SAS® GLOBAL FORUM 2018

USERS PROGRAM

SESSION 2885

# Plot Your Custom Regions on SAS Visual Analytics Geo Maps

Jitendra N. Pandey

April 8 – 11 | Denver, CO

#SASGF

# Plot Your Custom Regions on SAS® Visual Analytics Geo Maps

Jitendra N. Pandey

Electrolux Home Products Inc., Charlotte, NC

## ABSTRACT


- SAS® Visual Analytics Geo Region Map supports area plot of Country and State geographies out of the box.
- With the use of shape files available on US Census website and some updates to your Visual Analytics configuration, it can support the area plot of US Regions, Divisions, Counties and ZIP Code Tabulation Areas (or ZCTA) as well.
- Often times you are asked if you could plot your organization's custom regions on a map. It might be the sales regions or any other geographic regions of interest, defined in terms of ZIP, County, State, Division or US Census Region.
- Let us take an example of our favorite organization Orion Star. If Orion Star is a national organization, it is likely that its sales regions would be defined in Regions, Divisions or States geographies. If Orion Star was a regional organization, it would define its sales regions in Counties and if it was a local organization, it would probably define its sales regions in Counties or ZIP codes.


## INTRODUCTION TO SHAPE FILES AND GREMOVE PROCEDURE

- In order to plot your custom regions, derived from US Regions, Division, States, Counties and ZCTAs, you need the boundary coordinates of each constituting geography and then you need to eliminate all internal coordinates that form common internal boundaries. SAS' MAPSGFK library contains the geographic datasets that store the boundary coordinates of US states and counties. If you want to plot Regions, Divisions and ZCTAs, you can download corresponding shape files from US Census website and process them using MAPIMPORT and GREDUCE procedures.
- Once you have the boundary coordinates of all the constituting geographies in a map data set, you need to process this to create a data set that comprises of only the external boundaries of your custom regions.
- SAS/GRAPH® procedure GREMOVE comes in handy and does exactly that for you. GREMOVE procedure takes a map data set as input and produces an output data set that has the external boundary coordinates of your custom regions. This output data set can be further processed to update ATTRLOOKUP and CENTLOOKUP datasets located in your VALIB library to plot your custom regions on SAS® Visual Analytics Geo Region Map.
- Couple of examples of custom region definition are given below:

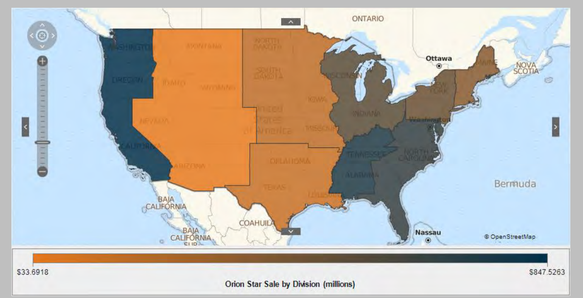
| Custom Region Name | Custom Region Code | Division Name | Division Code |
|---|---|---|---|
| Orion Star East | OD-1 | New England | 1 |
| Orion Star East | OD-1 | Middle Atlantic | 2 |
| Orion Star East | OD-1 | South Atlantic | 5 |
| Orion Star East Central | OD-2 | East North Central | 3 |
| Orion Star East Central | OD-2 | East South Central | 6 |
| Orion Star Central | OD-3 | West North Central | 4 |
| Orion Star Central | OD-3 | West South Central | 7 |
| Orion Star West | OD-4 | Mountain | 8 |
| Orion Star West | OD-4 | Pacific | 9 |

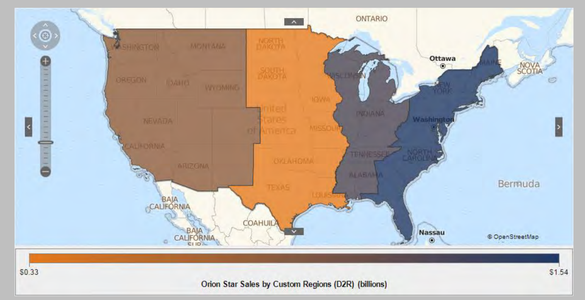
| Custom Region Name | Custom Region Code | County Name | County Code |
|---|---|---|---|
| Orion Star North | OC-1 | Rowan | 37159 |
| Orion Star North | OC-1 | Iredell | 37097 |
| Orion Star North | OC-1 | Catawba | 37035 |
| Orion Star East | OC-2 | Cabarrus | 37025 |
| Orion Star East | OC-2 | Stanly | 37167 |
| Orion Star Central | OC-3 | Mecklenburg | 37119 |
| Orion Star South | OC-4 | Union | 37179 |
| Orion Star South | OC-4 | Anson | 37007 |
| Orion Star West | OC-5 | Lincoln | 37109 |
| Orion Star West | OC-5 | Gaston | 37071 |

## CUSTOM REGION MAPPTING EXAMPLES

Orion Star National Inc. Divisions Sales plotted on SAS Visual Analytics Geo Map (Left, Standard): This map is plotted by importing the US Census Divisions shape file and processing and loading into ATTRLOOKUP and CENTLOOKUP datasets.

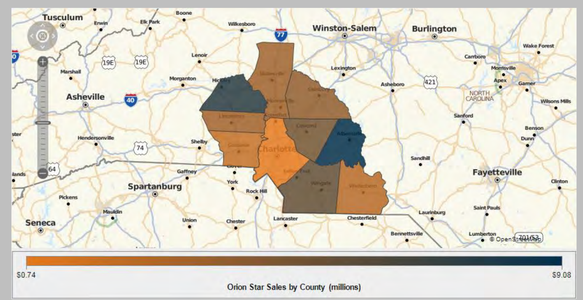
Orion Star National Inc. Custom Regions, defined in terms of US Census Divisions, Sales plotted on SAS Visual Analytics Geo Region Map (Right, Custom): This map is plotted from the output of GREMOVE procedure that used US Census Divisions shape file as input, and processed and loaded the output into ATTRLOOKUP and CENTLOOKUP datasets. Second input used here is the US Census Division to Custom Region (D2R) mapping table.

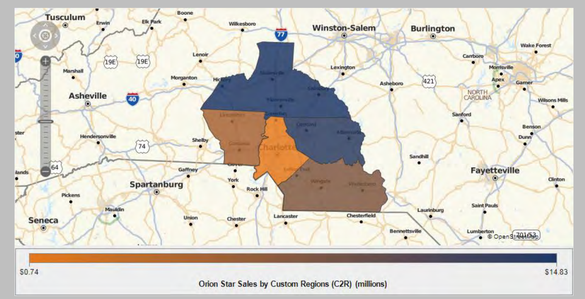



Orion Star Regional Inc. County Sales plotted on SAS Visual Analytics Geo Map (Left, Standard): This map is plotted by importing the US Census County shape file and processing and loading into ATTRLOOKUP and CENTLOOKUP datasets.

Orion Star Regional Inc. Custom Regions, defined in terms of US Counties, Sales plotted on a SAS Visual Analytics Geo Region Map (Right, Custom): This map is plotted from the output of GREMOVE procedure that used US Census County shape file as input, and processed and loaded the output into ATTRLOOKUP and CENTLOOKUP datasets. Second input used here is the US County to Custom Region (C2R) mapping table.



