Paper 2418-2018

A Macro that Creates U.S Census Tracts Keyhole Markup Language Files for Google Map Use

Ting Sa, Cincinnati Children’s Hospital Medical Center

ABSTRACT

This paper introduces a macro that can generate the Keyhole Markup Language (KML) files for U.S. census tracts. The generated KML files can be used directly by Google Maps to add customized census tracts layers with user-defined colors and transparencies. When someone clicks on the census tracts layers in Google Maps, customized information is shown. To use the macro, the user needs to prepare only a simple SAS® input data set and download the related KML files from the U.S. census Bureau. The paper includes all the SAS code for the macro and provides examples that show you how to use the macro as well as how to display the KML files in Google Maps.

INTRODUCTION

KML file is a file that can be used to put different layers onto the google map, like a point, a line or a polygon area. Also inside the KML file, you can define the styles of the layers, like changing the color and transparency of the layers, adding information to the layers etc. The U.S Census Bureau provides the census tracts KML file for each state. However, it is one file for the whole state, therefore, if the user only wants to select certain census tracts, or the user wants to customize the census tracts with special background color, transparency or customized information, the user can not directly use the KML file from the U.S Census Bureau. If the user is not familiar with the KML language, it will be hard to create a customized KML file. However, using the macro in this paper, the user just need to download the KML files from the U.S Census Bureau and create a simple SAS input data set and the macro will automatically create the customized census tracts KML files for the user. The Google Map can import this KML file and put the census tract layers on it. Figure 1 shows the customized Colorado census tracts displaying on the Google map. Figure 2 is the screenshot by further zooming in this map so that user can see the different transparencies among the census tracts.

