Using SAS Maps in Institutional Research

Andrew Merrill, Carnegie Mellon University, Pittsburgh, PA

Abstract

Creating a visual representation of data can be very useful way to demonstrate patterns that are not readily apparent in a table. Carnegie Mellon University's Institutional Research office uses SAS mapping software to display data in a user-friendly way. SAS 9.3 now allows anyone to make maps using the base software package. This poster shows the reader how to output Institutional Research data using the SAS GMAP procedure. Examples include maps displaying enrollment, alumni, and study abroad data.

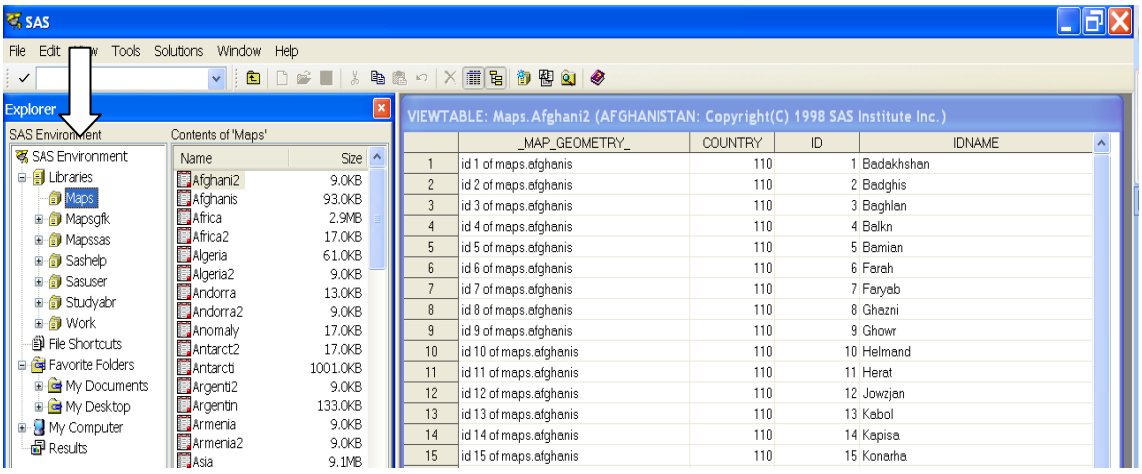
Introduction

One of the responsibilities of the Office of Institutional Research & Analysis (IRA) at Carnegie Mellon University is publishing yearly data and statistics in the institution's fact book. Recently, the IRA office has begun to make better use of maps in order to display data visually. Originally, maps only displayed U.S. enrollments and alumni residence by state. In 2011, IRA decided to start displaying enrollment and alumni data in the U.S. and the world. Specifically, the maps displayed in the fact book included enrollment data on first-year, undergraduate, graduate, international students. These new maps allow readers to visualize where students are coming from and where they are going after graduation in both the U.S. and in the world.

SAS has made improvements to their 9.3 base program that allows users to create professional looking maps with much greater ease. Specifically, the inclusion of anti-aliasing into the GMAP procedure makes lines look much smoother compared to earlier versions. SAS users now have quick access to maps by world, country, state, and county. Users of SAS 9.2 or 9.1 may need to download additional maps and configure their SAS software properly to create maps. Additional and archived maps are available for download at: http://support.sas.com/rnd/datavisualization/mapsonline/html/home.html

Available Maps

SAS 9.3 comes with maps pre-loaded onto your computer. You can view all the available maps by opening the library titled maps. You will not be able to see an image of the map until you run the GMAP procedure, which will be covered extensively in the rest of this paper. The screenshot below shows a partial list of maps that come with base SAS 9.3



The GMAP Procedure

Understanding how PROC GMAP works is the most critical part to creating and editing your own maps. The following is a brief explanation of the procedure, which highlights the main options used by the IRA office at Carnegie Mellon. However, there are many more options that are not listed in this paper. If you want to know about all of the options available, you should refer to SAS help or support.sas.com.

