Paper 220-2010

KML Macro: Integrating SAS® and Google API and Its Application in Mapping Manitoba's Health Data on Google Earth and Google Map

Mahmoud Azimaee, St. Boniface General Hospital (University of Manitoba), Winnipeg, MB, Canada

ABSTRACT

Google API allows developers to overlay their own customized maps on Google Earth and Google Map. It means that it is possible to colorize a map based on an estimated or known measure. This requires putting the geographical coordinates along with statistical data together in a specific XML file format called KML. This paper introduces KML macro (written by the author) that uses SAS® to quickly and easily create such a file in one step by a SAS user who does not need GIS knowledge or additional software. Also, it will be shown how to create an appropriate coordinates file for this macro in SAS format. This paper also provides a coordinate SAS data file for the province of Manitoba broken down by Regional Health Authority districts. An application of this macro in mapping health research data, results of a research project done and delivered by Manitoba Centre for Health Policy (MCHP) will be presented on Google map and Google Earth.

INTRODUCTION

Google API (Application Programming Interface) allows developers to overlay their own customized maps on Google Earth and Google Map. This capability makes it possible to present statistical data graphically on the dynamic maps of Google Maps or 3-D visualized Google Earth while all the features in these two powerful tools will still work. In order to do this, you will need to put your information on a text file in a series of special XML syntaxes called KML (Keyhole Markup Language). This paper will introduce a new SAS macro that simply creates such a file. To use this macro, you need to provide a SAS data file containing perimeter coordinates for the geographical area that you would like to present your data on. A proper SAS data set for the province of Manitoba in Canada by Regional Health Authority District (RHAD) has been created by the author. Furthermore it will be discussed that how to create such a perimeter coordinates file for any region in the world map. Then, the application of macro will be shown by using a few examples including an actual health research data in Manitoba, and finally the body of the macro will be briefly discussed.

DATA SETS

In order to use KML macro, first you need to construct two SAS data sets: 1) SAS Perimeter Coordinates 2) Rate data. Structure of these two data sets will be discussed as following.

1. SAS PERIMETER COORDINATES DATA

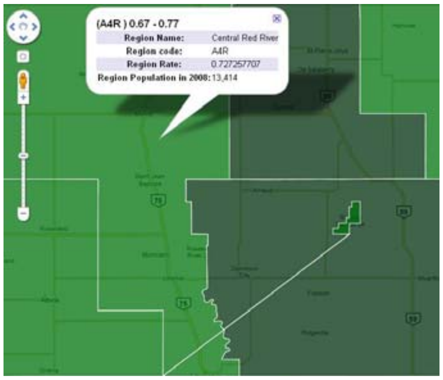
Nowadays, perimeter coordinates in common GIS format such as Shapefile (SHP, SHX, DBF), AutoCAD (DXF), Geography Markup Language (GML), ... are mostly available online through the governmental organizations around the world. In order to use these file types with KML macro, all you need is to convert GIS file of your intended region to a SAS data file once and forever. There are several free GIS software that allow you to convert these type of files to an appropriate text format readable by SAS.

Table 1 shows the structure of this data set which has multiple records per region and can have up to six variables. Four first variables are required and last two are optional. (A reserved variable name is a variable that you can not choose a different name for it)

Google first draws a polygon for each region and then colors it. Polygons will be drawn tracing counter-clockwise over perimeters of each region, so make sure that your data is sorted in a proper order by coordinates in each region. If some regions need to be made by more than one polygon, they must be marked with additional segments (seg). The most popular case of this situation will happen when there is an island which belongs to a region. If a region can be made by only one single polygon, then segment will take the value of 1 for all of the coordinates within that region, otherwise by starting coordinates of the second polygon, segment value must change to 2 and so on for the other polygons. Figures 1-a and 1-b show what would happen if a segment value remain intact while there are more than one polygon in a region.

Table 1 - Variables structure in Perimeter coordinates data set

Variable	Description	Is variable name reserved?	Required/Optional	Type
latitude	geographical latitude	yes	required	numeric
longitude	geographical longitude	yes	required	numeric
rhad	region's short name (code)	no	required	character
seg	region's segments	no	required	numeric
name	region's full name	yes	optional	character
pop	region's population in 2008	yes	optional	numeric