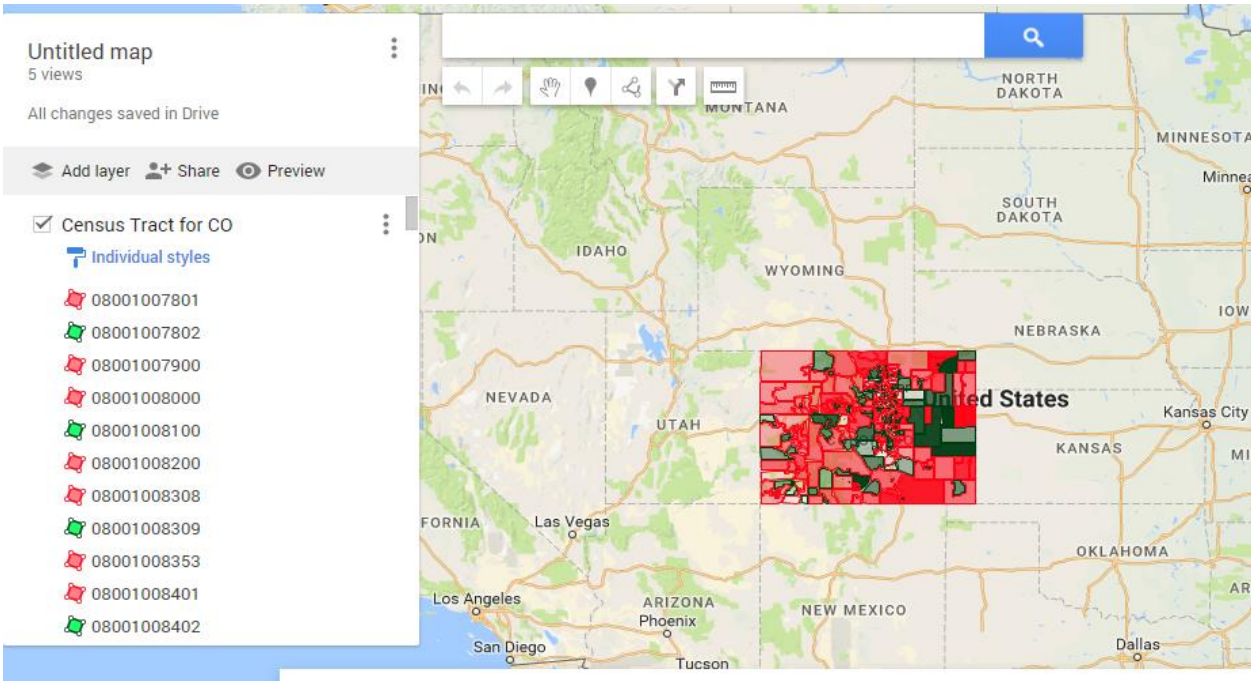
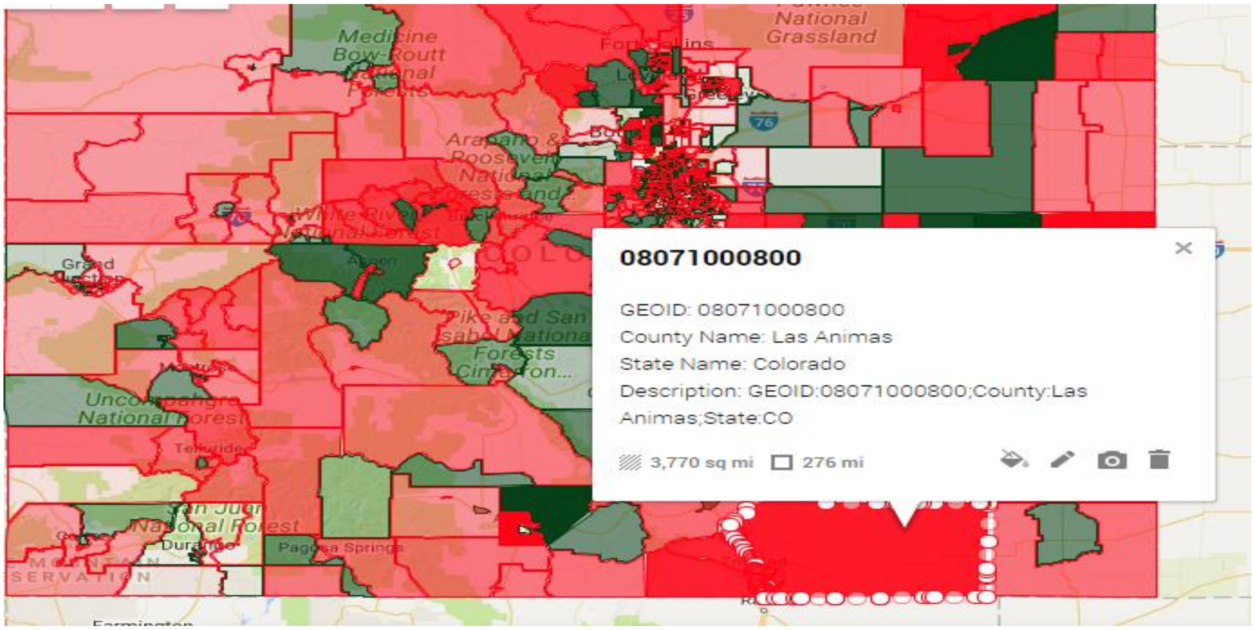


Figure 1. The Customized Census Tracts for the State of Colorado



08071000800

GEOID: 08071000800
County Name: Las Animas
State Name: Colorado
Description: GEOID:08071000800;County:Las Animas;State:CO

3,770 sq mi 276 mi

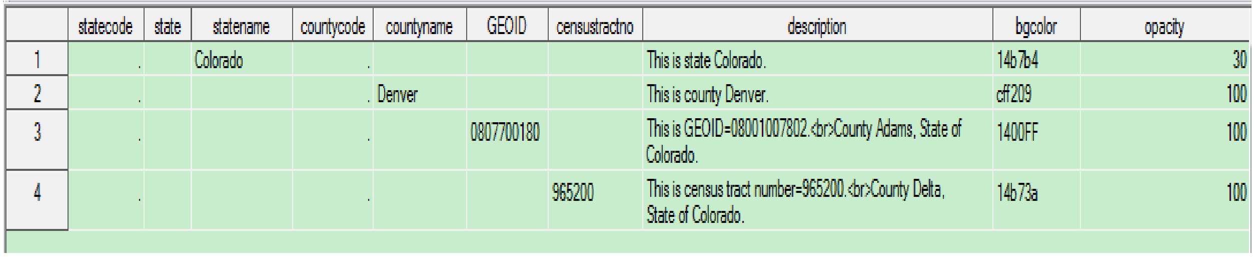
Medicine Bow-Routt National Forests
National Grassland
Fort Collins
Arapaho & Roosevelt National Forests and
White River National Forest
Grand Junction
Aspen
Pike and San Isabel National Forests Cimarron...
Montrose
Uncompahgre National Forest
Telluride
San Juan National Forest
Durango
Pagosa Springs
MOUNTAIN RESERVATION
Farmington