# Plot Your Custom Regions on SAS® Visual Analytics Geo Maps

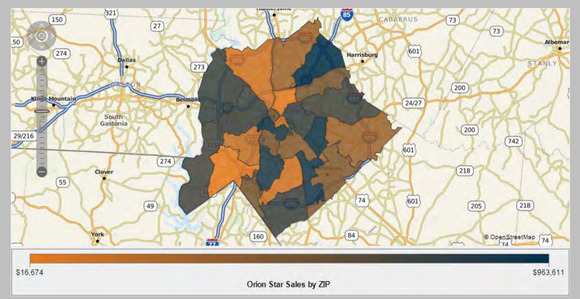
Jitendra N. Pandey

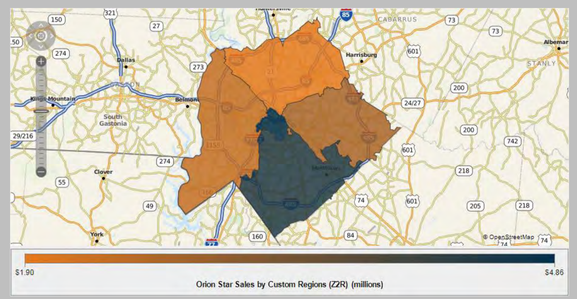
Electrolux Home Products Inc., Charlotte, NC

## CUSTOM REGION MAPPTING EXAMPLES

Orion Star Local Inc. ZIP Sales plotted on a SAS Visual Analytics Geo Map (Left, Standard): This map is plotted by importing the US Census ZCTA shape file and processing and loading into ATTRLOOKUP and CENTLOOKUP datasets.

Orion Star Local Inc. Custom Regions, defined in terms of ZCTA, Sales plotted on SAS Visual Analytics Geo Region Map (Right, Custom): This map is plotted from the output of GREMOVE procedure that used US Census ZCTA shape file as input, and processed and loaded the output into ATTRLOOKUP and CENTLOOKUP datasets. Second input used here is the ZCTA to Custom Region (Z2R) mapping table.




US Census website makes available many more shape files in addition to Region, Division, State, County and ZCTA. More granular shape files such as Census Blocks are also available for a download. This gives us more flexibility to derive a precise maps of our custom regions.

## CONCLUSIONS

As long as your organization's custom regions can be derived from any standard geographic shapes, you can easily derive the map of your custom regions using GREMOVE procedure and update your Visual Analytics configuration to plot them on SAS® Visual Analytics Geo Region Maps.

## REFERENCES

Hall, Angela M. 2016, "Creating Custom Map Regions in SAS® Visual Analytics," Proceedings of the SAS Global Forum 2016 Conference. Cary, NC. SAS Institute Inc. Available at:
https://sasglobalforum2016.lanyonevents.com/connect/fileDownload/session/79B7F9262FE3FC180EAD672D847A6B12/SAS3460_Hall-3460-2016.pdf

# SAS® GLOBAL FORUM 2018

April 8 – 11 | Denver, CO
Colorado Convention Center

#SASGF

Paper 2885-2018

# Plot Your Custom Regions on SAS® Visual Analytics Geo Maps

Jitendra N. Pandey, Electrolux Home Products Inc., Charlotte, NC

## ABSTRACT

SAS® Visual Analytics geo map supports country and state geographies out of the box. It can be programmed to plot counties, cities, and ZIP code tabulation areas as well. However, often times we are asked if we could create an area plot an organization's custom regions on SAS® Visual Analytics Geo Region Map. It might be the sales regions or any other nonstandard geographic regions of interest, defined in terms of standard geographic areas: Region, Division, State, County, ZIP Code Tabulation Area (or ZCTA) or any other standard geographic area defined by US Census. This paper outlines how you can plot such custom regions on a SAS® Visual Analytics Geo Region Map.

## INTRODUCTION

SAS® Visual Analytics Geo Region Map reads the data set ATTRLOOKUP located in DATA\VALIB folder of your SAS Visual Analytics compute server context (often D:\SAS\Config\LevN\SASAppVA\Data\VALIB) to determine if the given geography is a valid candidate for a Geo Region Map (exists with LEVEL=1) and using the Map ID value it looks up to the data set CENTLOOKUP and this data set takes it to the final data set (in MAPSGFK library for states and countries) that stores the ordered list of all the boundary coordinates to plot the map. As mentioned ear

If you want to plot more granular geographies, your SAS deployment might not have the map data sets of those geographies. However, you can download a vast variety of shape files from US Census website. You can import those shape files using MAPIMPORT procedure and further process the output using GREDUCE procedure to add the density to your map data sets and filter the data sets for a lower density and faster performance. Further, you can make some updates to your Visual Analytics configuration to plot these maps downloaded from US Census website. This functionality was presented in SAS Global Forum 2016 by Angela M. Hall. 'References' section of this document provides a link to that paper.

Now, the question is how you could plot the custom regions of your organization on SAS Visual Analytics Geo Region Map that may be different from standard geographies. The answer is, as long as the custom regions of your organization can be defined in terms of standard geographies, you can plot them on Geo Region Map without using any third party software.

Let us take an example of our favorite organization Orion Star. If Orion Star is a national organization, it is likely that its custom regions would be defined in terms of US Census Regions, Divisions or States geographies; least likely they might be defined in terms of Counties. If Orion Star were a regional organization, it would define its custom regions in terms of Counties and if it were a local organization, it would probably define its sales regions in more granular geographies such as ZIP codes or even Census Block. Some example of Orion Star's sales region could look like:

| Custom Region Name | Custom Region Code | Division Name | Division Code |
|---|---|---|---|
| Orion Star East | OD-1 | New England | 1 |
| Orion Star East | OD-1 | Middle Atlantic | 2 |
| Orion Star East | OD-1 | South Atlantic | 5 |
| Orion Star East Central | OD-2 | East North Central | 3 |
| Orion Star East Central | OD-2 | East South Central | 6 |
| Orion Star Central | OD-3 | West North Central | 4 |
| Orion Star Central | OD-3 | West South Central | 7 |
| Orion Star West | OD-4 | Mountain | 8 |
| Orion Star West | OD-4 | Pacific | 9 |

**Table 1. Custom Regions of Orion Star National Inc. defined in US Divisions**

| Custom Region Name | Custom Region Code | County Name | County Code |
|---|---|---|---|
| Orion Star North | OC-1 | Rowan | 37159 |
| Orion Star North | OC-1 | Iredell | 37097 |
| Orion Star North | OC-1 | Catawba | 37035 |
| Orion Star East | OC-2 | Cabarrus | 37025 |
| Orion Star East | OC-2 | Stanly | 37167 |
| Orion Star Central | OC-3 | Mecklenburg | 37119 |
| Orion Star South | OC-4 | Union | 37179 |
| Orion Star South | OC-4 | Anson | 37007 |
| Orion Star West | OC-5 | Lincoln | 37109 |
| Orion Star West | OC-5 | Gaston | 37071 |

