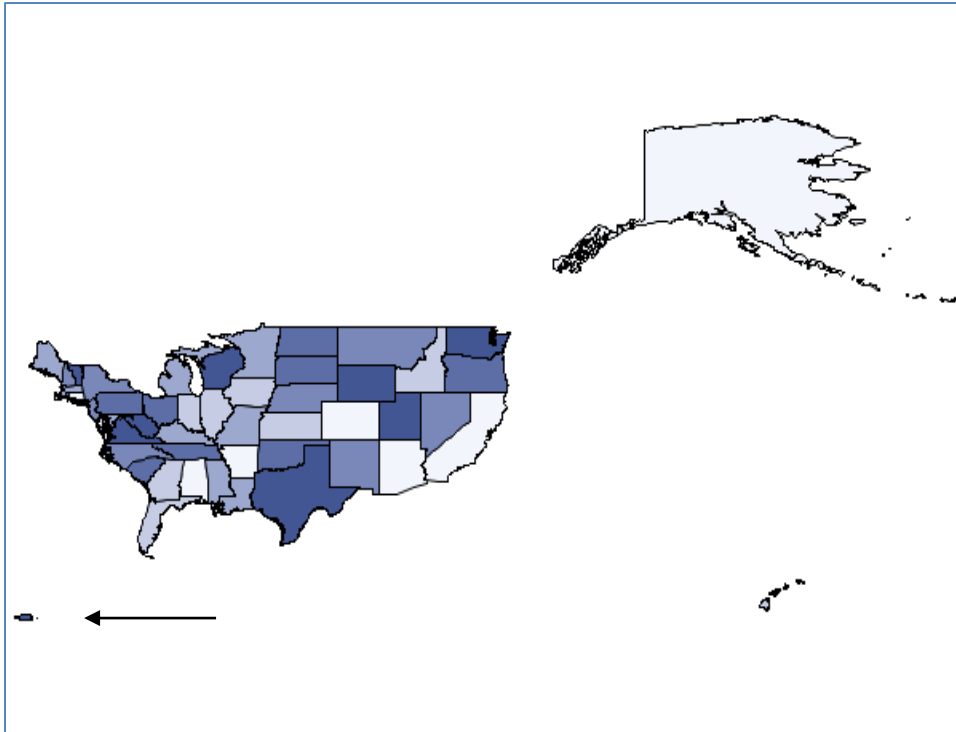


Figure 8 – MAPSSAS Data Is Unprojected

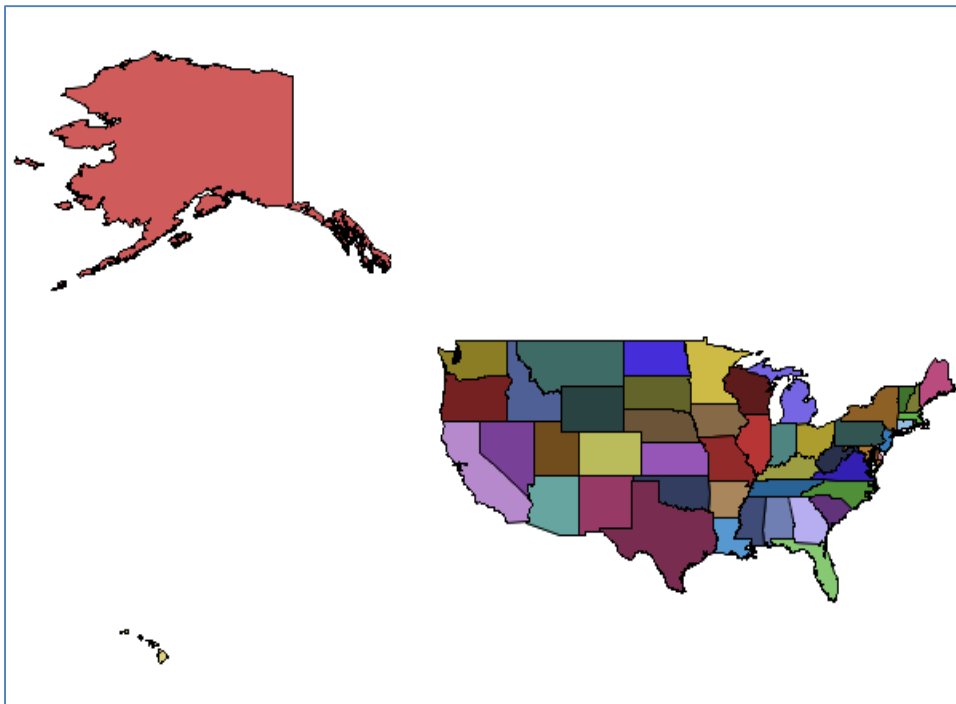


Figure 9 – MAPSGFK Is Projected

EXAMPLE 6 – PROC GPROJECT Usage

PROC GPROJECT was typically used with MAPSSAS data. This example shows how to convert code using PROC GPROJECT in the old maps (MAPSSAS) and use it in the new ones (MAPSGFK).

MAPSSAS data set:

```
proc GPROJECT data=maps.states
  out=mystates project=milller2;
  id state;
run;

proc GMAP data=mydata
  map=mystates;
  id state;
  choro state / nolegend;
run;
```

MAPSGFK data set:

```
proc GPROJECT data=mapsgfk.us_states
  out=mystates project=milller2
  degrees eastlong latlong;
  id state;
run;

proc GMAP data=mydata
  map=mystates;
  id state;
  choro state / nolegend;
run;
```

Note that because the MAPSGFK data is in degrees and uses the EAST longitude, you must add the DEGREES and EASTLONG options to PROC GPROJECT. The LATLONG option is also used to obtain the unprojected coordinates from the LAT and LONG variables instead of from X and Y.

EXAMPLE 6B – Using the Default Projection

Since X and Y are already projected in MAPSGFK data, another way to solve this problem is to use the default projection and remove the PROC GPROJECT code.

MAPSGFK data set:

```
/* Notice, GPROJECT was removed */
proc GMAP data=mydata map=mapsgfk.us_states;
  id state;
  choro state / nolegend;
run;
```

EXAMPLE 6C – Projecting Annotate Data