PROC GMAP

Syntax:

DATA=*response-data-set* -this dataset will contain actual values for information such as enrollment
MAP=*map-data-set* -this references the map being used from the map library
<*ALL*> this option is used if you wish to display the entire map, even if data is missing. This should be excluded if you wish to remove places like Antarctica or Greenland
ID -*id-variable-name* –ID refers to the variable that defines the map area. For instance, on a U.S. map the ID would likely be state. For the U.S. map, (maps.us) the ID is the variable statecode
CHORO –*response-variable-name* –CHORO (Choropleth) refers to two-dimensional. You can also define the map as BLOCK, PRISM, and SURFACE for three-dimensional maps. However, this paper only addresses CHORO maps.
LEGEND= *Legend<1…99>* -Legends are defined before running the GMAP procedure.
DESCRIPTION= '*Name*'(Alias DES=) –Description names the file in the results window
COUTLINE= '*Color*' – The COUTLINE option allows you to change the map outline and legend colors. Typically, black is used show borders.
ANNOTATE=*annotate-data-set* The Annotate option allows you to reference specific data sets for the CHORO statement. For instance, you can place specific cities or capitals on your maps if they are defined in an earlier annotate statement.
NAME='*name'* –specifies the name of the image output.

Example of the GMAP Procedure that will output a U.S. map of first-year students:

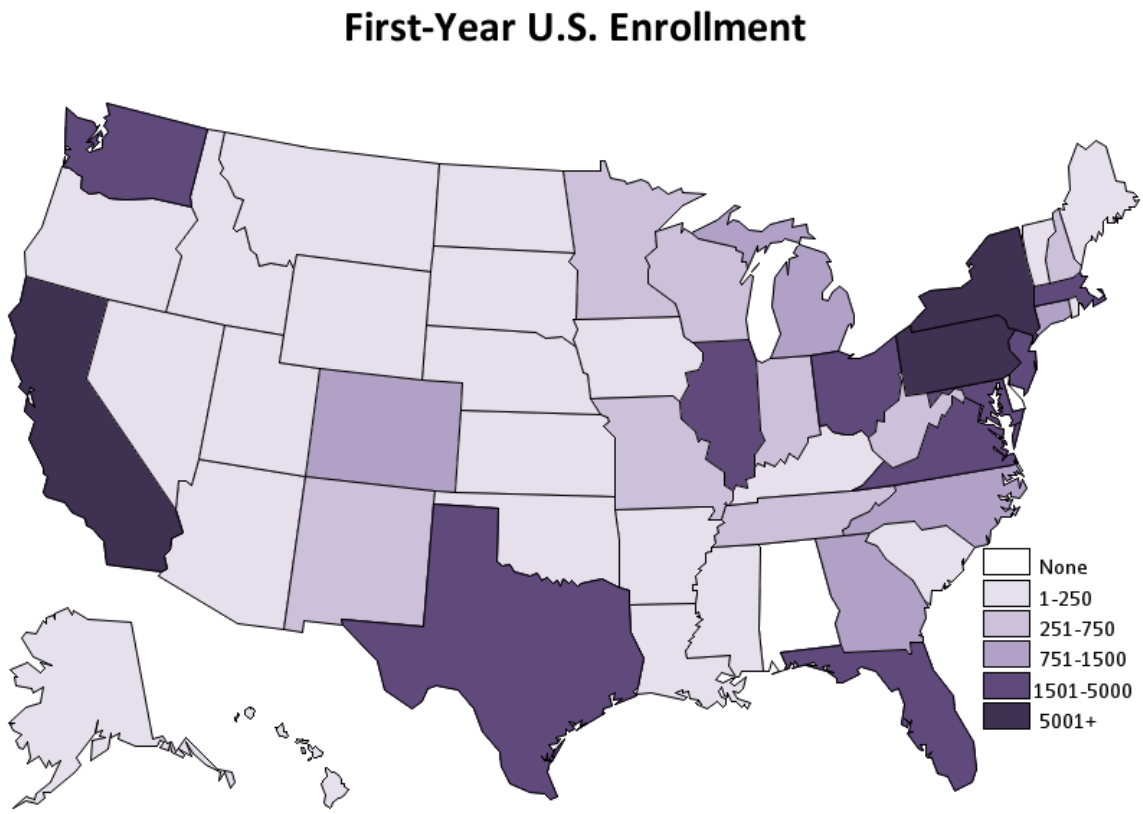
```
proc gmap data=firstyear map=maps.us;
   format students firstyear.;
   id state;
   choro students/ legend=legend1 discrete  coutline=black
   des='FirstYear' name="FY2012";
run;
quit;
```

Before Submitting the GMAP Procedure

In order for the GMAP procedure to run correctly, you will need to add a few additional pieces of syntax. Specifically, you will need to decide if you want to import your data or use a datalines statement. Additionally, there are options which allow you to edit your formats, the map legend, map colors, fonts, titles and many other features to make your map unique. Many of these options will be discussed throughout the following examples.