Figure 2. Different Transparency on the Layer

In this paper, you will get all the information you need to create the customized map shown in Figure 1 and Figure 2, including all the SAS codes for the macro. Besides I will show you how to add the KML file to the Google map and how to download the KML data file from the U.S. Census Bureau.

THE INPUT DATA SET FOR THE MACRO

Figure 3 shows the structure of the input data set that can be used by the macro. The column "statecode" contains the state FIPS codes. The "state" column contains the state abbreviations. The "statename" column contains the state name information. The column "countycode" contains the county FIPS codes. The "countyname" column contains the county name information. The column "GEOID" contains the GEOID code. The "censustractno" column contains the census tract number. The "description" column contains the text information for the layer, you can include the html tags into the text information. The "bgcolor" column contains the background color for the layer, you can search online to find the 6 digits hex color code for the background color. The "opacity" column contains the transparency percentage for each layer, the value range for this column is an integer number from 0 to 100.



	statecode	state	statename	countycode	countyname	GEOID	censustractno	description	bgcolor	opacity
1	.		Colorado	.				This is state Colorado.	14b7b4	30
2	.			.	Denver			This is county Denver.	cff209	100
3	.			.		0807700180		This is GEOID=08001007802. County Adams, State of Colorado.	1400FF	100
4	.			.			965200	This is census tract number=965200. County Delta, State of Colorado.	14b73a	100

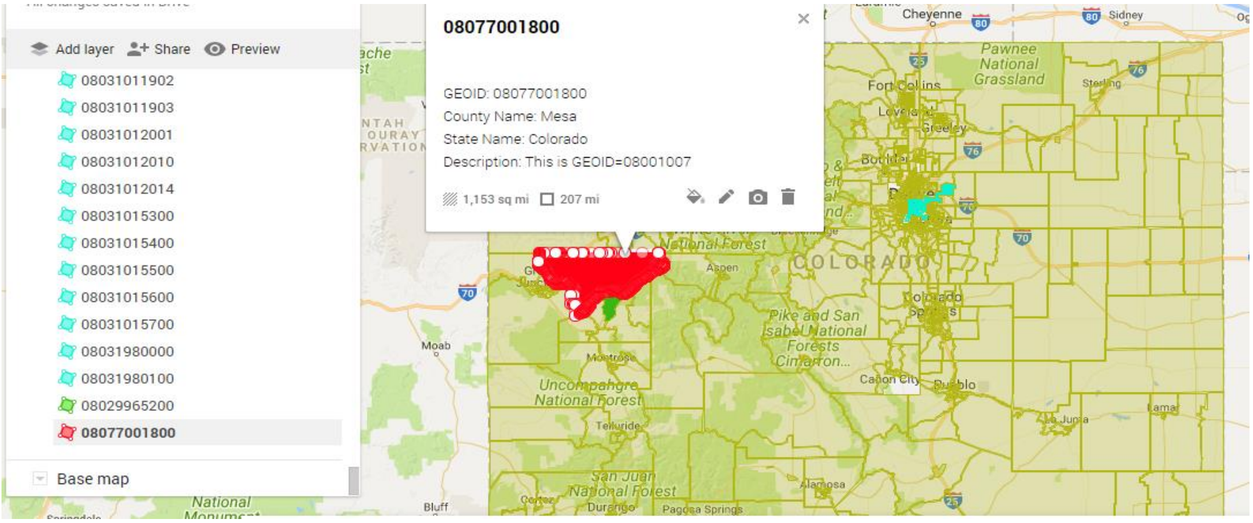
Figure 3. A Sample Input Data Set for the Macro

As you can see from the sample data set, you don't need to provide information for all the 3 columns "statecode", "state" and "statename". As long as you provide at least one value for these 3 columns, the macro will know the state you want to use. The same rule applies to the column "countycode" and "countyname", "GEOID" and "censustractno", as long as at least one of them has value, the macro will get the information. For the character values in those columns, they are case insensitive. Also if the user wants to include all the census tracts for this state, the user only needs to provide the state info, the values for the other columns can be missing. If the user wants to include all the census tracts for a county, the user only needs to provide the county information, the values for the other columns can be missing.