**Table 2. Sales Regions of Orion Star Regional Inc. defined in US Counties**

| Custom Region Name | Custom Region Code | ZCTA5CE10 Name | ZCTA5CE10 Code |
|---|---|---|---|
| Orion Star North | OZ-1 | 28216 | 28216 |
| Orion Star North | OZ-1 | 28269 | 28269 |
| Orion Star North | OZ-1 | 28262 | 28262 |
| Orion Star North | OZ-1 | 28213 | 28213 |
| Orion Star East | OZ-2 | 28215 | 28215 |
| Orion Star East | OZ-2 | 28205 | 28205 |
| Orion Star East | OZ-2 | 28212 | 28212 |
| Orion Star East | OZ-2 | 28227 | 28227 |
| Orion Star South | OZ-3 | 28209 | 28209 |
| Orion Star South | OZ-3 | 28211 | 28211 |
| Orion Star South | OZ-3 | 28210 | 28210 |
| Orion Star South | OZ-3 | 28226 | 28226 |
| Orion Star South | OZ-3 | 28270 | 28270 |
| Orion Star South | OZ-3 | 28105 | 28105 |
| Orion Star South | OZ-3 | 28277 | 28277 |
| Orion Star South | OZ-3 | 28134 | 28134 |
| Orion Star West | OZ-4 | 28214 | 28214 |
| Orion Star West | OZ-4 | 28208 | 28208 |
| Orion Star West | OZ-4 | 28217 | 28217 |
| Orion Star West | OZ-4 | 28273 | 28273 |
| Orion Star West | OZ-4 | 28278 | 28278 |
| Orion Star Central | OZ-5 | 28202 | 28202 |
| Orion Star Central | OZ-5 | 28280 | 28280 |
| Orion Star Central | OZ-5 | 28282 | 28282 |
| Orion Star Central | OZ-5 | 28244 | 28244 |
| Orion Star Central | OZ-5 | 28204 | 28204 |
| Orion Star Central | OZ-5 | 28203 | 28203 |
| Orion Star Central | OZ-5 | 28207 | 28207 |

**Table 3. Sales Regions of Orion Star Local Inc. defined in ZCTA**

## A NOTE ABOUT CUSTOM REGION CODE PREFIXES

The custom region codes provided in above three examples use prefixes OD, OC and OZ. These prefixes are not mandatory. However, since all three examples use the same custom region codes (1, 2, 3, 4, 5,) the prefixes were required to maintain the uniqueness of ATTRLOOKUP and CENTLOOKUP data sets at ID level when all of these custom region codes exist in these two data sets together. And this is possible that an organization might have more than one custom region definitions, one for sales and other for distribution network, for example. Another case to support the use of prefixes is: If you want to import

both, the Counties and ZCTAs in ATTRLOOKUP and CENTLOOK data sets, you must use a prefixes with County Codes and ZIP Codes else there will be conflicts.

## PLOTTING YOUR CUSTOM REGIONS

In order to plot your custom regions, derived from US Regions, Division, States, Counties and ZCTAs, you need the boundary coordinates of each constituting geography and then you need to eliminate all internal coordinates that form common internal boundaries. SAS' MAPSGFK library contains the geographic data sets that store the boundary coordinates of US states and counties (in addition to other nations and their states/provinces) and many of the mapping needs can be fulfilled by these data sets. However, if you want to map Regions, Divisions and ZIP Code Tabulation Areas or ZCTAs, you need to download corresponding shape files from US Census website.

## SHAPES AVAILABLE ON US CENSUS WEBSITE

US Census makes available various shape file for free download and has been publishing the same every year since the year 2000. The following table provides the link to these shape files:

| Shape | ID Variable | Link |
|---|---|---|
| US Regions | GEOID | https://www.census.gov/geo/maps-data/data/cbf/cbf_region.html |
| US Divisions | GEOID | https://www.census.gov/geo/maps-data/data/cbf/cbf_division.html |
| US States | GEOID | https://www.census.gov/geo/maps-data/data/cbf/cbf_state.html |
| US Counties | GEOID | https://www.census.gov/geo/maps-data/data/cbf/cbf_counties.html |
| US ZCTAs | ZCTA5CE10 | https://www.census.gov/geo/maps-data/data/cbf/cbf_zcta.html |

**Table 4. Some Shape Files available on US Census Website**

Above table presents only few standard geographies. You may find more standard geographies' shape files on US Census website to suit your needs. You can download the shape files for a geographies as small as a US Census Block. Once you have the boundary coordinates of all the constituting geographies, you need to process this data to create a data set that comprises of only the external boundaries of your custom regions.

## INTRODUCTION TO PROC GREMOVE

SAS/GRAPH® procedure GREMOVE comes in handy and does exactly that for you. GREMOVE procedure takes a map data set as input and produces an output data set that has the external boundary coordinates of your custom regions. This output data set is then used by GMAP procedure to plot your custom regions. The same output can be further processed to update ATTRLOOKUP and CENTLOOKUP data sets located in your VALIB library to plot the custom regions on SAS Visual Analytics Geo Region Map. The syntax of GREMOVE procedure is given below for your reference:

```
proc gremove data=libref.input_map out=libref.custom_region_map;
     by <Region Variable>;
     id <ID Var>;
run;
```

Where:

Region Variable is Custom Region Name or Code of your custom regions, and

ID Var is the unit of standard geographies coming from the input map data set.

## SUMMARY OF THE STEPS

Now you know how you can obtain the coordinates of the external boundary of your custom region. Here is a summary of the steps required to plot your regions on a SAS Visual Analytics Geo Region Map:

1. Determine the unit of constituting geographies of your custom region.
2. If the map data is available with your SAS deployment, proceed to Step 4.
3. Download the shape file of constituting geographies from US Census website.
4. Read the standard to custom region mapping table.
5. Get the map data of all constituting geographies by joining the data set from step 2 or 3 and 4.
6. Use the data set created in step 5 as an input to PROC GREMOVE.
7. Create unique entry in ATTRLOOKUP data set with LEVEL=0.
8. Create a centroid entry in CENTLOOKUP data set corresponding to LEVEL=0.
9. Use GREMOVE output to create a map data set of your custom regions in the supported format.
10. Use GREMOVE output to enter the name of the data set created in step 9 and centroids of your custom regions in CENTLOOKUP data set.

### A NOTE ABOUT THE SERVER RESTART

If you store the map data set of the custom regions in a data library other than VALIB, such as MAPCUSTM, you must restart your web application server. The author of this paper used VALIB to store the map data set of custom regions and he observed that a restart was not required. SAS Visual Analytics Designer was able to read the custom regions' map data sets without a restart.

## THE RESULTS

So, we performed all the steps mentioned in the previous section. Now let us plot all of the three standard to custom region examples provided earlier in this paper, on SAS Visual Analytics Designer and access them in Visual Analytics Viewer and see how the custom regions look compared with the standard geographic regions.

## DIVISION TO REGION (D2R)

The following map is plotted by importing the US Census Divisions shape file and processing and loading it into ATTRLOOKUP and CENTLOOKUP datasets.