Example #1: First-Year Enrollment in by U.S. State

The map in the first example shows a U.S. color map of first-year enrollment with six ranges. You can choose any color you want or simply use a grayscale. Additionally I've added a title, but this is not necessary. Here is what the output will look like:



The first thing that you will need to decide before creating your map is what data you want to use. You can reference an existing SAS dataset, import a dataset from an external source (discussed in example 2), or use a datalines statement. For this example we will use a datalines statement to create set with values that range from one to six. We're using a range from one to six because it is more efficient than typing individual enrollment for each state. Here is an example:

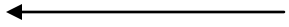
```
data work.firstyear_map;
   input statecode $ 1-2 students;
   datalines;
AL 1
AK 2
AZ 2
AR 2
CA 6
CO 4
CT 4
DE 1
FL 5
GA 4
HI 2
```

The variable statecode in the first column refers to the state abbreviation. The second column is our defined range of students from one to six

```
ID 2
IL 5
IN 3
IA 2
KS 2
KY 2
LA 2
ME 2
MD 5
MA 5
MI 4
MN 3
MS 2
MO 3
MT 2
NE 2
NV 2
NH 3
NJ 5
NM 3
NY 6
NC 4
ND 2
OH 5
OK 2
OR 2
PA 6
RI 2
SC 2
SD 2
TN 3
TX 5
UT 2
VT 2
VA 5
WA 5
WV 3
WI 3
WY 2;
run;
```

Next you will need to define the value of students using a format procedure. These can be defined in any range you prefer. The range you choose to define here should match the legend range that will be discussed later.

```
proc format;
value students
1 = "None"
2 = "1 - 250"
3 = "251 - 750"
4 = "751 - 1500"
5 = "1501 - 5000"
6 = "5001+";
run;
```



This is the value for the variable students that was just created in the datalines statement

Lastly, you'll need to define your map options such as legend colors, map title (if desired), fonts, map size and various other options. The following code shows how you can make a map with six different colors that range from light purple to dark purple.

```
%let name=firstyear;
filename odsout 'C:\output_folder';
goptions reset = global cback =white colors = (black) border;

ODS LISTING CLOSE;
ODS HTML path=odsout body="&name..htm"
(title="First-Year U.S. Enrollment")
style=sasweb;

pattern1 value=solid color=WHITE;
pattern2 value=solid color=CXE5E0EC;
pattern3 value=solid color=CXCCC0DA;
pattern4 value=solid color=CXB2A1C7;
pattern5 value=solid color=CX60497B;
pattern6 value=solid color=CX3F3151;

goptions gunit=pct htitle=6 ftitle="Calibri/Bold" htext=2.7
ftext="Calibri";
```

Macro variable, output location, background color, and border color

Change the title of your map here or use a single space to leave it blank

You can change the colors of your legend using the pattern<#> option. SAS recognizes basic colors like white, but uses hexadecimal codes instead of RGB (Red, Green, Blue) color codes. See the appendix for code on how to convert RGB colors to hexadecimal.

You can edit any of the above options to control the layout of your titles and legend: htitle=*title-size*, ftitle=*title-font*, htext=*legend- font-size*, ftext=*legend-font*

```
legend1 label=none shape=bar(4,3) value=(justify=left 'None' '1-250'
'251-750' '751-1500' '1501-5000' '5001+')
across=1 origin=(85,12) mode=share;
title1 lspace= 5 "First-Year U.S. Enrollment";
```

Your legend should match your values from the format procedure earlier. However, you can name each value in your legend in whatever way you prefer. I have listed mine from lowest to highest and matched them with the six colors values from lightest to darkest. The following definitions may also be useful:
shape=bar(<#,#>) – size of the colored bars in your legend,
origin=(<#,#>) – placement of the legend on the output
lspace=<#> - The number of line spaces before your title begins.

```
proc gmap data=work.firstyear_map map=maps.us;
 id statecode;
 choro students / midpoints = 1 2 3 4 5 6
 coutline=black
 legend=legend1
 des='FirstYear' name="FY2012";
run;
```

Run the GMAP Procedure by referencing your data (work.firstyear) and your map (maps.us). You will also need define the following options:
ID- this is the variable used to differentiate between states or countries. Often it is simply labeled id and can be found in the map viewtable
CHORO-refers to your actual data figures. In this example students references our enrollment numbers. The midpoints are the six ranges we defined earlier.