(A4R) 0.67 - 0.77

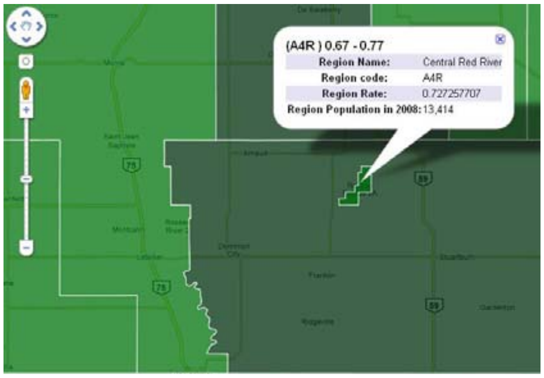
Region Name: Central Red River

Region code: A4R

Region Rate: 0.727257707

Region Population in 2008: 13,414

Figure 1-a. Without changing a segment


(A4R) 0.67 - 0.77

Region Name: Central Red River

Region code: A4R

Region Rate: 0.727257707

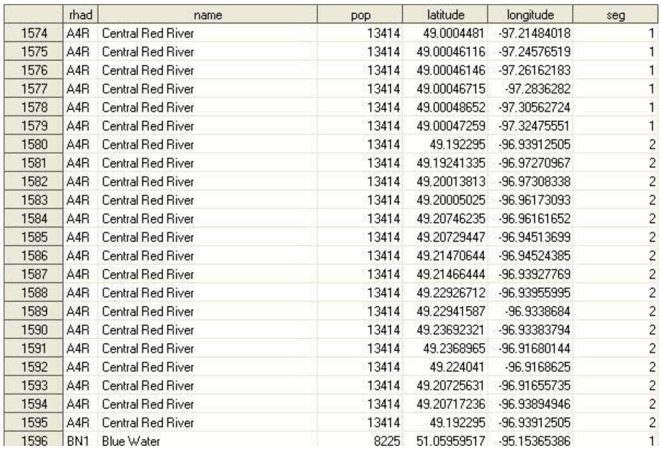
Region Population in 2008: 13,414

Figure 2-b. After defining a new segment

The last two optional variables will appear as additional information in an info-bubble when you click on each region in the map.

SAS Map library can be used with KML macro as well. In order to do this you need to make some modification on these data sets. First of all you must convert SAS coordinates from radians to degrees. Also to make your final kml file in a smaller size you do not need to keep all the coordinates which are in very high resolution format. The last required change would be re-defining sub-regions and segments by using CONCATENATE Function. These steps will be shown in the example section to create coordinates data sets for Canada and the United States from SAS Map library.

A perimeter coordinates data for the province of Manitoba has been created by the author based on the Regional Health Authority Districts (RHAD) which is available through author's email. Figure 2 shows how it looks like:



	rhad	name	pop	latitude	longitude	seg
1574	A4R	Central Red River	13414	49.0004481	-97.21484018	1
1575	A4R	Central Red River	13414	49.00046116	-97.24576519	1
1576	A4R	Central Red River	13414	49.00046146	-97.26162183	1
1577	A4R	Central Red River	13414	49.00046715	-97.2836282	1
1578	A4R	Central Red River	13414	49.00048652	-97.30562724	1
1579	A4R	Central Red River	13414	49.00047259	-97.32475551	1
1580	A4R	Central Red River	13414	49.192295	-96.93912505	2
1581	A4R	Central Red River	13414	49.19241335	-96.97270967	2
1582	A4R	Central Red River	13414	49.20013813	-96.97308338	2
1583	A4R	Central Red River	13414	49.20005025	-96.96173093	2
1584	A4R	Central Red River	13414	49.20746235	-96.96161652	2
1585	A4R	Central Red River	13414	49.20729447	-96.94513699	2
1586	A4R	Central Red River	13414	49.21470644	-96.94524385	2
1587	A4R	Central Red River	13414	49.21466444	-96.93927769	2
1588	A4R	Central Red River	13414	49.22926712	-96.93955995	2
1589	A4R	Central Red River	13414	49.22941587	-96.9338684	2
1590	A4R	Central Red River	13414	49.23692321	-96.93383794	2
1591	A4R	Central Red River	13414	49.2368965	-96.91680144	2
1592	A4R	Central Red River	13414	49.224041	-96.9168625	2
1593	A4R	Central Red River	13414	49.20725631	-96.91655735	2
1594	A4R	Central Red River	13414	49.20717236	-96.93894946	2
1595	A4R	Central Red River	13414	49.192295	-96.93912505	2
1596	BN1	Blue Water	8225	51.05959517	-95.15365386	1

Figure 2. Part of the perimeter coordinates data set for the province of Manitoba.

2) RATE DATA

It is assumed that you have already estimated your desired rate by the same regions in your perimeter data, and saved it in a SAS data set. This data set which is required to be a single record per region must have at least two variables: rhad (with the exact same name, properties and values in your perimeter data, and a numeric variable with the rates for each region. The rate variable can have any name.

CALLING THE MACRO

The KML macro has been written as a "Statement-style macro" by using STMT option which requires the IMPLMAC system option to be in effect. It can be turned on through SAS setting or simply by adding the following command before calling the KML macro:

Code Box 1: Macro Syntax.

```
option
implmac=1;
_kml <option>;
```

STMT option will let the macro to be invoked by its name, followed by a series of options.

MACRO OPTIONS

Default values are listed in the square brackets following the option name:

`rate_file=[]`

Rate data set name: It must be a SAS data set including SAS library name.

Example: `rate_file=project.diabeties`

`rate=[rate]`

Rate variable name: Insert the name of the variable that contains rate values in the rate data set. (Default name is rate)

Example: `rate=immunization`

`region=[rhad]`

Region variable name: Insert the name of the variable that contains regions short names (codes) in the rate and perimeter datasets. (Default name is rhad)

Example: `region=county`

`xy_file=[]`

Perimeter coordinates data set name: It must be a SAS data set including SAS library name.

Example: `xy_file=data.MB_by_districts`

`path=[]`

Full path: following by a file name for the .kml output file.

Example: `path='C:\project\Maps\Immunization.kml'`

`title=['Map Created by SAS KML Macro']`

Map's title: Choose a title for your map and type it within a quotation mark. If you ignore this option, default title (Map Created by SAS KML Macro) will be shown in the final map.