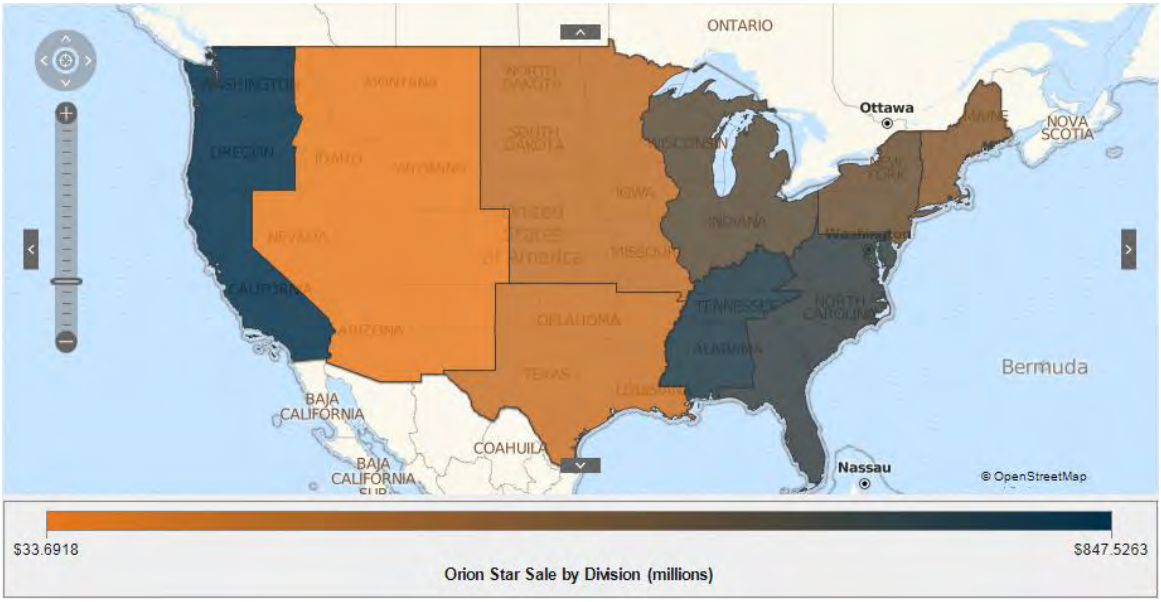



**Figure 1. Orion Star National Inc. Divisions Sales plotted on SAS Visual Analytics Geo Region Map**

The following map is plotted from the output of GREMOVE procedure that used US Census Divisions shape file as input, and processed and loaded the output into ATTRLOOKUP and CENTLOOKUP datasets. Second input used here is the US Census Division to Custom Region (D2R) mapping table.

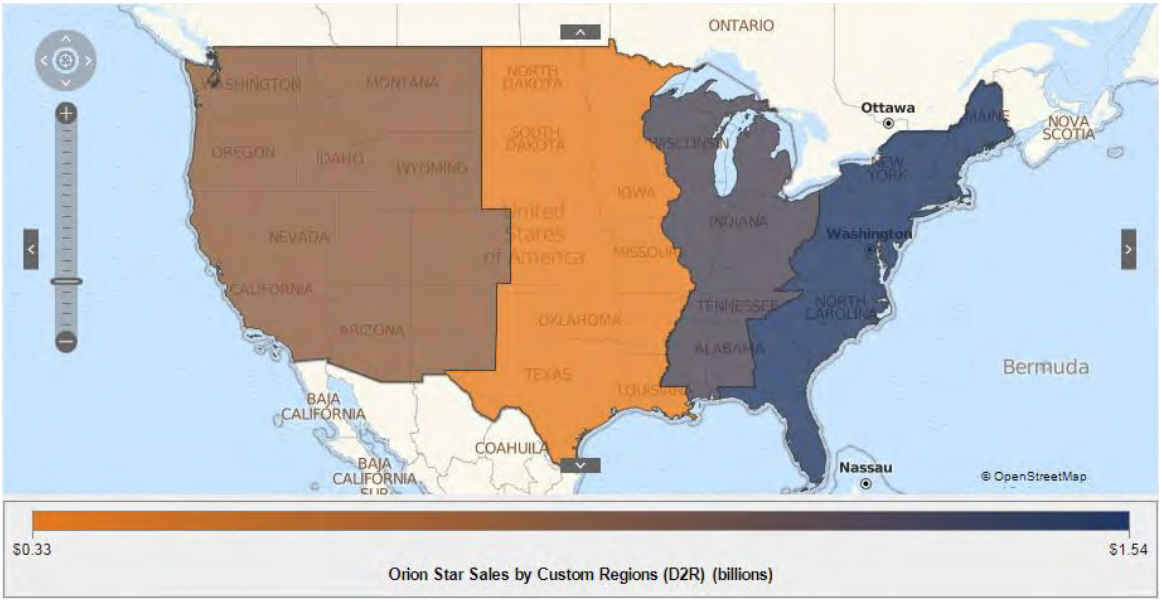



**Figure 2. Orion Star National Inc. Custom Regions Sales plotted on SAS Visual Analytics Geo Region Map**

## COUNTY TO REGION (C2R)

The following map is plotted by importing the US Census County shape file and processing and loading it into ATTRLOOKUP and CENTLOOKUP datasets.

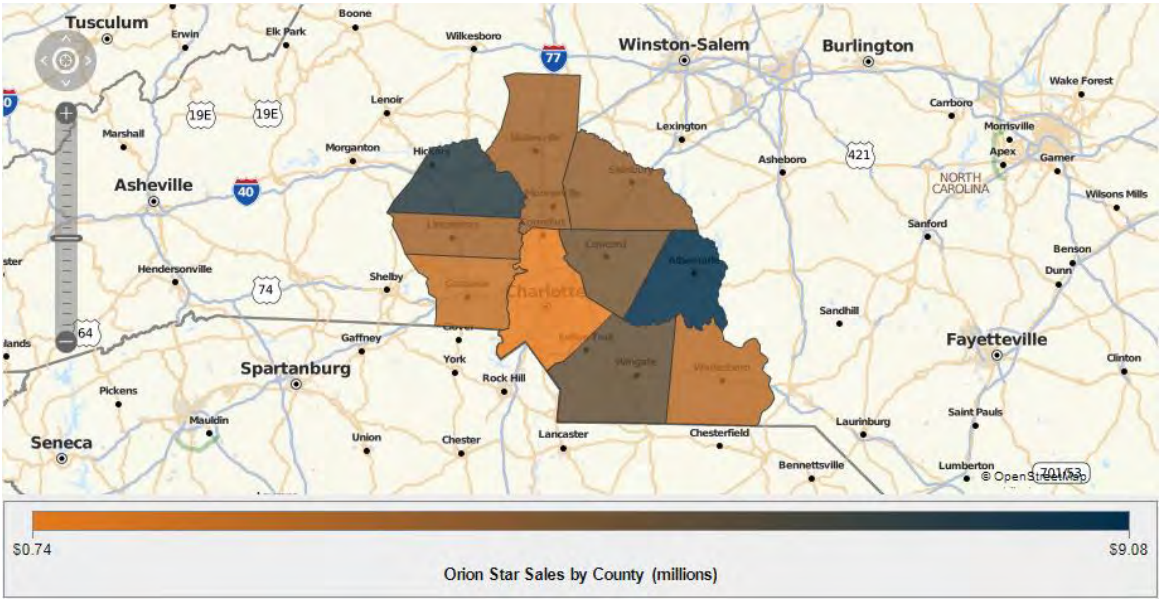



**Figure 3. Orion Star Regional Inc. County Sales plotted on SAS Visual Analytics Geo Region Map**

The following map is plotted from the output of GREMOVE procedure that used US Census County shape file as input, and processed and loaded the output into ATTRLOOKUP and CENTLOOKUP datasets. Second input used here is the US County to Custom Region (C2R) mapping table.


**Figure 4. Orion Star Regional Inc. Custom Regions Sales plotted on a SAS Visual Analytics Geo Region Map**

## ZCTA TO REGION (Z2R)

The following map is plotted by importing the US Census ZCTA shape file and processing and loading it into ATTRLOOKUP and CENTLOOKUP datasets.

