Paper SAS 4335-2020

Visualizing Geographical Data with a Tile Grid Map in SAS®

Lingxiao Li, SAS Institute Inc.

ABSTRACT

The tile grid map is an increasingly popular tool to visualize statistical data, such as the US population change by state, on a map. It mimics an actual map with a set of equal squares in a rectangular grid. For example, you can make a tile grid map of the United States with each square representing an individual state. Unlike the choropleth map, the tile grid map does not show the perception bias that favors larger regions. Furthermore, the squares are well-suited for laying out "rubber stamp" graphs with subsetted data to create map-based small multiples. You can use the small multiples to effectively compare and analyze data in different regions of the map. Although the Graph Template Language (GTL) does not directly support this visualization, you can easily make one in SAS with a combination of the DATA step, the SQL and SUMMARY procedures, and the SAS® ODS Graphics procedures (often called the Statistical Graphics procedures). This paper uses examples to show you how to systematically create a variety of tile grid maps that include time series and infographics.

INTRODUCTION

You can use the ODS Graphics system in SAS® to create all kinds of graphs, from the simple scatter plot to the more complicated multi-cell paneled graphs. It provides you with a rich set of features that allows you to creatively assemble a graph for visualizing hidden gems in the data from a unique perspective. In this paper, we will show you how to use ODS Graphics to create one such highly customized graph: the tile grid map.

If you want to visualize the average US population growth rate by state, you can create a classic choropleth map with the SGMAP procedure as shown in Figure 1.

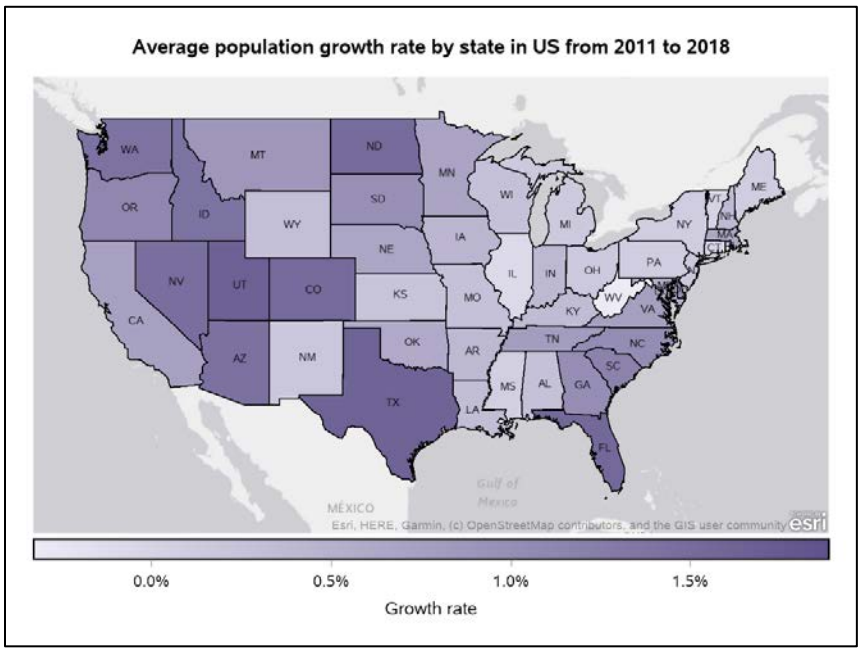

Figure 1. Average US Population Growth by State

The choropleth map inherently favors larger map regions and, as a result, could distort the traditional viewers' perceptions of the statistics in the map. In addition, if you want to identify more detailed information such as the population growth trend in each state, the polygonal shapes of the states with varying sizes make it difficult to label the states and include small multiples.

In recent years, a new type of map called the tile grid map has gained popularity as an alternate tool for visualizing geographical data. Other well-known names for the tile grid map are tile cartograms and mosaic cartograms. News outlets such as The Washington Post were among the first to adopt this mapping method (Keating and Park, 2015). You can view the blog by Severino Ribecca (2018) for a nice survey of the tile grid map.