Below are the SAS codes I’ve used to create the sample input data set in Figure 3. In this input data set, it asks the macro to generate all the census tract layers for Colorado state, then all the census tracts for the “Denver” county and the census tracts for GEOID=08077001800 and census tract number=965200.

```
data sampledata;
statecode=.;
length state $2. statename $15.;
state="";statename="Colorado";
countycode=.;
length countyname $25.;
countyname="";
length GEOID $11. censustractno $6. description $250 bgcolor $6.;
GEOID="";censustractno="";description="This is state Colorado.";
bgcolor="14b7b4";opacity=30;
output;
statecode=.;state="";statename="";countycode=.;countyname="Denver";
GEOID="";censustractno="";description="This is county Denver.";
bgcolor="cff209";opacity=100;
output;
statecode=.;state="";statename="";countycode=.;countyname="";
GEOID="08077001800";censustractno="";
description="This is GEOID=08001007802.<br>County Adams, State of
Colorado.";
bgcolor="1400FF";opacity=100;
output;
statecode=.;state="";statename="";countycode=.;countyname="";
GEOID="";censustractno="965200";
description="This is census tract number=965200.<br>County Delta, State of
Colorado.";
bgcolor="14b73a";opacity=100;
output;
run;
```

Figure 4 shows you the census tracts map for this sample input data.



08031011902
08031011903
08031012001
08031012010
08031012014
08031015300
08031015400
08031015500
08031015600
08031015700
08031980000
08031980100
08029965200
08077001800

08077001800

GEOID: 08077001800
County Name: Mesa
State Name: Colorado
Description: This is GEOID=08001007

1,153 sq mi 207 mi

Figure 4. The Census Tracts Map for the Sample Input Data

THE CREATECENSUSTRACTKML MACRO

You can find all the macro codes at the end of the paper. Immediately below is the structure of the macro:

```
%macro CREATECENSUSTRACTKML(kmlinputfile=,libnm=,datanm=,outputkml=);
```

- The “kmlinputfile” is used to indicate the location of the input KML file that the user has downloaded from the U.S. census bureau.

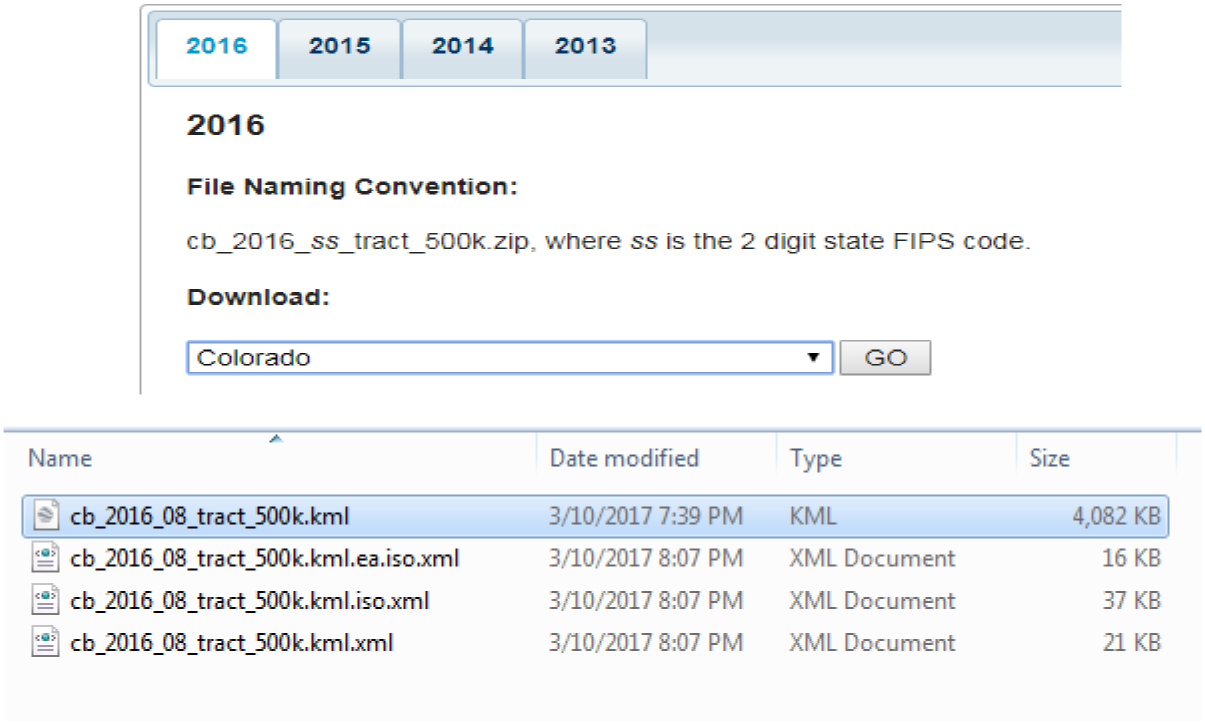
- The “libnm” is used to indicate the library name for the input dataset.
- The “datanm” is used to indicate the input SAS dataset names.
- The “outputkml” is used to indicate the location to save the output KML file.