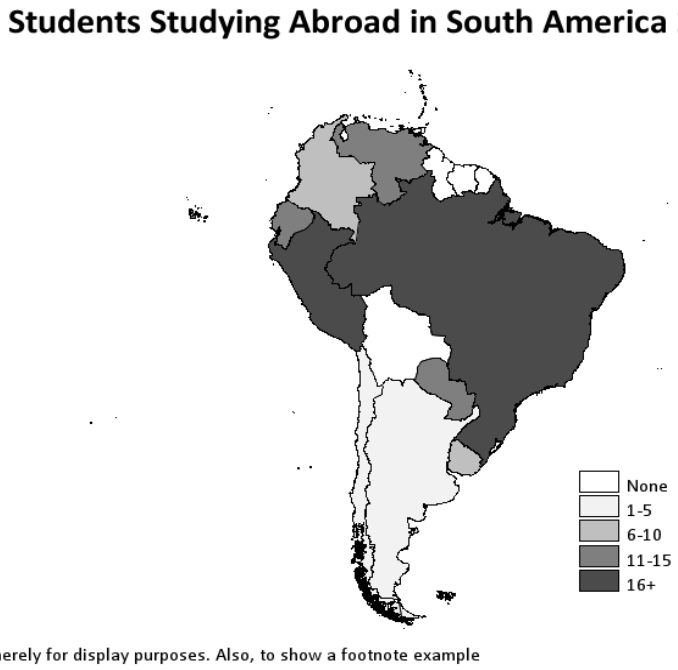
COUTLINE- is the outline color of state or country borders
LEGEND- this is the legend we defined in earlier syntax
DES- stands for description and is the name of the file written to the SAS results window
NAME-is the actual file name that will be stored on your computer

```
quit;
ODS HTML CLOSE;
ODS LISTING;
```

Lastly, you will need to end the ODS statement with a quit and close statement.

Example #2: Students Studying Abroad in South America

Another way Carnegie Mellon makes use of the GMAP procedure is visualizing where students are studying abroad. In the following example, I'll show how to import data from an Excel file and use that data to create a map of South America. Here is the map we will be making in this example:




```
*1. Import the data and rename variables*;
proc import datafile='C:\StudyAbroad.xlsx'
   dbms=excel
   out=work.studyabroad
   replace;
run;

data work.studyabroad;
   set work.studyabroad;
   rename Country=IDNAME;
   rename Study_Abroad_Students= sastudents;
run;
```

Renaming variables can be useful, but not always necessary

```
*2. Create a new dataset so that proc gmap understands the country
reference*;
data work.studyabroadmap;
   set maps.sameric2;
   keep ID idname;
run;
```

The dataset sameric2 matches country and ID. samerica uses variables within the dataset to draw the map

```
*3. Sort the data sets or the merge will not work*;
Proc sort data=work.studyabroad;
   by IDNAME;
run;

Proc sort data=work.studyabroadmap;
   by IDNAME;
run;

*4. Merge the datasets so that SAS can comprehend which country is
which in the GMAP procedure*;
data work.combined;
   merge work.studyabroad work.studyabroadmap;
   by IDNAME;
run;

*5. Create a new variable that categorizes students from 1 to 5*;
data work.combined;
   set work.combined;
   if sastudents ge 1 and sastudents le 5 then students = 2;
   else if sastudents ge 6 and sastudents le 10 then students = 3;
   else if sastudents ge 11 and sastudents le 15 then students = 4;
   else if sastudents ge 16 then students = 5;
   else students = 1 ;
run;

*6. Use a format function to change the names of the values of the
variable student*;
proc format;
   value students
   1 = 'None'
   2 = '1 - 5'
   3 = '6 - 10'
   4 = '11 - 15'
   5 = '16+';
run;
```