Example: `title='Proportion of Children Born in 1996/97 to 2001/02 With Complete Immunizations at Two Years'`

`opacity=[70]`

Percentage of opacity in the final map: Insert an integer number between 0 (being completely transparent) to 100 (being completely opaque). Recall that you are creating a map to be overlaid on the original Google Map / Earth. The default value of 70% is chosen for this option that relatively makes it possible that geographical signs such as roads, rivers and names remain visible enough.

Example: `opacity=83`

`color=['red']`

Color theme for the map: Three different color themes has been defined for the KML macro: 1) red, 2)green, 3)red2green. In the "red" theme, regions with higher rates will be colored with darker red and lower rates will be colored with brighter red. A similar procedure has been used for the "green" theme. For the "red2green" theme, regions with lower rates turn slowly to green.

Example: `color=’red2green’`

`method=['quintiles']`

Classification method: KML macro can apply four different methods to classify the regions by their rate values: 1) equal, 2) quintiles, 3) k-mean, and 4) sd.

Example: `method='k-mean'`

`showrate=[1]`

Show the rate values: By this option user can turn showing the rate values off or on. By default the exact rate value in each region will be shown in an info-bubble by clicking on the region. If there are concerns about the confidentiality of the rate values, assign value of 0 to this option to prevent showing of the exact values.

CLASSIFICATION METHODS

Equal Intervals: This method divides the range of the rate values into 5 equal classes.

Quintiles Method: This method distributes the rate values into 5 groups that contain an equal number of rates. These five groups are defines by calculating 20th, 40th, 60th, and 80th percentiles.

K-mean Clustering Method: This method classifies observations to 5 groups in which each observation belongs to the class with the nearest mean. A numerical iterative method is used by Proc FASTCLUS. According to procedure’s documentation, *“Each iteration reduces the least squares criterion until convergence is achieved.”*

Standard Deviation Method: SD method finds the mean value and then defines 0.5 standard deviation above and below the mean as the middle class. The two other class breaks will be defined by one standard deviation above and below the mean. The values beyond +/- one standard deviation from the mean, will be classified as the first and last classes.

EXAMPLE 1- MAP OF CANADA BY CENSUS SUB-DIVISIONS

The following code uses SAS map library to create a perimeter coordinates data set for Canada by census sub-divisions which will be compatible with KML macro. It maps a randomly generated rates by *RANUNI()* Function for each sub-division.

Code Box 2: Example 1. Map of Canada by census sub-divisions and random rates.

```
*****************************;
********* Example 1 *********;
*****************************;
libname mymaps 'C:\My Maps\';
data canada;
       set maps.canada3;
       *** Convert coordinates from radian to degree;
       pi = gamma(0.5)**2;
       longitude = x * -(180/pi);
       latitude = y * (180/pi);
       subdiv=province||CDCODE;   * Re-define sub-devisions;
       if longitude=. | latitude=. then delete;
       seg=CDCODE || segment;     * Re-define segments;
       if density < 4 ;           * Only keep data for a moderate resolution;
       run;
*** Add names of sub-divisions;
data mymaps.canada;
       merge canada(in=datain) maps.cancens(in=namein);
       by province cdcode;
       if datain;
       run;
*** Generate a random rate for each sub-division to map as an example;
data mymaps.rates_canada;
       set mymaps.canada;
       by subdiv;
       if first.subdiv;
       rate=ranuni(0)*100;
       keep subdiv rate;
       run;
*** Call KML Macro;
%include 'C:\My Maps\_kml.sas';
option implmac=1 ;
_kml
       rate_file=mymaps.rates_canada
       region=subdiv
       xy_file=mymaps.canada
       path='C:\My Maps\Canada.kml'
       title='Canada by Census Sub-Divisions, Rates are Random'
       opacity=72
       color='red2green'
       method='equal';
run;
```

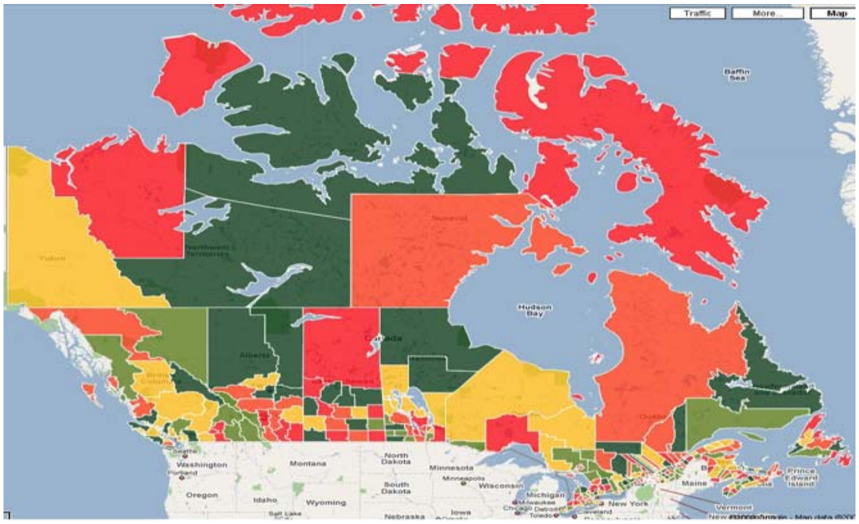


Figure 3- Map of Canada by census sub-divisions and random rates

EXAMPLE 2- MAP OF THE UNITED STATES BY COUNTIES

The following code uses SAS map library to create a perimeter coordinates data set for the United States by counties to be used with KML macro. Rates are the "Average Household Income" which is available in SAS Map library.