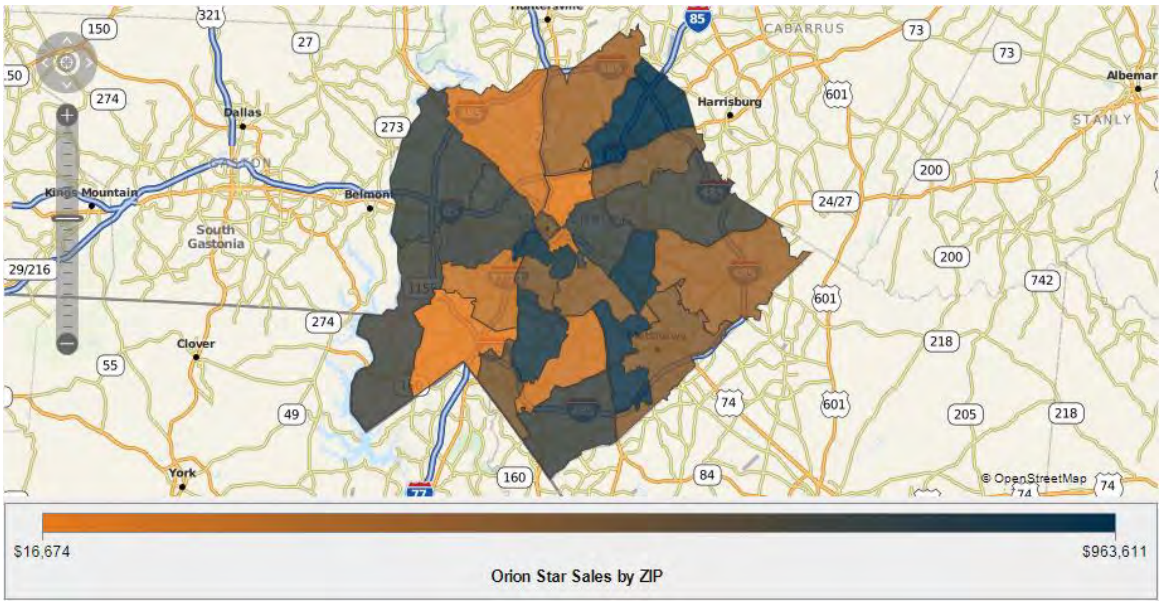



**Figure 5. Orion Star Local Inc. ZIP Sales plotted on a SAS Visual Analytics Geo Region Map**

The following map is plotted from the output of GREMOVE procedure that used US Census ZCTA shape file as input, and processed and loaded the output into ATTRLOOKUP and CENTLOOKUP datasets. Second input used here is the ZCTA to Custom Region (Z2R) mapping table.

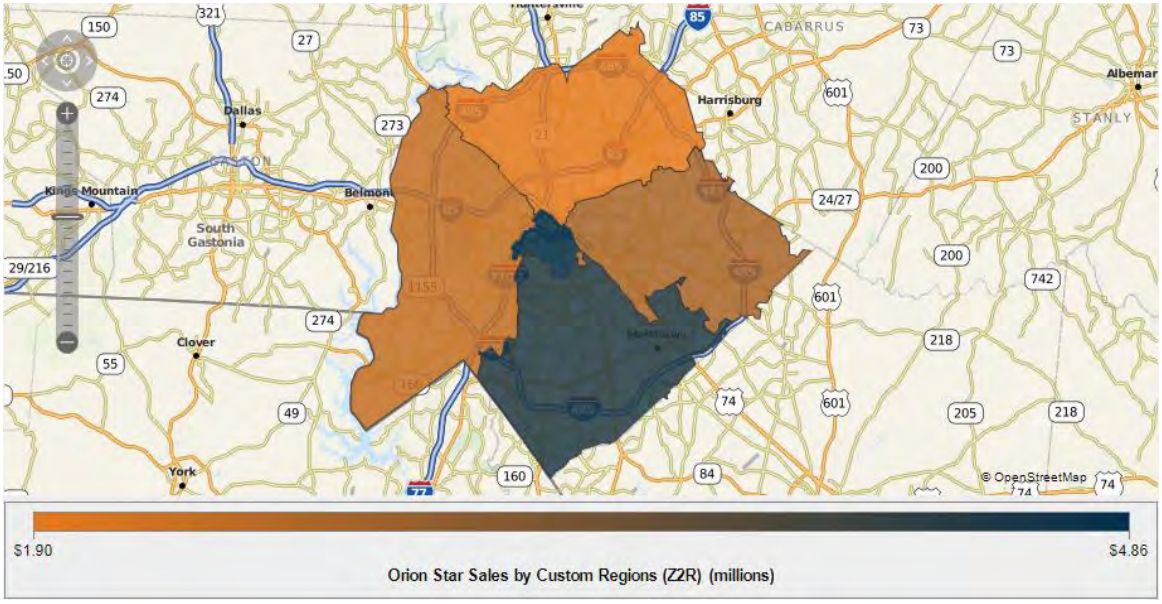



**Figure 6. Orion Star Local Inc. Custom Regions Sales plotted on SAS Visual Analytics Geo Region Map**

## CONCLUSION

As long as your organization's custom regions can be derived from any standard geographic shapes, you can easily derive the map of your custom regions using GREMOVE procedure and update your Visual Analytics configuration to plot them on SAS® Visual Analytics Geo Region Maps.

## REFERENCES

Hall, Angela M. 2016, "Creating Custom Map Regions in SAS® Visual Analytics," *Proceedings of the SAS Global Forum 2016 Conference.* Cary, NC. SAS Institute Inc. Available at:

https://sasglobalforum2016.lanyonevents.com/connect/fileDownload/session/79B7F9262FE3FC180EAD672D847A6B12/SAS3460_Hall-3460-2016.pdf