Notice that the format for 'None' is equal to 1 instead of being equal to zero.

```
*7. Choose options and run PROC GMAP*;
%let name=students;
filename odsout 'C:\output_location'; *You will need to name your
output location for the physical file here*;
goptions reset = global cback =white colors = (black) border;

ODS LISTING CLOSE;
ODS HTML path=odsout body="&name..htm" (title="Study Abroad in South
America") style=sasweb;

pattern1 value=solid color=CXFFFFFF;
pattern2 value=solid color=CXF2F2F2;
pattern3 value=solid color=CXBFBFBF;
```

```
pattern4 value=solid color=CX7F7F7F;
pattern5 value=solid color=CX4B4B4B;

goptions gunit=pct htitle=5 ftitle="Calibri/Bold" htext=2.7
ftext="Calibri"; *htitle=title size, htext=legend size, ftitle=title
font*;

legend1 label=none shape=bar(4,3) value=(justify=left
'None' '1-5' '6-10' '11-15' '16+')
across=1 origin=(70,10) mode=share;
```

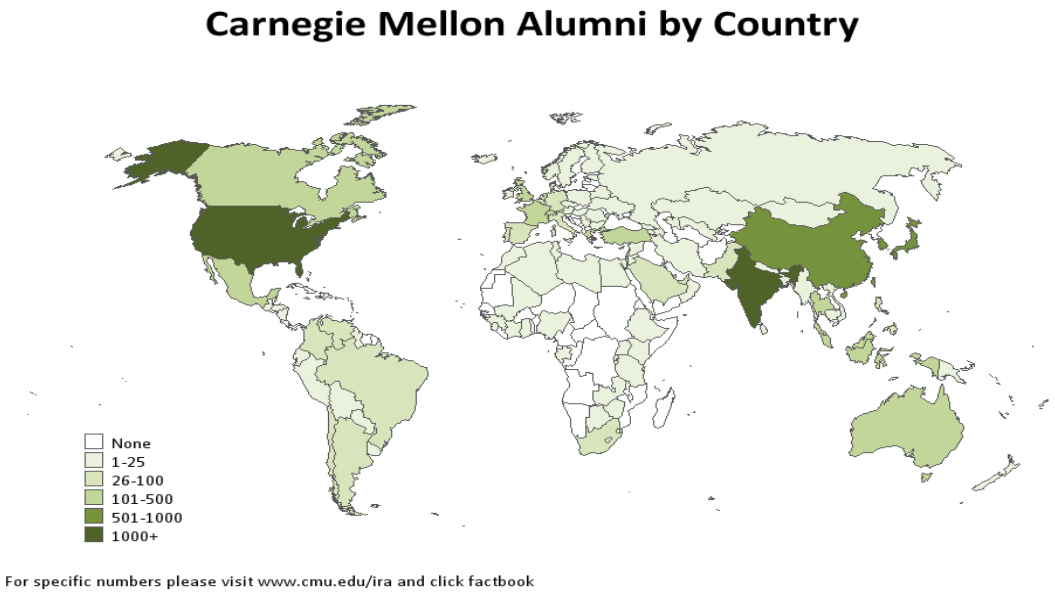
For example 2, I moved the legend by changing the origin. The first number is on the x axis, while the second is on the y axis.

```
title1 lspace= 5 "Students Studying Abroad in South America 2012";
*lspace=line space from the top*;
footnote1 justify=left "This data is merely for display purposes. Also,
to show a footnote example";

proc gmap data=work.combined map=maps.samerica;
  id id;
 choro students / midpoints=1 2 3 4 5
 coutline=black
 legend=legend1
 des='StudyAbroad' name="SouthAmericaStudyAbroad";
run;

quit;
ODS HTML CLOSE;
ODS LISTING;
```