In the past, when you had annotate data, you had to combine it with the MAPSSAS data set and project both data sets together. If you are using MAPSGK data, you don't have to reproject the map

data if you are using the default projections. This is because a data set (PROJPARM) is created and saved with each MAPGFK data set as it is projected. This information can be referenced to project the annotate data in the same manner.

MAPSGFK data set:

```
/* ONLY project the annotate data */
proc GPROJECT data=anno out=annop
  parmin=mapsgfk.projparm parmentry=us_states;
run;

proc GMAP data=mydata map=mapsgfk.us_states
  anno=annop;
  id state;
  choro state / nolegend;
run;
```



EXAMPLE 7 – Using the Unicode Variables

Apply the \$UESC50. format to any Unicode variable to view the national characters in an English SAS UTF8 session. The following example formats the IDNAMEU variable.

MAPSGFK data set:

```
data my_attr;
  set mapsgfk.reunion_attr;
  Unicode_idname=input(IDNAMEU, $uesc50.);
run;

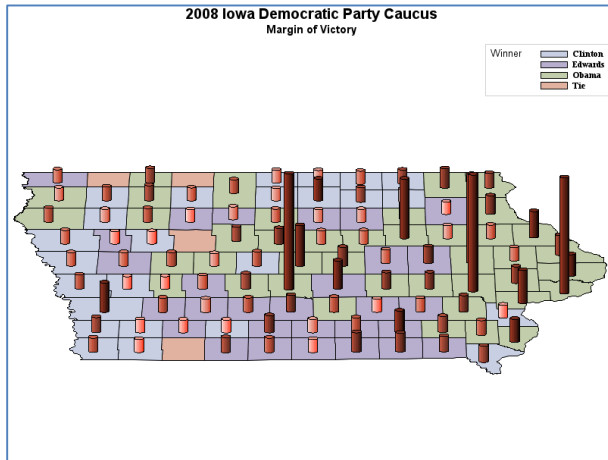
goptions ftext='albany amt';
proc GMAP data=my_attr map=mapsgfk.reunion;
  id id;
  choro Unicode_idname;
run;
quit;
```

MORE EXAMPLES

There are more examples available with this paper. Look up the title of this paper in the Proceedings (<http://support.sas.com/resources/papers/proceedings16>). Under the title will be a link that reads: Download the data files (ZIP).