Obviously in these two examples if you need to map a specific province or state you can simply restrict the data on that province/state using a WHERE or IF statement on province/state.

Code Box 3: Example 2. Map of The United States by counties.

```
*****************************;
********* Example 2 *********;
*****************************;

libname mymaps 'C:\My Maps\';
data mymaps.USA;

       *** Get names and Average Household Income of the counties;

       merge maps.counties(in=datain)
             maps.usaac(in=namein keep=state county cntyname AHI89);
       by state county;
       if datain;
       rename cntyname=name;
       *** Convert coordinates from radian to degree;
       pi = gamma(0.5)**2;
       longitude = x * -(180/pi);
       latitude = y * (180/pi);
       cnty=state||county;    * Re-define counties;
       if longitude=. | latitude=. then delete;
       seg=County || segment; * Re-define segments;
       if density < 4 ;       * Only keep data for a moderate resolution;
       run;

*** Call KML Macro;

%include 'C:\My Maps\_kml.sas';
option implmac=1 ;

_kml
       rate_file=mymaps.USA
       rate=AHI89
       region=cnty
       xy_file=mymaps.USA
       path='C:\My Maps\USA.kml'
       title='Average Household Income in USA by Counties, '
       opacity=75
       color='green'
run;
```

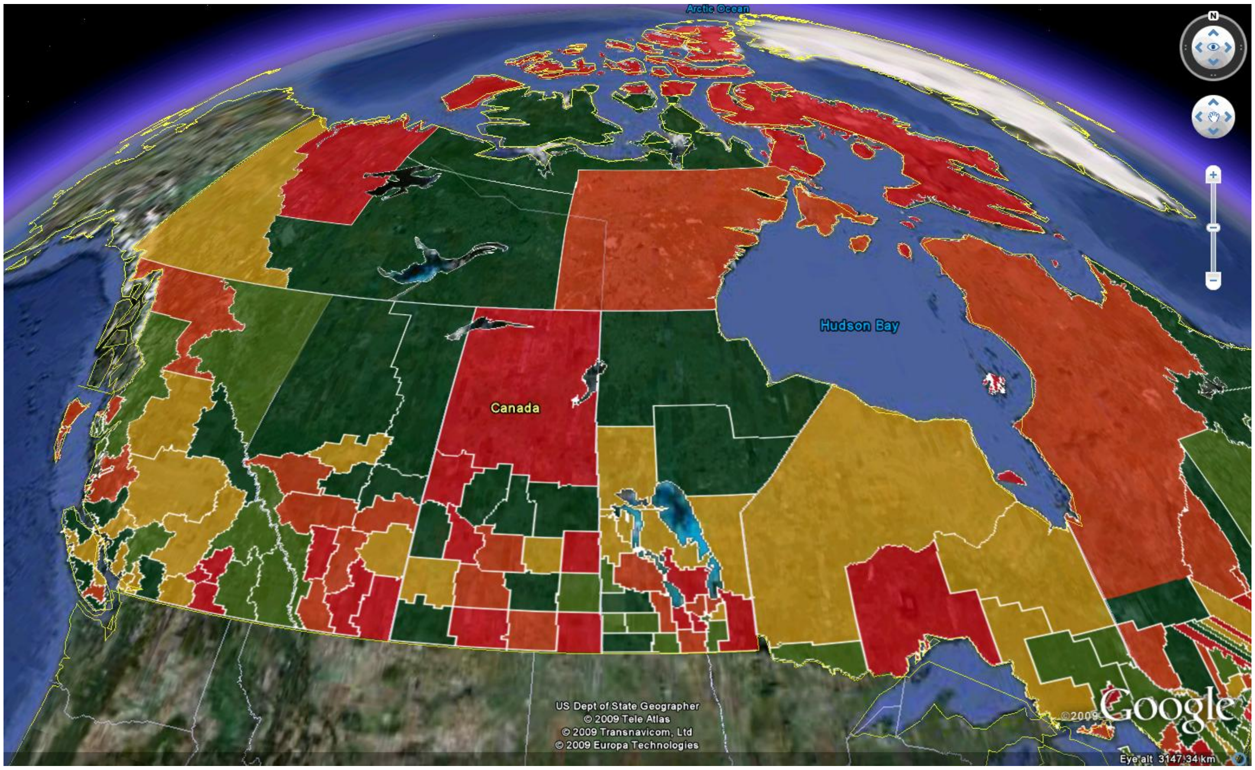


Figure 4- Map of Canada by census sub-divisions and random rates (on Google Earth)