Deleted a midpoint in the gmap procedure

Example #3: Alumni residence by Country

The last example I'll show is a world map of Alumni by country. This map is a bit more difficult to make simply because there are so many datalines. However, you can always use the import procedure to avoid this issue. Here is the world map that the code below will output:



```
%let name=alumni;
filename odsout ' C:\output_location'';

proc sql;
   create table world as
   select -1*long as x, lat as y, segment, cont, id as country
   from maps.world
   where (density<=1) and (segment<=3) and
   (country ^= 143) and (country ^=405);
quit; run;

proc sql;
   create table alumni as
   select unique cont, id as country
   from maps.world;
quit; run;

proc gproject data=world out=world
   dupok eastlong project=Winkel2;
   id cont country;
run;

options fmtsearch=(sashelp.mapfmts);

data alumni; set alumni;
   length  countryname $20;
   countryname=put(country,glcnsm.);
   alum=.;
run;
options nocenter ps=500;

/*This datalines statement is incredibly long, but I kept it in the
paper to show all the countries*/
data alumni_data;
input alum cont country countryname $ 28-50;
if alum le 25 then alum_range=2;
else if alum le 100 then alum_range=3;
else if alum le 500 then alum_range=4;
else if alum le 1000 then alum_range=5;
else if alum le 100000 then alum_range=6;
if alum eq . then alum_range=1;
datalines;
  .       91        61      Navassa Island
  .       91        72      Puerto Rico
  .       91        78      Virgin Islands U S
  .       91       100      Aruba
 10       91       180      Bahamas
  5       91       195      Bermuda
  .       91       227      Belize
  1       91       231      British Virgin Islan
 429      91       260      Canada
  1       91       268      Cayman Islands
 13       91       295      Costa Rica
  .       91       300      Cuba
  2       91       320      Dominican Republic
  3       91       330      El Salvador
```

You can also use the sql procedure to create datasets. The country does not equal (^=) removes Greenland and Antarctica

The gproject procedure lets you define the overall map style. For a full listing of styles refer to the gproject section in SAS help

Column 1: Number or alumni
Column 2: Continent code
Column 3: Country code
Column 4: Country name

3	91	415	Guatemala
.	91	420	Haiti
3	91	430	Honduras
5	91	487	Jamaica
175	91	595	Mexico
.	91	609	Montserrat
3	91	665	Nicaragua
4	91	710	Panama
.	91	763	Saint Kitts/Nevis
.	91	765	St.Helena
.	91	773	St.Pierre Miquelon
.	91	906	Turks/Caicos Islands
10000	91	926	United States
3	92	142	Anguilla
.	92	149	Antigua/Barbuda
48	92	150	Argentina
3	92	184	Barbados
4	92	205	Bolivia
64	92	220	Brazil
32	92	275	Chile
45	92	285	Colombia
.	92	318	Dominica
11	92	325	Ecuador
.	92	337	Falkland Islands
.	92	355	French Guiana
.	92	406	Grenada
.	92	407	Guadeloupe
1	92	418	Guyana
.	92	591	Martinique
.	92	640	Netherlands Antilles
3	92	715	Paraguay
23	92	720	Peru
.	92	770	St.Lucia
.	92	775	Saint Vincent/Grenad
.	92	840	Suriname
4	92	887	Trinidad And Tobago
4	92	930	Uruguay
29	92	940	Venezuela
1	93	115	Azerbaijan
1	93	120	Albania
1	93	135	Armenia
.	93	140	Andorra
13	93	165	Austria
.	93	185	Bosnia/Herzegovina
44	93	190	Belgium
1	93	211	Belarus
9	93	245	Bulgaria
8	93	310	Czech Republic
17	93	315	Denmark
3	93	331	Estonia
.	93	336	Faroe Islands
6	93	340	Finland
102	93	350	France
3	93	390	Georgia
95	93	394	Germany
124	93	400	Greece
.	93	405	Greenland

.	93	416	Guernsey
.	93	440	Croatia
8	93	445	Hungary
5	93	450	Iceland
18	93	470	Ireland
60	93	480	Italy
.	93	488	Jan Mayen Islands
.	93	495	Jersey
.	93	541	Latvia
.	93	542	Lithuania
.	93	548	Slovakia
.	93	553	Liechtenstein
3	93	570	Luxembourg
.	93	574	Macedonia
.	93	576	Moldova
.	93	588	Man
1	93	590	Malta
2	93	607	Monaco
62	93	630	Netherlands
17	93	685	Norway
2	93	688	Serbia
8	93	730	Poland
95	93	735	Portugal
19	93	755	Romania
.	93	782	San Marino
4	93	789	Slovenia
70	93	830	Spain
.	93	845	Svalbard
19	93	850	Sweden
110	93	855	Switzerland
275	93	925	United Kingdom
6	93	928	Ukraine
1	93	970	Yugoslavia
5	94	125	Algeria
.	94	141	Angola
.	94	187	Bassas Da India
9	94	210	Botswana
1	94	252	Burundi
.	94	257	Cameroon
.	94	264	Cape Verde
.	94	269	Central African Rep.
.	94	273	Chad
1	94	286	Comoros
.	94	290	Congo
.	94	291	Zaire
.	94	311	Benin
.	94	317	Djibouti
.	94	327	Eritrea
.	94	332	Equatorial Guinea
.	94	334	Europa Island
4	94	335	Ethiopia
1	94	388	Gabon
.	94	389	Gambia
8	94	396	Ghana
4	94	417	Guinea
1	94	485	Ivory Coast
.	94	497	Juan De Nova Island