## RECOMMENDED READING

- *SAS/GRAPH® 9.4: Mapping Reference. GREMOVE Procedure*

## CONTACT INFORMATION

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

Jitendra N. Pandey
Electrolux Home Products. Inc., Charlotte, NC
jitendra.pandey@electrolux.com

```
/************************************************************************/
/* WARNING: PLEASE MAKE SURE YOU BACKUP ATTRLOOKUP & CENTLOOKUP DATA SETS */
/* USING A METHOD MOST COMFORTABLE TO YOU, BEFORE RUNNING THIS PROGRAM.   */
/************************************************************************/
/*            01_ADD_STANDARD_GEOGRAPHIC_POLYGONS_TO_VA.SAS             */
/************************************************************************/
/* SUBJECT: SAS GLOBAL FORUM 2018                                       */
/* PAPER NO: 2885-2018                                                  */
/* TITLE: PLOT YOUR CUSTOM REGIONS ON SAS(R) VISUAL ANALYTICS GEO MAPS    */
/* PROG: ADD_STANDARD_POLYGONS_TO_VA.SAS                                */
/* AUTHOR: JITENDRA N. PANDEY                                           */
/* EMAIL: JITENDRA.PANDEY@ELECTROLUX.COM, JITENDRA.PANDEY@GMAIL.COM       */
/*----------------------------------------------------------------------*/
/* DESCRIPTION:                                                         */
/* THIS PROGRAM READS THE STANDARD GEOGRAPHIC SHAPE FILED DOWNLOADED FROM */
/* FROM US CENSUS WEBSITE USING MAPIMPORT AND GREDUCE PROCEDURES AND IT   */
/* GENERATES MAP DATA SETS OF THOSE STANDARD SHAPES. THE MAP DATA SET OF  */
/* STANDARD SHAPES ARE THEN LOADED INTO ATTRLOOKUP AND CENTLOOPUP DATA    */
/* SET OF YOUR SAS DEPLOYMENT.                                          */
/*----------------------------------------------------------------------*/
/* PARAMETER DEFINITIONS:                                               */
/*----------------------------------------------------------------------*/
/* STDPOLYTYPE:                                                         */
/* STANDARD GEOGRAPHY POLYGON TYPE, ZIP, COUNTY, DIVISION OR REGION.      */
/*                                                                      */
/* SHAPEFILE:                                                           */
/* STANDARD GEOGRAPHIC SHAPE FILE DOWNLOADED FROM US CENSUS WEBSITE.      */
/* PLEASE NOTE THAT THE FOLDER LOCATION OF THE SHAPE FILES IS GIVEN       */
/* IN THE SETTINGS SECTION OF THE PROGRAM.                              */
/*                                                                      */
/* IDVAR:                                                               */
/* STANDARD GEOGRAPHY SHAPE FILE ID VAR. ZCTA5CE10, GEOID, GEOID, GEOID.  */
/*                                                                      */
/* DENSITY:                                                             */
/* HIGHEST VALUE OF GREDUCE DENSITY TO CONSIDER. 5, 4, 4, 4.              */
/*                                                                      */
/* REGION_LABEL:                                                        */
/* VALUE TO POPULATE IDLABEL, IDNAME, ISONAME AND KEY COLUMNS. THE VALUES */
/* USED: US ZIP Codes, US Counties, US Divisions, US Regions.             */
/*                                                                      */
/* REGION_PREFIX:                                                       */
/* UNIQUE 2-LETTER ISO CODE. ZZ, ZC, ZD, ZR                             */
/*                                                                      */
/* REGION_ISO:                                                          */
/* UNIQUE ISO CODE, 000, 000, 000, 000.                                 */
/*                                                                      */
/* PROVINCE_LABEL:                                                      */
/* VALUE TO POPULATE IDNAME IN PROVINCE DATA SET.                       */
/*                                                                      */
/* PROVINCE_DATASET:                                                    */
/* THE NAME OF MAP DATA SET OF THE SHAPES DOWNLODED FROM US CENSUS.       */
/************************************************************************/
%macro ImportStandardShapes
       (STDPOLYTYPE=,
        SHAPEFILE=,
        IDVAR=,
        DENSITY=,
```

```
         REGION_LABEL=,
         REGION_PREFIX=,
         REGION_ISO=,
         PROVINCE_LABEL=,
         PROVINCE_DATASET=
        );

     /*****************************************/
     /* SETTINGS BASED ON YOUR FOLDER STRUCTURE */
     /*****************************************/
     %let VALIBLOC=D:\SAS\Config\Lev1\SASAppVA\Data\VALIB;
     %let BACKUPLOC=D:\SAS\Config\Lev1\SASAppVA\Data\VALIB\BACKUP;
     %let SHAPEFILELOC=D:\SASGF2018\SHAPE_FILES;
     %let POLYDATA=D:\SASGF2018\POLYGON_DATA;

     libname VALIB base "&VALIBLOC.";
     libname VALIBBAK base "&BACKUPLOC.";
     libname POLYDATA base "&POLYDATA.";

     /*********************************************************/
     /* PLEASE USE THE BACKUP METHOD YOU ARE COMFORTABLE WITH */
     /*      BACKUP ATTRLOOKUP AND CENTLOOKUP DATA SETS       */
     /*********************************************************/
     proc copy out=VALIBBAK in=VALIB datecopy memtype=data;
          select attrlookup centlookup;
     run;

     /*********************************************************/
     /* IMPORT THE SHAPE FILE DOWNLOADED FROM US CENSUS WEBSITE */
     /*********************************************************/
     proc mapimport datafile="&SHAPEFILELOC.\&SHAPEFILE."
out=polydata.&STDPOLYTYPE._shape;
          ID &IDVAR.;
     run;
     proc greduce data=polydata.&STDPOLYTYPE._shape
out=polydata.&STDPOLYTYPE._polygons_hd;
          ID &IDVAR.;
     run;
     data work.&STDPOLYTYPE._polygon1;
          set polydata.&STDPOLYTYPE._polygons_hd;
          if DENSITY <= &DENSITY.;
     run;
     data work.&STDPOLYTYPE._polygon1;
          set work.&STDPOLYTYPE._polygon1;
          SEQUENCE=_N_;
     run;
     proc sort data=work.&STDPOLYTYPE._polygon1
out=work.&STDPOLYTYPE._polygon2(drop=SEQUENCE);
          by &IDVAR. SEQUENCE;
     run;
     data &PROVINCE_DATASET.;
          set work.&STDPOLYTYPE._polygon2;
          length ID $15;
          ID = "&REGION_PREFIX.-"||compress(&IDVAR.);
          IDNAME = compress("&PROVINCE_LABEL.-" || compress(&IDVAR.));
          LONG = X;
          LAT = Y;
```

```
          ISO = "&REGION_ISO.";
          RESOLUTION = 1;
          LAKE = 0;
          ISOALPHA2 = "&REGION_PREFIX.";
          ADMINTYPE = "ZIPPOLYGONS";
          keep ID SEGMENT IDNAME LONG LAT X Y ISO DENSITY RESOLUTION LAKE
ISOALPHA2 ADMINTYPE;
     run;
     proc sql noprint;
          delete * from valib.attrlookup where ISONAME = "&REGION_LABEL.";
          delete * from valib.centlookup where MAPNAME =
"&PROVINCE_DATASET.";
     quit;
     proc sql noprint;
          insert into valib.attrlookup
          values (
               "&REGION_LABEL.",    /* IDLABEL=STATE/PROVINCE LABEL */
               "&REGION_PREFIX.",   /* ID=SAS MAP ID VALUE */
               "&REGION_LABEL.",    /* IDNAME=STATE/PROVINCE NAME */