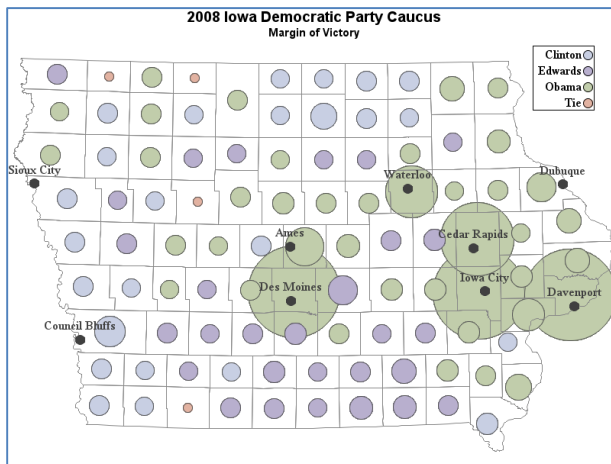
The first set of examples starts with the name “GFK”. These are examples that were written for past SAS Global Forums or SUGIs that have been rewritten to use the MAPSGFK maps. You can find the originals at <http://support.sas.com/rnd/papers/>.

GFKBlock.sas



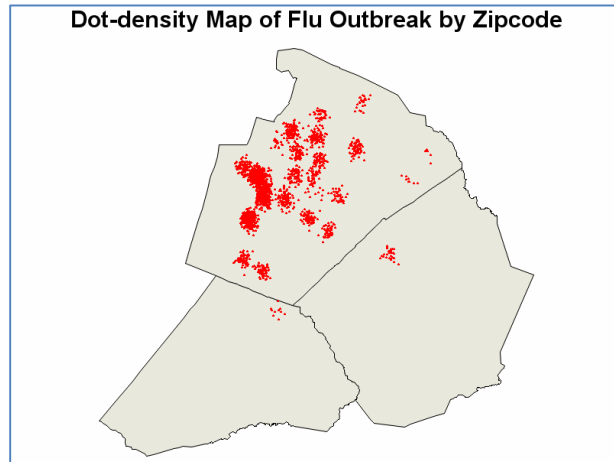
Original was from: SAS Global Forum 2009.
The paper was: Tips and Tricks IV: More SAS/GRAPH Map Secrets.
The original sample was from: iowabubble.sas.
Purpose: To create an Iowa election block map.

GFKBubble.sas



Original was from: SAS Global Forum 2009.
The paper was: Tips and Tricks IV: More SAS/GRAPH Map Secrets.
The original sample was from: iowabubble.sas.
Purpose: To create an Iowa election bubble map.

GFKdotdensity.sas



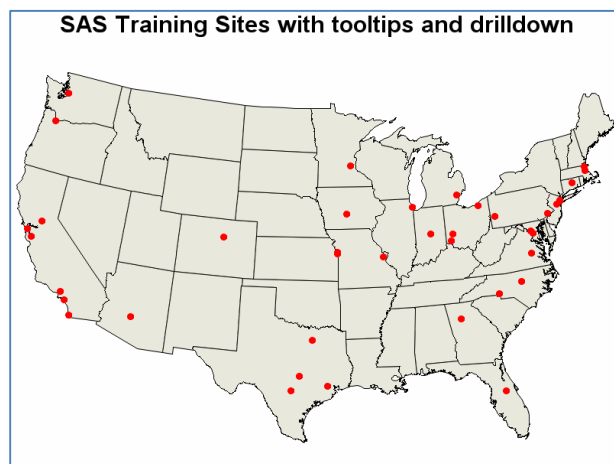
Original was from: SUGI 29

The paper was: Tips and Tricks II: Getting the most from your SAS/GRAPH Maps.

The original sample was from: dotdensity.sas.

Purpose: To simulate a dot-density map.