Below is an example showing you how to call the macro, the result KML file will be saved into “C:\kml\test.kml” file. The “cb_2016_08_tract_500k.kml” is the kml file that you’ve downloaded from the U.S. Census Bureau for the state of Colorado.

```
%getCensusTractsByState(kmlinputfile=%str(C:\kml\cb_2016_08_tract_500k.kml),
libnm=work,datanm=sampledata,outputkml=%str(C:\kml\test.kml));
```

DOWNLOAD THE KML FILE FROM THE U.S CENSUS BUREAU

You can download the census tract kml file for each state through the website of the U.S. Census Bureau. This is the web site address: https://www.census.gov/geo/maps-data/data/kml/kml_tracts.html.On the webpage, just select the state you want to download and a zip file will be automatically downloaded. Once you’ve got the zip file, upzip the file and you will see 4 files are included. For the macro in this paper, it needs to use the file whose file type is “kml”. See the screenshots in Figure 5 for more details:



2016

File Naming Convention:

cb_2016_*ss*_tract_500k.zip, where *ss* is the 2 digit state FIPS code.

Download:

Colorado

Name	Date modified	Type	Size
cb_2016_08_tract_500k.kml	3/10/2017 7:39 PM	KML	4,082 KB
cb_2016_08_tract_500k.kml.ea.iso.xml	3/10/2017 8:07 PM	XML Document	16 KB
cb_2016_08_tract_500k.kml.iso.xml	3/10/2017 8:07 PM	XML Document	37 KB
cb_2016_08_tract_500k.kml.xml	3/10/2017 8:07 PM	XML Document	21 KB

Figure 5. The Downloaded Census Tract KML File from the U.S. Census Bureau

ADD KML FILE TO THE GOOGLE MAP

You can use the “Google My Maps” application to import your KML file to the Google map. Below are the steps to import the KML file:

1. Go to website https://www.google.com/mymaps

2. Use your gmail account to log in, if you don’t have one or want to share the maps with others, you can create a new gmail account.