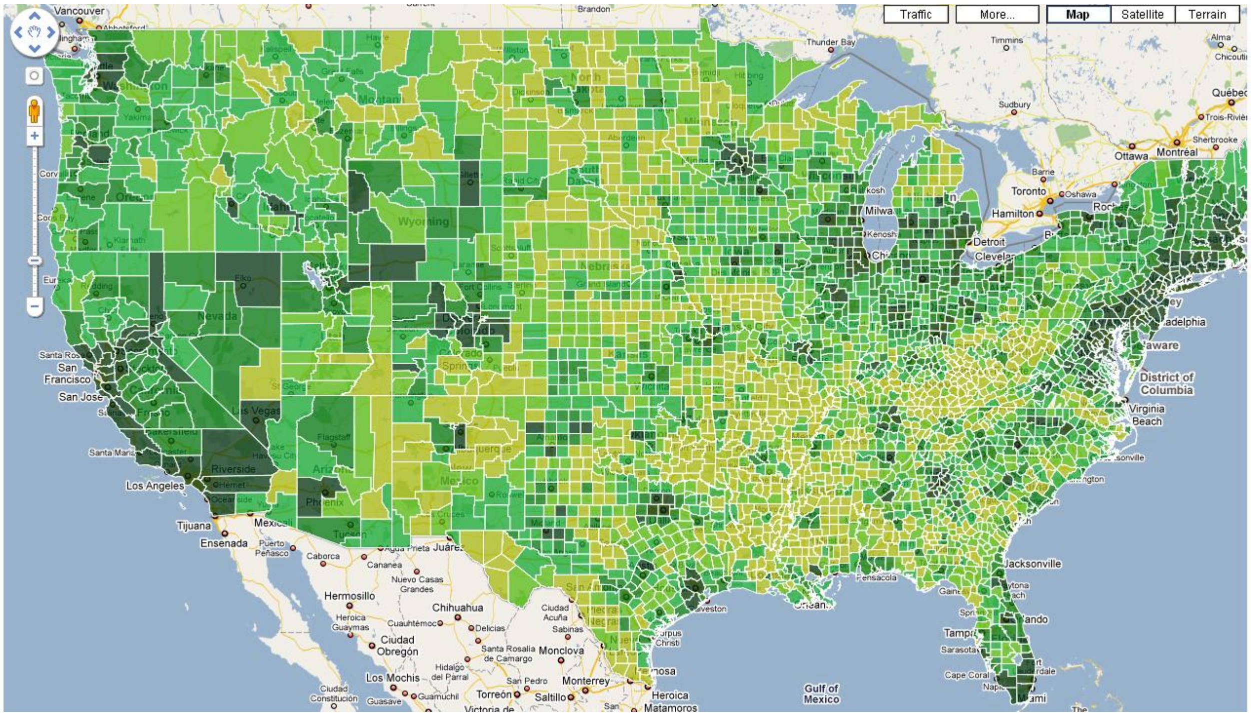
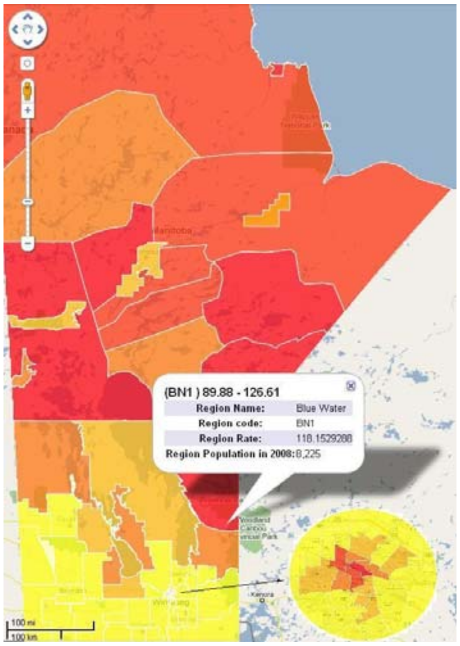


Figure 5- Average Household Income in the USA by counties. (Data from SAS Map Library)

EXAMPLE 3- MAP OF PROVINCE OF MANITOBA BY REGIONAL HEALTH AUTHORITY DISTRICTS

This example uses coordinates data set created by the author to map “Teen Pregnancy Rate in Manitoba from 1996/97-2003/04”. Data comes from a research project delivered and published by “Manitoba Centre for Health policy”. Rates are age-adjusted per 1000 females age 15-19. Regional health authority districts (RHAD) break the province down to 80 individual districts. Coordinates data set has been prepared based on the most recent boundaries of the Manitoba Regional Health Authority.



(BN1) 89.88 - 126.61

Region Name: Blue Water

Region code: BN1

Region Rate: 118.1529288

Region Population in 2008: 8,225

Figure 6- Teen Pregnancy Rate in Manitoba from 1996/97-2003/04 by RHAD’s

```
*****************************;
********* Example 3 *********;
*****************************;

libname mymaps 'C:\My Maps\';


%include 'C:\My Maps\_kml.sas';
option implmac=1 ;

_kml
       rate_file=mymaps.MCHP_Data
       rate=pragnancy
       xy_file=mymaps.MB_by_districts
       path='C:\My
Maps\teen_pragnancy.kml'
       title='Teen Pregnancy Rates
1996/97-2003/04, (Age-adjusted rate of
teen pregnancies per 1,000 females age
15-19)'
       method='equal';
run;
```

Code Box 4: Example 3. Map of Manitoba by RHAD’s

All four classification methods were tested on this data to show the importance of choosing a right method. Knowing that teen pregnancy rate in Manitoba is highly right-skewed; see how you get completely different maps using each method in figure 7.