8	94	505	Kenya
1	94	543	Lesotho
.	94	545	Liberia
2	94	550	Libya
.	94	575	Madagascar
.	94	577	Malawi
1	94	585	Mali
.	94	592	Mauritania
2	94	593	Mauritius
.	94	594	Mayotte
11	94	610	Morocco
.	94	615	Mozambique
.	94	667	Niger
13	94	670	Nigeria
.	94	737	Guinea-Bissau
.	94	750	Reunion
5	94	758	Rwanda
.	94	783	Sao Tome/Principe
.	94	787	Senegal
.	94	788	Seychelles
2	94	790	Sierra Leone
.	94	800	Somalia
38	94	801	South Africa
7	94	818	Zimbabwe
.	94	821	Namibia
.	94	831	Western Sahara
.	94	835	Sudan
3	94	847	Swaziland
4	94	865	Tanzania
.	94	883	Togo
.	94	889	Tromelin Island
1	94	890	Tunisia
2	94	910	Uganda
7	94	922	Egypt
.	94	927	Burkina Faso
1	94	990	Zambia
.	95	110	Afghanistan
2	95	181	Bahrain
11	95	182	Bangladesh
1	95	200	Bhutan
.	95	228	British Indian Ocean
4	95	250	Myanmar
7	95	255	Cambodia
16	95	272	Sri Lanka
642	95	280	China
602	95	281	Taiwan
14	95	305	Cyprus
.	95	393	Gaza Strip
.	95	399	Glorioso Islands
315	95	435	Hong Kong
1365	95	455	India
12	95	460	Iran
.	95	465	Iraq
78	95	475	Israel
544	95	490	Japan
8	95	500	Jordan
.	95	510	Kyrgyzstan

.	95	514	Korea,North
918	95	515	Korea,South
15	95	520	Kuwait
11	95	525	Kazakhstan
.	95	530	Laos
14	95	540	Lebanon
2	95	573	Macau
108	95	580	Malaysia
1	95	583	Maldives
1	95	608	Mongolia
13	95	616	Oman
3	95	625	Nepal
79	95	700	Pakistan
.	95	714	Paracel Islands
125	95	747	Qatar
.	95	784	Tajikistan
41	95	785	Saudi Arabia
335	95	795	Singapore
15	95	825	Russia
.	95	833	Spratly Islands
2	95	858	Syria
270	95	875	Thailand
44	95	888	United Arab Emirates
224	95	905	Turkey
.	95	909	Turkmenistan
1	95	931	Uzbekistan
12	95	945	Vietnam
.	95	955	West Bank
.	95	965	Yemen
.	96	60	American Samoa
.	96	62	Jarvis Island
.	96	63	Micronesia
.	96	64	Baker Island
.	96	65	Howland Island
1	96	66	Guam
.	96	67	Johnston Atoll
.	96	68	Kingman Reef
1	96	69	Northern Mariana Isl
.	96	70	Palmyra Atoll
.	96	71	Midway Island
.	96	73	Marshall Islands
.	96	75	Palau
.	96	80	Wake Island
.	96	155	Ashmore/Cartier
139	96	160	Australia
.	96	229	Solomon Islands
1	96	232	Brunei
.	96	282	Clipperton Island
.	96	284	Cocos Islands
.	96	293	Cook Islands
.	96	294	Coral Sea Islands
.	96	338	Fiji
1	96	367	French Polynesia
.	96	369	French Southern Terr
.	96	398	Kiribati
.	96	424	Heard/Mcdonald
167	95	458	Indonesia

```
  .       96      516      Christmas Island
  .       96      621      Nauru
  1       96      645      New Caledonia
  .       96      651      Vanuatu
  22      96      660      New Zealand
  .       96      672      Niue
  .       96      683      Norfolk Island
  1       96      712      Papua New Guinea
  64      95      725      Philippines
  .       96      727      Pitcairn Islands
  .       96      884      Tokelau
  .       96      886      Tonga
  .       96      908      Tuvalu
  .       96      950      Wallis/Futuna Island
  .       96      963      Samoa
  .       97      212      Bouvet Island
  .       397     397      Gibraltar
;
run;

/*This data statement allows you to see the country name and the number
of alumni in that country when viewed in a webpage. If you view the
output in the SAS results viewer it should show you how it works.*/
data alumni_data; set alumni_data;
   length  myhtmlvar $400;
   myhtmlvar='title='||quote( trim(left(countryname))||':
  '||trim(left(alum))||' Alumni');
run;

GOPTIONS DEVICE=html;
ODS LISTING CLOSE;
ODS HTML path=odsout body="&name..htm"
 (title="Carnegie Mellon Alumni by Country") style=d3d;

goptions border;
goptions cback=CXFFFFFF ctext=black;
goptions xpixels=900 ypixels=600;

pattern1 value=solid color=WHITE;
pattern2 value=solid color=CXEAF1DD;
pattern3 value=solid color=CXD7E4BC;
pattern4 value=solid color=CXC2D69A;
pattern5 value=solid color=CX75923C;
pattern6 value=solid color=CX4F6228;

goptions gunit=pct htitle=6.25 ftitle="Calibri/bold" htext=2.7
ftext="Calibri";
 title "Carnegie Mellon Alumni by Country";
 footnote1 j=l "For specific numbers please visit www.cmu.edu/ira and
 click factbook";

 legend1 label=(position=top f="Calibri" h=3 ''
  j=c f="Calibri" h=3 )
  order=(1 2 3 4 5 6)
  shape=bar(.15in,.15in)
```