3. Click the “+ CREATE A NEW MAP” on the left top of the page, see the screenshot in Figure 6.

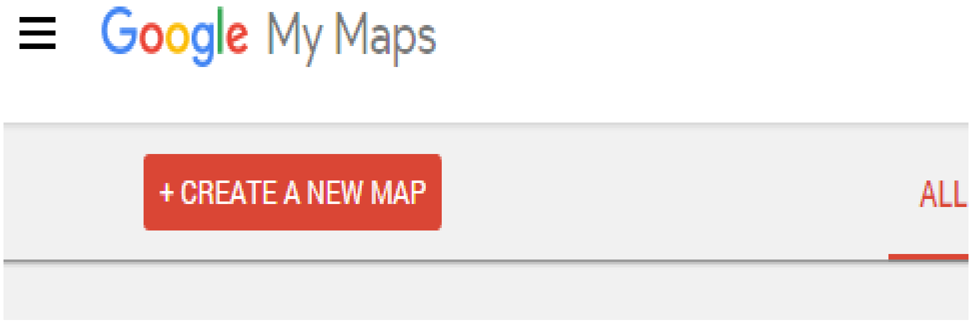
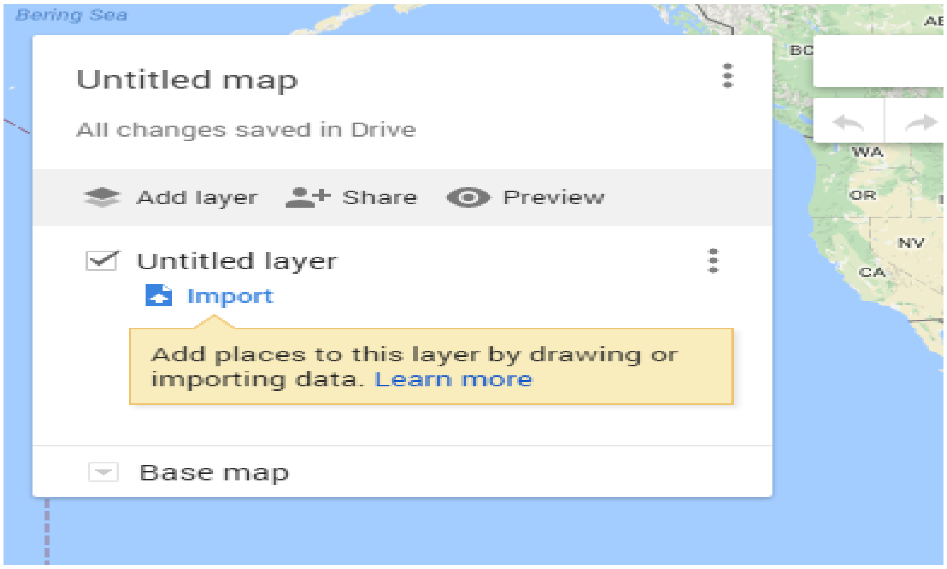


Figure 6. The Screenshot for the Main Page of the “Google My Maps”

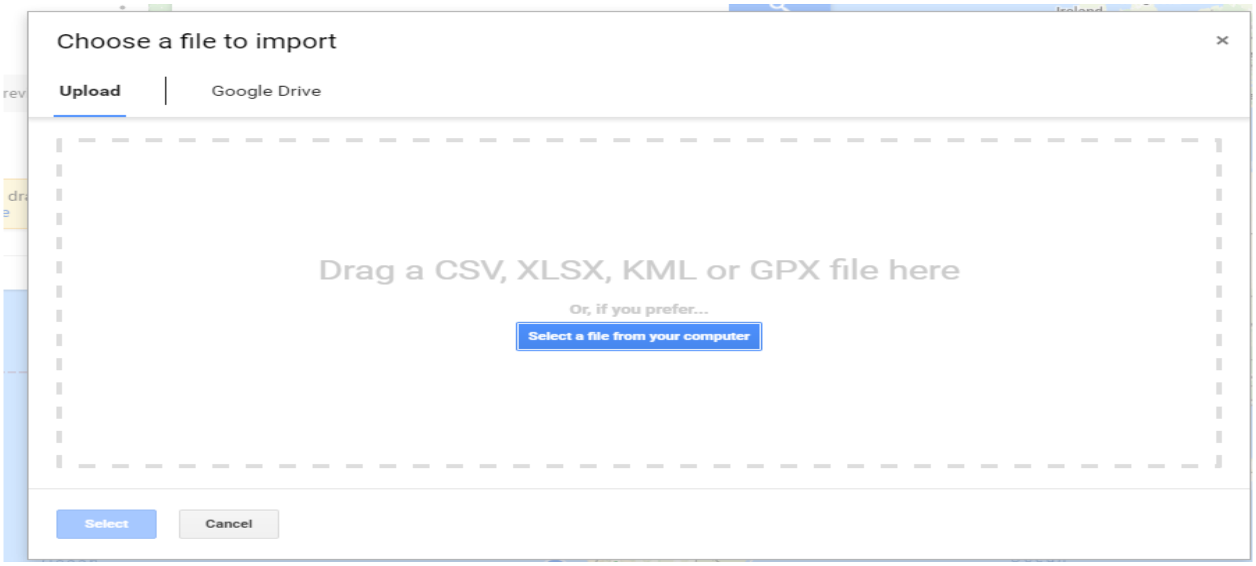
4. Then it will link to a new Google map web page. On this page, click the “Import” link. See the screenshot in Figure 7.