               "",                  /* ID1NAME=COUNTRY NAME */
               "",                  /* ID2NAME */
               "&REGION_ISO.",      /* ISO=COUNTRY ISO NUMERIC CODE */
               "&REGION_LABEL.",    /* ISONAME */
               "&REGION_LABEL.",    /* KEY */
               "",                  /* ID1=COUNTRY ISO 2-LETTER CODE */
               "",                  /* ID2 */
               "",                  /* ID3 */
               "",                  /* ID3NAME */
               0                    /* LEVEL (0=COUNTRY LEVEL, 1=STATE LEVEL)
*/
              );
     quit;
     proc sql noprint;
          create table work.attrlookup_add as
          select distinct
          IDNAME as IDLABEL,            /* IDLABEL=STATE/PROVINCE LABEL */
          ID as ID,                     /* ID=SAS MAP ID VALUE */
          IDNAME as IDNAME,             /* IDNAME=STATE/PROVINCE NAME */
          "&REGION_LABEL." as ID1NAME,  /* ID1NAME=COUNTRY NAME */
          "" as ID2NAME,                /* ID2NAME */
          "&REGION_ISO." as ISO,        /* ISO=COUNTRY ISO NUMERIC CODE */
          "&REGION_LABEL." as ISONAME,  /* ISONAME */
          trim(IDNAME)||"|&REGION_LABEL." as KEY, /* KEY */
          "&REGION_PREFIX." as ID1,     /* ID1=COUNTRY ISO 2-LETTER CODE */
          "" as ID2,                    /* ID2 */
          "" as ID3,                    /* ID3 */
          "" as ID3NAME,                /* ID3NAME */
          1 as LEVEL                    /* LEVEL (1=STATE LEVEL) */
          from &PROVINCE_DATASET.
          order by ID; /* ORDER BY MAKES SURE THAT THE STANDARD SHAPES YOU
          ARE ADDEING ARE SORED BY ID VARIABLE, THIS IS NOT MANDATORY.  */
     quit;
     data valib.attrlookup;
          set valib.attrlookup  work.attrlookup_add;
     run;
     proc datasets lib=work nolist;  delete attrlookup_add;  quit;  run;
     proc sql noprint;
```

```
          insert into valib.centlookup select distinct
          "&PROVINCE_DATASET." as MAPNAME,
          "&REGION_PREFIX." as ID,
          avg(X) as X,
          avg(Y) as Y
          from &PROVINCE_DATASET.;

          insert into valib.centlookup select distinct
          "&PROVINCE_DATASET." as MAPNAME,
          ID as ID,
          avg(X) as X,
          avg(Y) as Y
          from &PROVINCE_DATASET. group by ID;
     quit;
%mend ImportStandardShapes;

%ImportStandardShapes
     (STDPOLYTYPE=ZIP,
      SHAPEFILE=CB_2016_US_ZCTA510_500K.SHP,
      IDVAR=ZCTA5CE10,
      DENSITY=5,
      REGION_LABEL=US ZIP Codes,
      REGION_PREFIX=ZZ,
      REGION_ISO=000,
      PROVINCE_LABEL=ZIP,
      PROVINCE_DATASET=VALIB.USZIPPOLY1);
%ImportStandardShapes
     (STDPOLYTYPE=COUNTY,
      SHAPEFILE=CB_2016_US_COUNTY_500K.SHP,
      IDVAR=GEOID,
      DENSITY=4,
      REGION_LABEL=US Counties,
      REGION_PREFIX=ZC,
      REGION_ISO=000,
      PROVINCE_LABEL=COUNTY,
      PROVINCE_DATASET=VALIB.USCOUNTY1);
%ImportStandardShapes
     (STDPOLYTYPE=DIVISION,
      SHAPEFILE=CB_2016_US_DIVISION_500K.SHP,
      IDVAR=GEOID,
      DENSITY=4,
      REGION_LABEL=US Divisions,
      REGION_PREFIX=ZD,
      REGION_ISO=000,
      PROVINCE_LABEL=DIVISION,
      PROVINCE_DATASET=VALIB.USDIVISION1);
%ImportStandardShapes
     (STDPOLYTYPE=REGION,
      SHAPEFILE=CB_2016_US_REGION_500K.SHP,
      IDVAR=GEOID,
      DENSITY=4,
      REGION_LABEL=US Regions,
      REGION_PREFIX=ZR,
      REGION_ISO=000,
      PROVINCE_LABEL=REGION,
      PROVINCE_DATASET=VALIB.USREGION1);
```

```
/*************************************************************************/
/* WARNING: PLEASE MAKE SURE YOU BACKUP ATTRLOOKUP & CENTLOOKUP DATA SETS */
/* USING A METHOD MOST COMFORTABLE TO YOU, BEFORE RUNNING THIS PROGRAM.   */
/*************************************************************************/
/*             02_DERIVE_ADD_CUSTOM_POLYGONS_TO_VA.SAS                   */
/*************************************************************************/
/* SUBJECT: SAS GLOBAL FORUM 2018                                        */
/* PAPER NO: 2885-2018                                                   */
/* TITLE: PLOT YOUR CUSTOM REGIONS ON SAS(R) VISUAL ANALYTICS GEO MAPS    */
/* PROG: DERIVE_AND_ADD_CUSTOM_POLYGONS_TO_VA.SAS                        */
/* AUTHOR: JITENDRA N. PANDEY                                            */
/* EMAIL: JITENDRA.PANDEY@ELECTROLUX.COM, JITENDRA.PANDEY@GMAIL.COM       */
/*-----------------------------------------------------------------------*/
/* DESCRIPTION:                                                          */
/* THIS PROGRAM READS THE STANDARD TO CUSOTM GEOGRAPHIC MAPPING FILES,    */
/* MERGES IT WITH THE MAP DATA SETS OF STANDARD GEOGRAPHIES TO GET THE    */
/* COORDINATES OF ALL CONSTITUTING SHAPES PROCESS THIS DATA SET USING     */
/* GREMOVE PROCEDURE AND USES THE OUTPUT OF GREMOVE PROCEDURE TO LOAD     */
/* CUSTOM REGION ENTRIES INTO CENTLOOKUP AND ATTRLOOKUP DATA SETS.        */
/*-----------------------------------------------------------------------*/
/* PARAMETER DEFINITIONS:                                                */
/* FOLLOWED BY THE VALUES USED IN THE PAPER IN ORDER: ZIP, COUNTY,        */
/* DIVISION, REGION.                                                     */
/*                                                                       */
/* MYREGIONMAPTABLE:                                                     */
/* STANDARD GEOGRAPHY TO CUSTOM REGION MAPPING TABLE FILE. THE FILES USED */
/* IN THIS PAPER:                                                        */
/* ORION_STAR_REGIONS_Z2R.TXT, ORION_STAR_REGIONS_Z2R.TXT,               */
/* ORION_STAR_REGIONS_Z2R.TXT, ORION_STAR_REGIONS_Z2R.TXT.               */
/*                                                                       */
/* STDUNIT:                                                              */
/* UNIT OF CONSTITUTING STANDARD GEOGRAPY. ZIP, COUNTY, DIVISION, REGION. */
/*                                                                       */
/* MAPTYPE:                                                              */
/* STANDARD GEOGRAPHY TO CUSTOM REGION MAPPING TABLE, THIS IS USED AS AN  */
/* IDENTIFIER FOR THE DATASETS CREATED IN THE PROGRAM. Z2R, C2R, D2R, R2R.*/
/*                                                                       */
/* IDVAR:                                                                */
/* STANDARD GEOGRAPHY SHAPE FILE ID VAR. ZCTA5CE10, GEOID, GEOID, GEOID.  */
/*                                                                       */
/* DENSITY:                                                              */
/* HIGHEST VALUE OF GREDUCE DENSITY TO CONSIDER. 5, 4, 4, 4.             */
/*                                                                       */
/* REGION_LABEL:                                                         */
/* VALUE TO POPULATE IDLABEL, IDNAME, ISONAME AND KEY COLUMNS. THE VALUES */
/* USED: OrionStarRegionsByZIP, OrionStarRegionsByCounty,                */
/* OrionStarRegionsByDivision, OrionStarRegionsByRegion.                 */
/*                                                                       */