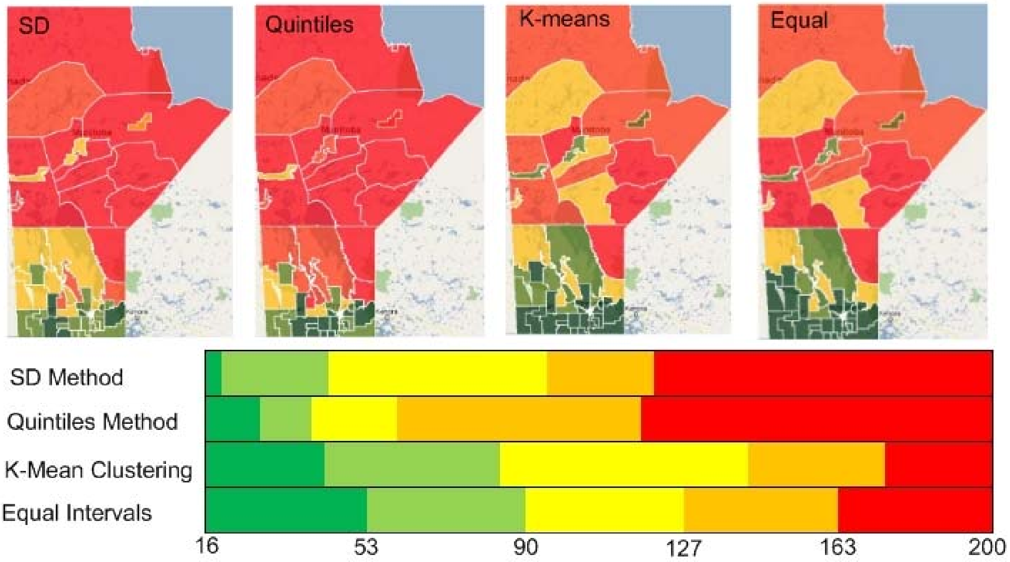


Figure 7- Comparing different classification methods.

HOW TO USE KML FILE?

There are several ways to view a KML map file:

- If you are able to upload kml file to a URL address then you would just need to go to maps.google.com and put that URL on the search box and hit the “Search Maps” button.
- If you have a Google account, then sign in maps.google.com and then go to My Maps>Create New Map>Import and upload your file. (You will be provided with the options of sharing this map with everyone, selected people or keep it private.)
- If you have Google Earth software installed on your computer, you can just click on the KML file and it will be opened by Google Earth and you will have a 3-D view of your map.
- There is also free Google Earth plug-in available online for Firefox and Internet Explorer that make you able to view your KML file 3-D within your browser.

It is strongly recommended to compress your KML file as a zip file. It will make navigation on your map faster and easier. Also Google Map has a limitation for the size of the files. Zipped KML files are known as KMZ file. After compressing the KML file you may change its extension from .zip to .kmz manually, however you do not have to do this.

BODY OF THE MACRO

The full code of the macro is given in Code Box 5. It starts with creating a copy of the coordinate data set sorted by the region. Then it calculates percentiles, mean and standard deviation of the rates over the whole data set. Also macro checks if user has requested using equal method instead of quintiles to replace equal intervals with the calculated percentiles in the previous step. If the user’s choice has been k-mean methods, then macro runs Proc FASTCLUS to find the boundaries and add it to the rate files. Then the rate data set has to be merged with the coordinate data set.

In the next step macro assigns each region to one of the five categories based on the selected classification method which were calculated in the previous step. Also depends on the selected color theme, a proper color code is assigned to each of five categories. Since the final goal is to write all the information in an XML language, colors have to be coded in six digits hexadecimal.

In the final step, macro starts writing all the required information into a text file to create a map in an XML format by using special KML syntaxes. In order to do this job, it opens a text file for writing on the pre-defined path by user. Google API expects color codes in 8 digits arranged in order of *oobbggrr*, while so far we have only 6 digits for each color in order of *bbggrr*. The first 2 digits actually represent the opacity of the color in hexadecimal in the range of 00 to FF (from transparent to opaque). Macro needs to convert opacity which were already entered in percentage to hexadecimal and put in the beginning of each color code.

The rest of the code deals with XML language which will not be discussed in this paper. Interested readers are encouraged to check Google API documentation on the internet for details.

CONCLUSION

This macro integrates two powerful tools with each other and enables researchers to present their research results easily and quickly on Google Map or in 3D-view of Google Earth. All the required coordinate data sets can be easily prepared by converting available data sets in SAS Map Library. It is also possible to create any specific coordinates data set by converting available coordinate files in other GIS formats.

Using KML macro specially is the best solution for web based research reports as the maps created by this macro can be easily embedded to any webpage, while all the zooming and navigating features still can be reserved.

REFRENCES

1. Burlew Michele M. SAS Macro Programming Made Easy. SAS Institute Inc., Cary, NC, 2006.

2. Drukenbrod Josh, Mintz David. Using SAS® and Google Earth to Access and Display Air Pollution Data. SAS Global Forum 2008

3. Manitoba Centre for Health Policy (MCHP), Teen Pregnancy Rate in Manitoba.

4. SAS Institute Inc. SAS® 9.1 Macro Language Reference. SAS Institute Inc. Cary, NC, 2004.

5. Google Institute, Google Maps API Concepts, Online on the internet.

6. Google Institute, Google Earth API Developer's Guide, Online on the internet.

ACKNOWLEDGMENTS

I would like to thank Charles Burchill (Manager, Program and Analysis Systems, and Privacy Officer at the Manitoba Centre for Health Policy) for his great advices, comments and encouragements in writing and testing this macro. Also I would like to thank Dr. Ali Reza Fotouhi, the first person who opened me a new horizon for statistical analysis by teaching me the basic of SAS programming.