Add places to this layer by drawing or importing data. Learn more

Figure 7. The Screenshot for the “Import” Link

5. Then you will see a box pop out, see the screenshot in Figure 8. You can just drag your KML file from your computer to this box area, the KML file will be imported to the Google map. See the screenshot in Figure 7. You can import more than one KML file to the Google map by click the “Add layer” link. You can also check or uncheck the layer checkbox to show or hide the layer.



Choose a file to import

×
Upload
Google Drive
Drag a CSV, XLSX, KML or GPX file here
Or, if you prefer...
Select a file from your computer
Select
Cancel

Figure 8. The Screenshot for the Box that Imports your KML File

There are many good benefits of using the “Google My Maps” application. One thing is that you can save the map file and share with other people. People can view your interactive maps online without installing any other software. You can learn more about how to use “Google My Maps” by doing the Google search.

THE CREATESTATECOUNTYKML MACRO CODES

Presented below are the SAS codes for the createStateCountyKML macro.

```
%macro CREATECENSUSTRACTKML(kmlinputfile=,libnm=,datanm=,outputkml=);
*read the whole kml file into the SAS data set censustract_kml;
data WORK.censustract_kml;
%let _EFIERR_ = 0;
infile "&kmlinputfile." delimiter='09'x MISSOVER
DSD lrecl=32767 firstobs=2 ;
informat lineval $30651. ;
format lineval $30651. ;
input
lineval $
;
if _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection macro
variable */
run;

data censustract_kml;
set censustract_kml;
rowno=_n_;
run;

*get the starting row number information for each Placemark tag;
data placemarkstart;
set censustract_kml;
where index(lineval,'<Placemark')>0;
matchno=_n_;
rename rowno=placemark_startno;
run;
```

```
*get the ending row number information for each Placemark tag;
data placemarkend;
set censustract_kml;
where index(lineval,"</Placemark>")>0;
matchno=_n_;
rename rowno=placemark_endno;
run;

*get the GEOID informaton;
data GEOID;
set censustract_kml;
where index(lineval,'<SimpleData name="GEOID">')>0;
length GEOID $11.;
GEOID=compress(scan(scan(lineval,2,">"),1,"<"));
statecode=input(substr(GEOID,1,2),best.);
countycode=input(substr(GEOID,3,3),best.);
length censustractno $6.;
censustractno=compress(substr(GEOID,6));
matchno=_n_;
keep GEOID--matchno;
run;

*merge the GEOID information with the Placemark tag information;
data GEOID;
merge GEOID placemarkstart placemarkend ;
by matchno;
drop lineval;
run;
proc sort data=GEOID;by statecode countycode;run;

*get the state, county information from SASHELP.ZIPCODE;
proc sql;
create table statecountyinfo as
select distinct STATE as statecode,STATECODE as state,STATENAME as
statename,
COUNTY as countycode,COUNTYNM as countyname
from SASHELP.ZIPCODE
order by statecode,countycode;
quit;

data GEOID;
merge GEOID(in=a) statecountyinfo;
by statecode countycode;
if a;
run;

data coordinates;
set censustract_kml;
where index(lineval,'<coordinates')>0 and
index(lineval,'</coordinates>')>0;
coordinate=compress(lineval);
run;

proc sql;
create table GEOID_with_coordinates as
select g.*,coordinate
from geoid as g,coordinates as c
```

```
where placemark_startno<=c.rowno<=placemark_endno
order by matchno;
quit;

proc sql;
*if user only provides state level info, then just
choose all the state's census tracts;
create table statelevel as
select g.*,bgcolor,opacity,description
from &libnm..&datanm as i,Geoid_with_coordinates as g
where cmiss(i.statecode,i.state,i.statename)<3 and
cmiss(i.countycode,i.countyname,i.GEOID,i.censustractno)=4 and
(i.statecode=g.statecode or lowcase(i.statename)=lowcase(g.statename) or
lowcase(i.state)=lowcase(g.state));

*if user only provides state and county level info, then just