GFKdrillhover.sas



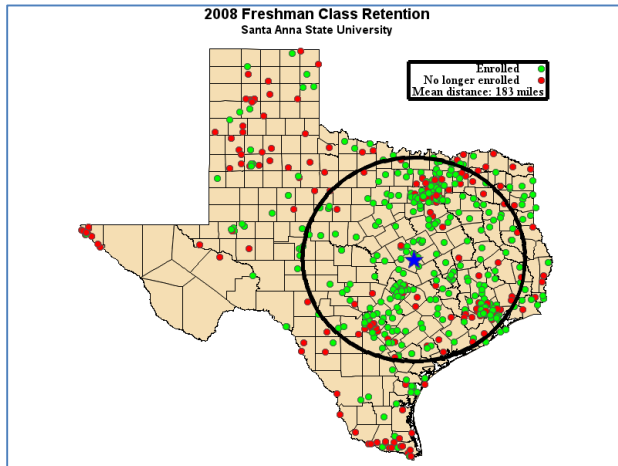
Original was from: SUGI 29

The paper was: Tips and Tricks II: Getting the most from your SAS/GRAPH Maps.

The original sample was from: drillhover.sas.

Purpose: To add drilldown and tooltips to a choro map and an annotated map with points.

GFKsasu2.sas



Original was from: SAS Global Forum 2009.

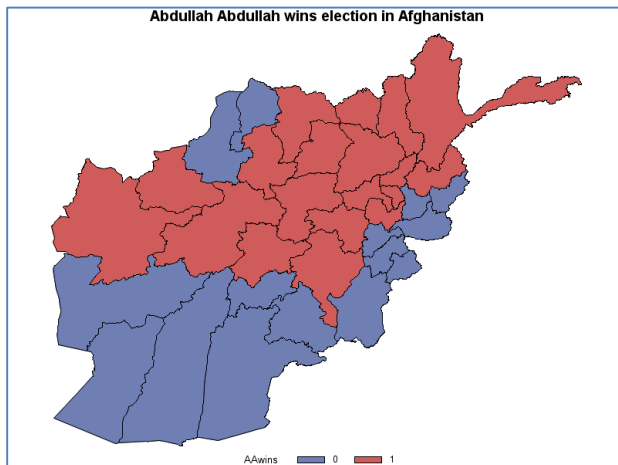
The paper was: Tips and Tricks IV: More SAS/GRAPH Map Secrets.

The original sample was from: sasu2.sas.

Purpose: To show a student distribution with average distance circled on a map.

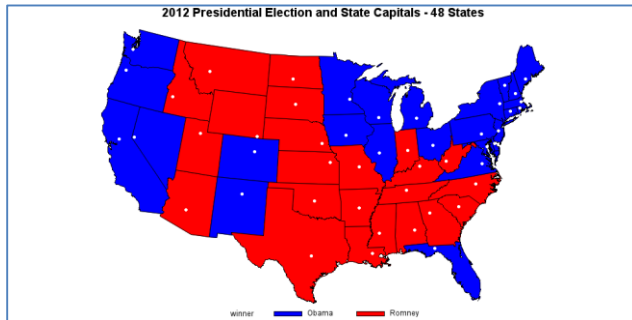
The second set of examples contains new programs. They show how to use MAPSGFK maps. Examples include showing election results by state, showing a choropleth map and annotated points, putting two countries' maps together, and processing a group of continent maps and a country map.

AfghanElection.sas



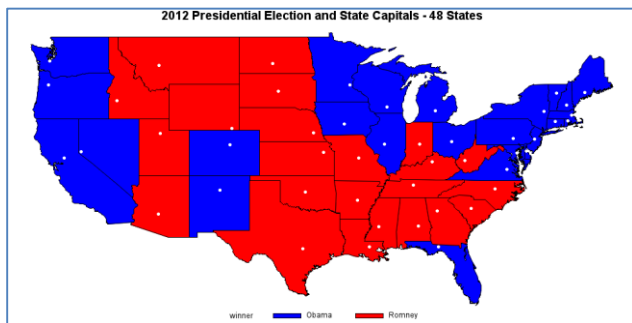
This example shows a choro map of the Afghanistan election. It shows the election results by state.