CONTACT INFORMATION

Your comments and questions are greatly appreciated and encouraged. Please contact the author at:

Mahmoud Azimaee
University of Manitoba - Faculty of Medicine
St. Boniface General Hospital
Work Phone: (204) 975-7712
E-mail: azimaee@gmail.com
Web: www.dastneveshteha.com

Code Box 5: Macro code.

```
%macro _kml ( rate_file=,
                      rate=rate,
                      region=rhad,
                      seg=seg,
               xy_file=,
                      path=,
              title='Map Created by SAS KML Macro',
                      opacity=70,
                      color='red',
                      method='quintiles',
                      showrate=1
            ) / stmt;
%put ;
%put *************************************************************;
%put ** KML Macro Vesrion 1.1                                   **;
%put ** By: Mahmoud Azimaee  (azimaee@gmail.com)                **;
%put ** University of Manitoba - St. Boniface General Hospital **;
%put ** Winnipeg, Manitoba, Canada                              **;
%put ** Februray 26, 2010                                       **;
%put *************************************************************;
%put;
%put *************************************************************;
%put Running KML Macro...;
%put Rate dataset: &rate_file;
%put Rate Variable: &rate;
%put By variable: &region;
%put Coordinate dataset: &xy_file;
%put Classification Method: &method;
%if showrate=1 %then %put Show the Rate Values on the Map: Yes;
%put The map file will be output in &path;
%put *************************************************************;
%put;

** Make a copy of original coordinates data sorted by regions;
**************************************************************;
proc sort data=&xy_file out=xy;
       by &region;
       run;

** Run the Classification Method based on user choise ;
** Calculate percentiles and mean and SD for overal rates ;
***********************************************************;
proc univariate data=&rate_file noprint;
       var &rate;
       output out=prc  pctlpts=0 20 40 60 80 100 pctlpre=p mean=m std=sd;
       run;

** Merge percentiles file with the coordinates data;
***************************************************;
data prc;
       set prc;
       link=1;
       run;

data rates;
       set &rate_file;
       link=1;
       run;

%if &method ^='k-mean' %then %do;
       data rates;
       merge rates prc;
```

Code Box 5 (continued): Macro code.

```
             by link;
             if &method='equal' then do;
                    p20 = p0 + (p100-p0)/5;
                    p40 = p20 + (p100-p0)/5;
                    p60 = p40 + (p100-p0)/5;
                    p80 = p60 + (p100-p0)/5;
                    end;
             if &method='sd' then do;
                    p20 = m - sd;
                    p40 = m - .5*sd;
                    p60 = m + .5*sd;
                    p80 = m + sd;
                    end;
             drop link;
             run;
%end;

**** Run the K-Mean Classification Method;
*****************************************;
%if &method ='k-mean' %then %do;
       proc fastclus data=&rate_file maxc=5 maxiter=10 out=clusters noprint;
             var &rate;
             run;
       proc means data=clusters nway noprint;
             var &rate;
             class cluster;
             output out=outstat min=min max=max;
             run;
       proc sort data=clusters;
             by cluster;
             run;
       data rates;
             merge clusters outstat (keep=cluster min max);
             by cluster;
             run;
       proc sort data=rates;
             by &region;
             run;
%end;
** Categorize the regions in 5 categories by their rates;
********************************************************;
data xy;
       length legend $60 ;
       merge xy rates;
       by &region;
       %if &method ^='k-mean' %then %do;
             if p0 <= &rate =<p20 then do;
                    if &color='red' then color_code='14F0FF';
                    if &color='green' then color_code='14B4A0';
                    if &color='red2green' then color_code='143C00';
                    legend='(' || &region || ') ' || round(p0, 0.11)  || ' - ' ||
round(p20, 0.11);
                    end;
             if p20 < &rate =<p40 then do;
                    if &color='red' then color_code='14B4FF';
                    if &color='green' then color_code='14B450';
                    if &color='red2green' then color_code='147850';
                    legend='(' || &region || ') ' || round(p20, 0.11)+ .01 || ' -
' || round(p40, 0.11) ;
                    end;
             if p40 < &rate =<p60 then do;
                    if &color='red' then color_code='1478FF';
```