Figure 2 is a tile grid map that uses the same data as Figure 1 to show the average population growth-rate by state as well as the growth-rate trend in each state in one view.

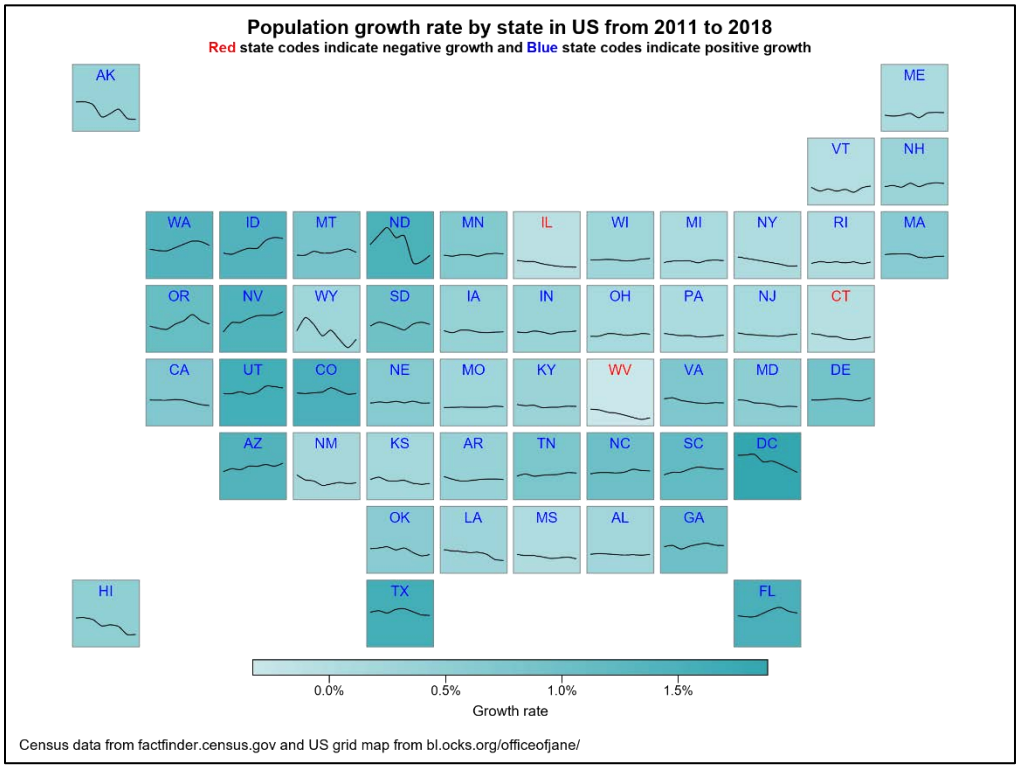

Figure 2. US Population Growth by State

This graph reveals all kinds of interesting findings about the population change in the US. The shading of each tile represents the average growth-rate, the embedded time series shows the growth-rate trend for each state, and lastly, the color of the state code indicates whether the state had positive or negative average growth. The graph is created by none other than your very own ODS Graphics!

The tile grid map does not display a precise geographical map like the choropleth map. However, it does not silence the smaller areas, and at the same time, the square shapes make it much easier to include small multiples like the time series above. In addition, the map-approximation of the small multiples helps you compare the statistics across geographical regions and locate any region-of-interest for more detailed information in the form of an embedded graph.

In this paper, you will learn how to use the DATA step, the SQL procedure, the SUMMARY procedure, and ODS Graphics to produce a tile grid map. The code that generates the tile

grid map in Figure 2 can be easily extended to handle different data and other types of small multiples.

We will break down the creation of the tile grid map into three major steps:

1. Data preparation.
2. Generating the small-multiple graphs as images.
3. Using the images from Step 2 as custom markers in a scatter plot of the grid map data.