ChoroAnno.sas



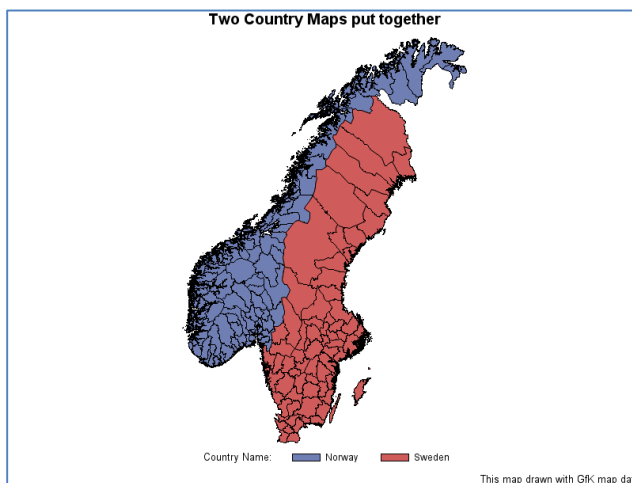
This example shows a choro map of the 2012 Presidential election with annotated points of the capitals of each state. It uses PROC GEOCODE to convert each city to a latitude and longitude.

ChoroAnno2.sas



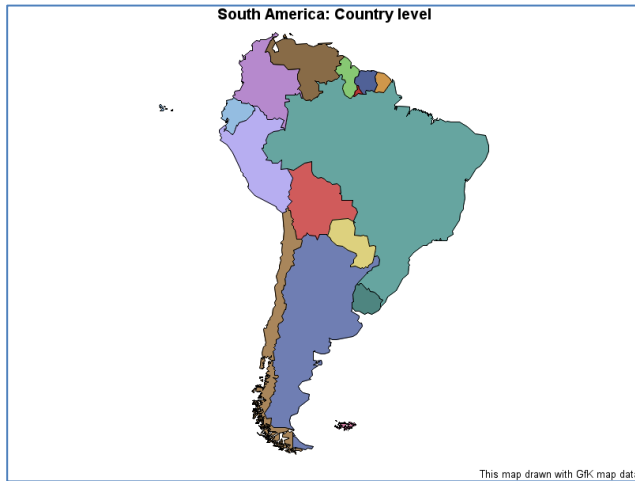
This is identical to ChoroAnno, but it uses the projection of MAPSGFK.US_STATES to project the dots.

NorwaySweden.sas

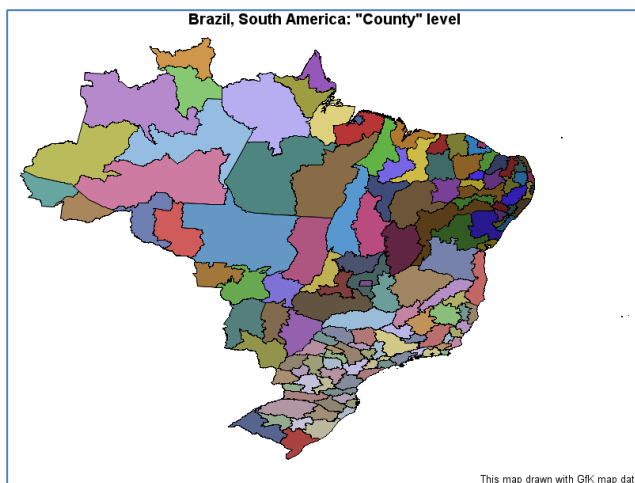


This example puts two country maps together into one map.

Samerica.sas



The first part of this example illustrates how to show the country, state, and county maps for a continent map.



The second part of this example shows how to display the state and county parts of a country map.

CONCLUSION

In this paper, we describe the problems with the old MAPSSAS data, discuss the new MAPSGFK maps, and give examples of how to use the MAPSGFK maps.

RESOURCES

“PROC Geocode: Finding locations outside the US.” SAS Presentations at SAS Global Forum 2012, Cary, NC: SAS Institute Inc. Available <http://support.sas.com/rnd/papers>

“Tips and Tricks II: Getting the most from your SAS/GRAPH maps.” Papers from SUGI 29, Cary, NC: SAS Institute Inc. Available <http://support.sas.com/rnd/papers>

“Tips and Tricks IV: More SAS/GRAPH Map Secrets.” SAS Presentations at SAS Global Forum 2009. Cary, NC: SAS Institute Inc. Available <http://support.sas.com/rnd/papers>

SAS Maps Online Web Site: <http://support.sas.com/rnd/datavisualization/mapsonline/index.html>

For additional maps, visit the SAS landing page at the GfK web site: GfK-GeoMarketing.com/SAS

CONTACT INFORMATION

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

Darrell Massengill
SAS Institute
SAS Campus Drive
Cary, NC 27513
Darrell.Massengill@sas.com
<http://www.sas.com>

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.