PROC GMAP Example

A Simple Choropleth Map

This example draws a map of Europe, minus Greenland and Svalbard and shows the Population Annual Growth Rate for each country. The ALL option causes all polygons to be drawn even if there is no data for that country. CDEFAULT=red will cause countries without a value in the DATA= dataset to be colored red.

```sas
proc gmap maps=maps.europe(where=(id ne 405 and id ne 845)) data=sashelp.demographics(where=(cont=93)) all;
  id id;
  choro popAGR / cdefault=red;
run;
quit;
```

PROC MAPIMPORT Example

Importing a ZCTA Shapefile

PROC MAPIMPORT imports a shapefile into a SAS map data set that can be used with PROC GMAP. The U.S. Census Bureau web site has many different types of shapefiles available for downloading. A ZIP Code Tabulation Area (ZCTA) is commonly used to approximate a ZIP Code area on a map. The following code reads in the shapefile and outputs the myzcta data set. The ID statement forces zcta5ce10 to be a unique ID for the map polygons.

```sas
proc mapimport out=myzcta datafile="C:\tl_2010_37_zcta510.shp";
  id zcta5ce10;
run;
```

PROC GINSIDE Example

Finding the Points Inside of the State

This example uses PROC GINSIDE to determine which points fall inside or outside of the state of North Carolina. The points represent store locations. 1 Subset the map data set to only the state of NC, 2 specify the stores data set which contains the store locations and 3 create an output data set containing stores inside of NC. Note that both the MAP= and DATA= data must be unprojected or projected the same and both must be in radians or degrees.

```sas
proc ginside map=map (where=(state=037)) data=stores out=inside;
  id state;
run;
```

The output on a map shows that 2 points (red) are outside of NC:

Mapping Tips

* The SAS Annotate Facility can be combined with PROC GMAP to add points, lines, text, images and other graphics to your map.

* If PROC GMAP draws a CHORO map with stray lines, it usually means that the ID variable specified is not unique for the map data. The stray lines are drawn as GMAP tries to connect separated polygons.

* Use the STATISTICS= option with the PROC GMAP AREA, BLOCK, CHORO, PRISM and SURFACE statements to set the statistic for how GMAP will chart your data. If the STATISTICS option is not used and your response data contains multiple observations for the same polygonal area, then only the first value is used.

* PROC GPROJECT can be used to clip your map without projecting it. Set PROJECT=NONE and set the clipping area coordinates with LONGMIN=, LONGMAX=, LATMIN= and LATMAX=.

* By default, PROC GINSIDE will exclude points that fall exactly on the polygonal border because they could fall in multiple polygonal areas. The INCLUDEBORDER option will cause them to be counted in the first polygon processed.

* The ID statement should be used with PROC MAPIMPORT to ensure that the ID is unique for the polygons.

For complete information, refer to the SAS 9.4 documentation at support.sas.com/documentation.
The GMAP procedure displays the map. The ID statement designates the matching variable between the response data and the map data. Each of the subsequent BLOCK, CHORO and PRISM statements display the map in a different way. BLOCK places a bar with a height based on the response value at the centroid of each map region. CHORO colors each region based on a response. PRISM raises (extrudes) the map region to a height based on the response value. The AREA statement can be used with the BLOCK and PRISM statements to alternately control the color of the regions. Each of those statements has a number of common options that are listed at the end of the section.

**PROC GMAP Syntax**

```
Common Options
The following options are available for the BLOCK, CHORO and PRISM statements:
- ANNOTATE=annotate-data-set
- CDEFAULT=empty-area-fill-color
- CEMPTY=empty-area-line-color
- COUTLINE=area-line-color
- DESCRIPTION=description
- HTML=variable-name
- HTML1_LEGEND=variable-name
- NAME=name
- STRETCH
- URL=character-variable
- WOUTLINE=area-outline-width
- XSIZE=xsize
- YSIZE=ysize

The following options are available for the AREA, BLOCK, CHORO and PRISM statements:
- DISCRETE
- LEGEND=LEGEND<1..99>
- LEVELS=number-of-response-levels | ALL
- MIDPOINTS=value-list | OLD
- MISSING=NOLEGEND | PERCENT RANGE
- STATFMT=format-specification
- STATISTIC=FIRST | SUM | FREQ | MEAN
- UNIFORM
```

**PROC GINSIDE Syntax**

The GINSIDE procedure associates polygon attributes like country, state, or postal codes in the MAP data set with points in the input DATA set.

```
PROC GINSIDE
DATA=input-data-set
MAP=map-data-set
<OUT=output-data-set>
<INSIDEONLY> <INCLUDEBORDER>
ID id-variable-name-1 <...id-variable-name-n>
```

**PROC GPROJECT Syntax**

The GPROJECT procedure alters the latitude/longitude coordinates of a map data set to present the map more favorably, including clipping the map to extents.

```
PROC GPROJECT
<Data=input-data-set>
<OUT=output-data-set>
<Project=(<DATELINE> <DEGREES>)
<DUOPOLY>
<EXTRUDING>
<TEMPLE>
<TRANSFORM>=(<LATMIN>=min-latitude | LONGMIN=long-latitude | LONGMIN=long-minimum-longitude | LONGMIN=long-max-longitude | LATMIN=lat-max-latitude | LONGMIN=long-max-longitude | LATMIN=lat-min-longitude | LONGMIN=long-min-longitude | MERIDIAN=central-longitude | NODATELINE>)
<PARALL1>=latitude | <PARALL2>=latitude | <POLELAT>=latitude | <POLELON>=longitude | <RSSIZE>=<DEGREES>
<OUT><DATA>=set>
```

**PROC GREMOVE Syntax**

The GREMOVE procedure removes interior boundaries of a hierarchical map. For instance, it can remove US county boundaries, leaving only state outlines.

```
PROC GREMOVE
<Data=input-data-set>
<OUT=output-data-set>
ÂNCE=(<DESCENDING> <OUTER-NAME-1 <...<DESCENDING> > <variable-NAME-1>)
<OUT><DATA>=set>
```

**PROC GREMOVE Syntax**

The GREMOVE procedure generalizes the vertices of polygons to remove superfluous points from the border.

```
PROC GREMOVE
<Data=input-data-set>
<OUT=output-data-set>
<EXCLUDE> (<MINIMUM-DISTANCE>)
<NUMPOINTS>
<NOCLEAN>
ID id-variable-name-1 <...id-variable-name-n>
```

**PROC MAPIMPORT Syntax**

The MAPIMPORT procedure creates SAS map data sets from Esri shape files.

```
PROC MAPIMPORT
<Datafile=filename>
<DATAFILE=filename>
<CONTENTS>
<CREATE_ID>
ID field-identifier(s)
EXCLUDE field-identifier(s)
SELECT field-identifier(s)
RENAME field-identifier-1 = variable-name-1 <...<field-identifier-n = variable-name-n>
```