/* REGION_PREFIX:                                                        */
/* UNIQUE 2-LETTER ISO CODE. OZ, OC, OD OR OR.                           */
/*                                                                       */
/* REGION_ISO:                                                           */
/* UNIQUE ISO CODE. 000, 000, 000, 000.                                  */
/*                                                                       */
/* REGION_DATASET:                                                       */
/* THE NAME OF CUSTOM REGION MAP DATA SET.  MAKE SURE THAT THE NAME OF    */
/* THIS DATASET ENDS WITH 1, WHICH IS THE LEVEL VALUE FOR THIS REGIONS'S  */
```

```
/* ENTRY IN ATTRLOOKUP DATASET. THE VALUES USED IN THIS PAPER ARE:        */
/* ORIONZ2R1, ORIONC2R1, ORIOND2R1, ORIONR2R1.                            */
/*************************************************************************/
%macro DriveImportCustomRegions
       (MYREGIONMAPTABLE=,
        STDUNIT=,        /* ZIP, COUNTY, DIVISION, REGION */
        MAPTYPE=,        /* Z2R, C2R, D2R, R2R */
        IDVAR=,          /* ZCTA5CE10, GEOID, GEOID, GEOID */
        DENSITY=,        /* 5, 4, 4, 4 */
        REGION_LABEL=,   /* OrionStarRegionsByZIP, OrionStarRegionsByCounty,
        OrionStarRegionsByDivision, OrionStarRegionsByRegion */
        REGION_PREFIX=,  /* OZ, OC, OD OR OR */
        REGION_ISO=,     /* 000, 000, 000, 000 */
        REGION_DATASET=, /* VALIB.ORIONZ2R1, VALIB.ORIONC2R1,
VALIB.ORIOND2R1,
        VALIB.ORIONR2R1 */
       );

     /*****************************************/
     /* SETTINGS BASED ON YOUR FOLDER STRUCTURE */
     /*****************************************/
     %let VALIBLOC=D:\SAS\Config\Lev1\SASAppVA\Data\VALIB;
     %let POLYDATA=D:\SASGF2018\POLYGON_DATA;
     %let SHAPEFILELOC=D:\SASGF2018\SHAPE_FILES;

     libname VALIB base "&VALIBLOC.";
     libname POLYDATA base "&POLYDATA.";

     proc sql noprint;
          create table work.&STDUNIT._polygons as
          select L.*, monotonic() as SEQUENCE from
polydata.&STDUNIT._polygons_hd as L
          where L.DENSITY le &DENSITY.;
     quit;

     /*************************************************/
     /* READ ORION START'S CUSTOM REGION MAPPING TABLE */
     /*************************************************/
     data work.orion_star_&MAPTYPE.;
          infile "&SHAPEFILELOC.\&MYREGIONMAPTABLE." dsd firstobs=2;
          length CUST_REGION_NAME $32  REGION $15  &STDUNIT._NAME $32
&STDUNIT._CODE $5;
          input CUST_REGION_NAME  REGION  &STDUNIT._NAME  &STDUNIT._CODE;
     run;

     /*********************************************************************/
     /* MERGE IT WITH THE POLYGON DATA SET TO GET THE COORDINATES OF ALL  */
     /* CONSTITUTING POLYGONS.                                            */
     /*********************************************************************/
     proc sql noprint;
          create table work.orion_star_&MAPTYPE._shape1 as
          select R.CUST_REGION_NAME, R.REGION, L.*
          from work.&STDUNIT._polygons as L right join
work.orion_star_&MAPTYPE. as R
          on L.&IDVAR.=R.&STDUNIT._CODE
          order by R.REGION, L.SEQUENCE;
     quit;
```

```
     /*******************************************************/
     /* USE GREMOVE PROCEDURE TO REMOVE INTERNAL BOUNDARIES */
     /*******************************************************/
     proc gremove data=work.orion_star_&MAPTYPE._shape1 out=&REGION_DATASET.;
          by REGION;
          id &IDVAR.;
     run;
     data &REGION_DATASET.;
          set &REGION_DATASET.;
          ID=REGION;
          IDNAME=CUST_REGION_NAME;
          LONG=X;
          LAT=Y;
          ISO="&REGION_ISO.";
          RESOLUTION=1;
          LAKE=0;
          ISOALPHA2="&REGION_PREFIX.";
          ADMINTYPE="ORION STAR REGIONS";
          keep ID SEGMENT IDNAME LONG LAT X Y ISO DENSITY RESOLUTION LAKE
ISOALPHA2 ADMINTYPE;
     run;
     proc sql noprint;
          insert into valib.attrlookup
          values (
               "&REGION_LABEL.",    /* IDLABEL=STATE/PROVINCE LABEL */
               "&REGION_PREFIX.",   /* ID=SAS MAP ID VALUE */
               "&REGION_LABEL.",    /* IDNAME=STATE/PROVINCE NAME */
               "",                  /* ID1NAME=COUNTRY NAME */
               "",                  /* ID2NAME */
               "&REGION_ISO.",      /* ISO=COUNTRY ISO NUMERIC CODE */
               "&REGION_LABEL.",    /* ISONAME */
               "&REGION_LABEL.",    /* KEY */
               "",                  /* ID1=COUNTRY ISO 2-LETTER CODE */
               "",                  /* ID2 */
               "",                  /* ID3 */
               "",                  /* ID3NAME */
               0                    /* LEVEL (0=COUNTRY, 1=STATE) */
              );
     quit;
     proc sql noprint;
          insert into valib.attrlookup
               select distinct
               IDNAME,              /* IDLABEL=STATE/PROVINCE LABEL */
               ID,                  /* ID=SAS MAP ID VALUE */
               IDNAME,              /* IDNAME=STATE/PROVINCE NAME */
               "&REGION_LABEL.",    /* ID1NAME=COUNTRY NAME */
               "",                  /* ID2NAME */
               "&REGION_ISO.",      /* ISO=COUNTRY ISO NUMERIC CODE */
               "&REGION_LABEL.",    /* ISONAME */
               trim(IDNAME) || "|&REGION_LABEL.",  /* KEY */
               "&REGION_PREFIX.",   /* ID1=COUNTRY ISO 2-LETTER CODE */
               "",                  /* ID2 */
               "",                  /* ID3 */
               "",                  /* ID3NAME */
               1                    /* LEVEL (1=STATE LEVEL) */
               from &REGION_DATASET.;
```

```
     quit;
     proc sql noprint;
          insert into valib.centlookup
          select distinct
               "&REGION_DATASET." as MAPNAME,
               "&REGION_PREFIX." as ID,
               avg(X) as X,
               avg(Y) as Y
               from &REGION_DATASET.;

          insert into valib.centlookup
               select distinct
               "&REGION_DATASET." as MAPNAME,
               ID as ID,
               avg(X) as X,
               avg(Y) as Y
               from &REGION_DATASET.
               group by ID;
     quit;
%mend DriveImportCustomRegions;

%DriveImportCustomRegions(
     MYREGIONMAPTABLE=ORION_STAR_REGIONS_Z2R.TXT,
     STDUNIT=ZIP,
     MAPTYPE=Z2R,
     IDVAR=ZCTA5CE10,
     DENSITY=5,
     REGION_LABEL=OrionStarRegionsByZIP,
     REGION_PREFIX=OZ,
     REGION_ISO=000,
     REGION_DATASET=VALIB.ORIONZ2R1);
%DriveImportCustomRegions(
     MYREGIONMAPTABLE=ORION_STAR_REGIONS_C2R.TXT,
     STDUNIT=COUNTY,
     MAPTYPE=C2R,
     IDVAR=GEOID,
     DENSITY=4,
     REGION_LABEL=OrionStarRegionsByCounty,
     REGION_PREFIX=OC,
     REGION_ISO=000,
     REGION_DATASET=VALIB.ORIONC2R1);
%DriveImportCustomRegions(
     MYREGIONMAPTABLE=ORION_STAR_REGIONS_D2R.TXT,
     STDUNIT=DIVISION,
     MAPTYPE=D2R,
     IDVAR=GEOID,
     DENSITY=4,
     REGION_LABEL=OrionStarRegionsByDivision,
     REGION_PREFIX=OD,
     REGION_ISO=000,
     REGION_DATASET=VALIB.ORIOND2R1);
```