Hover your mouse over a country on the SAS output to see how this works

Defining order here means you don't have to use the midpoint option in the gmap procedure. Shape allows you to change the dimensions of the legend

```
  value=(justify=left 'None' '1-25' '26-100' '101-500' '501-1000'
  '1000+')
  position=(bottom left)  offset=(7,3)
  across=1 mode=share;

 proc gmap data=alumni_data map=world all;
  id cont country;
  choro alum_range / discrete
   coutline=CX5A5A5A
   legend=legend1
   html=myhtmlvar
   des="" name="&name";
 run; quit;
ODS HTML CLOSE;
ODS LISTING;
```

Additional Tips

- Use one color in shades that range from light to dark. Different colors might make your map stand out, but it may also distract from easily seeing the patterns on your map.
- Consider your audience when you decide whether to use grayscale or color
- If the legend extends beyond the border of the map it will disappear. You may need to adjust the location or size of the legend in order for it to appear properly
- The SAS help feature has information on hundreds of other options that could be useful to you. Linking data points, displaying actual numbers on states, and layering maps are just a few examples of the options available.

Summary

The purpose of this paper was to give an overview of the GMAP procedure and highlight how it can be used to visually display institutional research data. There are many other advanced features of the GMAP procedure that are available, but do not fit within the scope of this paper. However, it should be clear that this procedure can be incredibly useful to institutional researchers using SAS 9.3 base software.

Converting RGB (Red, Green, Blue) to Hexadecimal

Many programs like Microsoft Excel or Microsoft Word use an RGB scale to show color values. However, SAS 9.3 only uses hexadecimal values instead of RGB to represent colors. SAS support has made code available that allows you to change your red, green, and blue color values and see the corresponding hexadecimal value in the log. Here is the code provided by SAS support:

```
data _null_;
/* Input values */
 red=42;
 green=31;
 blue=170;
/* Convert to hex */
hexred=put(red,hex2.);
 hexgreen=put(green,hex2.);
 hexblue=put(blue,hex2.);
/* Create as a SAS RGB color */
```


Change your RGB values here

```
 sasrgb="CX"||hexred||hexgreen||hexblue;
/* Write the new value to the Log */
 put sasrgb;
run;
```

Check the log for the hexadecimal code after submitting the statement.

References

The examples explained paper have been adapted from Robert Allison's work on designing SAS Maps. His website has many useful examples on designing maps and has been a crucial resource in my understanding of SAS Graph and PROC GMAP.
http://robslink.com/SAS/Home.htm

Author Contact

Andrew Merrill
Carnegie Mellon University – Institutional Research and Analysis
5000 Forbes Avenue, UTDC Building
Pittsburgh, PA 15213

(412) 268-2262
amerrill@cmu.edu
www.cmu.edu/ira