Code Box 5 (continued): Macro code.

```
                    if &color='green' then color_code='3CA014';
                    if &color='red2green' then color_code='14B4FF';
                    legend='(' || &region || ') ' || round(p40, 0.11)+ .01 || ' -
' || round(p60, 0.11) ;
                    end;
             if p60 < &rate =<p80 then do;
                    if &color='red' then color_code='143CFF';
                    if &color='green' then color_code='147800';
                    if &color='red2green' then color_code='143CFF';
                    legend='(' || &region || ') ' || round(p60, 0.11)+ .01 || ' -
' || round(p80, 0.11) ;
                    end;
             if p80 < &rate =<p100 then do;
                    if &color='red' then color_code='1400FF';
                    if &color='green' then color_code='143C00';
                    if &color='red2green' then color_code='1400FF';
                    legend='(' || &region || ') ' || round(p80, 0.11)+ .01 || ' -
' || round(p100, 0.11) ;
                    end;
             if &rate=. then do;
                           color_code='FFFFFF';
                           legend='(' || &region || ') ' || 'Unknown/Unreportable';
                           end;
             drop p0 p20 p40 p60 p80 p100 m sd;
       %end;
       %if &method ='k-mean' %then %do;
             if cluster=1 then do;
                    if &color='red' then color_code='14F0FF';
                    if &color='green' then color_code='14B4A0';
                    if &color='red2green' then color_code='143C00';
                    end;
             if cluster=2 then do;
                    if &color='red' then color_code='14B4FF';
                    if &color='green' then color_code='14B450';
                    if &color='red2green' then color_code='147850';
                    end;
             if cluster=3 then do;
                    if &color='red' then color_code='1478FF';
                    if &color='green' then color_code='3CA014';
                    if &color='red2green' then color_code='14B4FF';
                    end;
             if cluster=4 then do;
                    if &color='red' then color_code='143CFF';
                    if &color='green' then color_code='147800';
                    if &color='red2green' then color_code='143CFF';
                    end;
             if cluster=5 then do;
                    if &color='red' then color_code='1400FF';
                    if &color='green' then color_code='143C00';
                    if &color='red2green' then color_code='1400FF';
                    end;
             legend='(' || &region || ') ' || round(min, 0.11)  || ' - ' ||
round(max, 0.11);
             if &rate=. then do;
                           color_code='FFFFFF';
                           legend='(' || &region || ') ' || 'Unknown/Unreportable';
                           end;
             drop cluster distance min max;
       %end;
       rhad_seg = &region || &seg;
       run;
```

Code Box 5 (continued): Macro code.

```
proc sort data=xy;
        by rhad_seg;
       run;

** Create KML file;
******************;
data _null_ ;
       filename kml &path ;
       file kml;
       set xy end=last;
       by rhad_seg;

       ** Convert opacity value from percentage to Hexadecimal;
       *******************************************************;
       length aa $ 2;
       opc=&opacity;
       if opc=0 then opc=opc + .0001;
       opacity16=((opc/100)*256) - 1;
       aa=substr(put(opacity16, hex.),7,2);
       *******************************************************;

       if _n_=1 then do;
              put '<?xml version="1.0" encoding="UTF-8"?>';
              put ' <kml xmlns="http://earth.google.com/kml/2.0">';
              put '  <Document>';
              map='   <name>' || &title || '</name>';
              put map;
              put '  <Schema>';
              put '
<SimpleField><name>name</name><type>string</type></SimpleField>';
              put '
<SimpleField><name>rhad</name><type>string</type></SimpleField>';
              put '
<SimpleField><name>rate</name><type>double</type></SimpleField>';
              put '
<SimpleField><name>pop</name><type>double</type></SimpleField>';
              put '  </Schema>';
              put '   <LookAt>';
              put '    <longitude>-95.98556</longitude>';
              put '    <latitude>55.49306</latitude>';
              put '    <range>2308295</range>';
              put '   </LookAt>';
              end;
              if first.rhad_seg then do;
                     put '   <Placemark>';
                     regioname = '    <name>' || legend || '</name>';
                     put regioname;
                     put '    <Style>';
                     put '     <LineStyle>';
                     put '      <color>FFFFFFFF</color>';
                     put '      <width>1</width>';
                     put '     </LineStyle>';
                     put '     <PolyStyle>';
                     clr='      <color>' || aa || color_code || '</color>';
                     put clr;
                     put '     </PolyStyle>';
                     put '     <IconStyle> </IconStyle>';
                     put '    <balloonStyle> </balloonStyle>';
                     put '     <LabelStyle> </LabelStyle>';
                     put '    </Style>';
                     put '<description>';
```

Code Box 5 (continued): Macro code.

```
                    put ' <![CDATA[';
                    put '  <center>';
                    put '   <table>';
                    atr = "    <tr bgcolor='#E3E3F3'><th>Region Name: </th><td>"
|| name || "</td></tr>";
                    put atr;
             atr = "    <tr bgcolor=''><th>Region code: </th><td>" || &region ||
"</td></tr>";
                    put atr;
             if &showrate then atr = "    <tr bgcolor='#E3E3F3'><th>Region Rate:
</th><td>" || &rate || "</td></tr>";
                           else atr = "    <tr bgcolor='#E3E3F3'><th>Region Rate:
</th><td>Not Allowed by Map creator</td></tr>";
                    put atr;
                    popform=put(pop, comma30.);
             atr = "    <tr bgcolor=''><th>Region Population in 2008: </th><td>"
|| popform || "</td></tr>";
                    put atr;
                    put '  </table>';
             put ' </center>';
                    put ' ]]>';
                    put '</description>';
                    put '<Snippet></Snippet>';
                    put '    <Polygon>';
                    put '     <tessellate>1</tessellate>';
                    put '      <outerBoundaryIs>';
                    put '       <LinearRing>';
                    put '        <coordinates>';
             end;
             coordinate='          ' || longitude || ',' || latitude || ',0 ' ;
             put coordinate;
             if last.rhad_seg then do;
                    put '        </coordinates>';
                    put '       </LinearRing>';
                    put '      </outerBoundaryIs>';
                    put '     </Polygon>';
                    put '   </Placemark>';
             end;
       if last then do;
             put'  </Document>';
             put ' </kml>';
             end;
       run;

%mend _kml;
```