choose all the county's census tracts;
create table countylevel as
select g.*,bgcolor,opacity,description
from &libnm..&datanm as i,Geoid_with_coordinates as g
where cmiss(i.countycode,i.countyname)<2 and
cmiss(i.GEOID,i.censustractno)=2 and
(i.countycode=g.countycode or
(lowcase(i.countyname)=lowcase(g.countyname)));

*if user only census tract info, then choose that census tract;
create table censustractlevel as
select g.*,bgcolor,opacity,description
from &libnm..&datanm as i,Geoid_with_coordinates as g
where cmiss(i.GEOID,i.censustractno)<2 and
i.GEOID=g.GEOID or i.censustractno=g.censustractno;
quit;

data all;
set statelevel(in=a) countylevel(in=b) censustractlevel(in=c);
if a then no=1;
else if b then no=2;
else if c then no=3;
run;
proc sort data=all;by no geoid;run;

data _null_;
set all;
if _n_=1 then call symput('statename',state);
run;

data _null_ ;
filename kml "&outputkml.";
file kml;
set all end=last;
by no geoid;
** Convert opacity value from percentage to Hexadecimal;
*******************************************************;
length aa $ 2;
if opacity=0 then opacity=opacity + .0001;
opacity16=((opacity/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>';
put " <name>Census Tract for &statename. </name>";
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.geoid then do;
put ' <Placemark>';
regioname = ' <name>' || geoid || '</name>';
put regioname;
put ' <Style>';
put ' <LineStyle>';
put ' <color>FFFFFFFF</color>';
put ' <width>1</width>';
put ' </LineStyle>';
put ' <PolyStyle>';
clr=' <color>' || aa || bgcolor || '</color>';
put clr;
put ' </PolyStyle>';
put ' <IconStyle> </IconStyle>';
put ' <balloonStyle> </balloonStyle>';
put ' <LabelStyle> </LabelStyle>';
put ' </Style>';
put '<description>';
put ' <![CDATA[';
put ' <center>';
put ' <table>';
atr = " <tr bgcolor='#E3E3F3'><th>GEOID: </th><td>"|| geoid ||
"</td></tr>";
put atr;
atr = " <tr bgcolor=''><th>County Name: </th><td>" || countyname
||"</td></tr>";
put atr;
atr = " <tr bgcolor='#E3E3F3'><th>State Name:</th><td>" || statename ||
"</td></tr>";
put atr;
atr = " <tr bgcolor=''><th>Description: </th><td>" || description ||
"</td></tr>";
put atr;
put ' </table>';
put ' </center>';
put ' ]]>';
put '</description>';
put '<Snippet></Snippet>';
end;
put ' <Polygon>';
put ' <tessellate>1</tessellate>';
put ' <outerBoundaryIs>';
put ' <LinearRing>';
put coordinate;
put ' </LinearRing>';
put ' </outerBoundaryIs>';
put ' </Polygon>';
if last.geoid then do;
put ' </Placemark>';
end;
if last then do;
put' </Document>';
put ' </kml>';
end;
run;
%mend;
```

CONCLUSION

The macro presented in this paper can be used as a helpful tool to create the KML file for any U.S. census tract. Once you understand this macro, you can further extend its function to create other KML files for certain areas.

REFERENCES

1. Azimaee,Mahmoud. April 2010. “KML Macro: Integrating SAS® and Google API and Its Application in Mapping Manitoba's Health Data on Google Earth and Google Map”. Proceedings of the SAS Global 2010 Conference, Seattle, WA. Available at

   http://support.sas.com/resources/papers/proceedings10/220-2010.pdf

2. Ting Sa. April 2017. “A Macro that can Create U.S State and U.S County KML Files for Google Map Use”. Proceedings of the SAS Global 2017 Conference, Orlando, FL. Available at

   http://support.sas.com/resources/papers/proceedings17/1132-2017.pdf

ACKNOWLEDGMENTS

The author wishes to thank the Division of Biostatistics and Epidemiology at Cincinnati Children's Hospital Medical Center for its support.

CONTACT INFORMATION

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

Ting Sa
Division of Biostatistics and Epidemiology, Cincinnati Children’s Hospital Medical Center
5136363674
Ting.Sa@CCHMC.ORG