DATA PREPARATION

To make a tile grid map, you need two kinds of data sets: an equally spaced rectangular grid that approximates the geographical regions, and an associated data set containing the response data that you want to visualize.

THE US GRID MAP

There are many versions of the rectangular grid that can be used for a US tile grid map. For example, Jane Pong (2019) has a website that collects a variety of the US grid maps from different news organizations. For most examples in this paper, we will use the version from the National Public Radio (NPR) as shown in Figure 3.

AK											ME
										VT	NH
	WA	ID	MT	ND	MN	IL	WI	MI	NY	RI	MA
	OR	NV	WY	SD	IA	IN	OH	PA	NJ	CT	
	CA	UT	CO	NE	MO	KY	WV	VA	MD	DE	
		AZ	NM	KS	AR	TN	NC	SC	DC		
				OK	LA	MS	AL	GA			
HI				TX					FL		

Figure 3. US Grid Map from NPR

Part of the data set is shown in Table 1. The data set named GRIDMAP has three columns of interest: STATE, X, and Y. The STATE column contains the state codes or abbreviations. This column serves as the link between the grid-map data set and the response data set.

Obs	state	x	y
1	AK	2	1
2	AL	9	7
3	AR	7	6
4	AZ	4	6
5	CA	3	5

Table 1. Part of the US Grid Map Data (GRIDMAP)

In this paper, the basic idea for making the tile grid map is to use a scatter plot to draw custom markers representing the small multiples at the XY locations of the grid.

THE US POPULATION GROWTH DATA

In Figure 2, the US population growth data set named POPGROWTH is gathered from the US census data between 2010 and 2018, as partially shown in Table 2. The columns that are of interest are YEAR, STATE, and RATE. The RATE column serves as the response variable and is computed as the growth rate over the previous year. This data set will be used to generate all the small-multiple time series.

Obs	year	state	rate
1	2011	AK	0.011391
2	2012	AK	0.011580
3	2013	AK	0.009099
4	2014	AK	-0.001001
5	2015	AK	0.001684
6	2016	AK	0.005365
7	2017	AK	-0.002317
8	2018	AK	-0.003174
9	2011	AL	0.002797
10	2012	AL	0.003486
11	2013	AL	0.003093

Table 2. Part of the Population Growth Data (POPGROWTH)

In the following sections, we will go through several data-preparation steps that are needed prior to making the tile grid map.

COMPUTING THE MEAN GROWTH RATE BY STATE

We first use PROC SUMMARY to compute the mean growth rate for each state:

```
proc summary data=popgrowth nway;
  var rate;
  class state;
  output out=popgrowth_mean(drop=_type_ _freq_) mean=;
run;
```

The output data set is POPGROWTH_MEAN. We then merge it with the GRIDMAP data set:

```
data mygridmap;
    merge gridmap popgrowth_mean;
    by state;
    texty=y-0.3;
run;
quit;
```

As it will become clear in the code on Page 8, we add a new column TEXTY that offsets the Y coordinate by a fraction to help label the state. This merged data set is the one that we will use to assemble the final tile grid map of Figure 2.

COMPUTING THE EXTREME VALUES

The final tile grid map needs to equate the row spacing and the column spacing. This can be accomplished by computing the ranges of the X and Y coordinates to derive the aspect ratio of the US grid map. In addition, the time series that shows the growth rate for each state in Figure 2 needs a uniform scale to help us better compare trends across states. Therefore, we need to compute the extreme values of the rates from the original response data set. Finally, we want to place the distinct state codes to a space-separated list for later use. The following PROC SQL code computes the values and puts the results into the macro variables X_MIN, X_MAX, Y_MIN, Y_MAX, RATE_MIN, RATE_MAX, and SYMBOLS:

```
proc sql noprint;
  select min(x), max(x), min(y), max(y) into :x_min, :x_max, :y_min, :y_max
         from mygridmap;
  select min(rate), max(rate) into :resp_min, :resp_max from popgrowth;
  select distinct state into :symbols separated by ' ' from popgrowth;
quit;
```

We then compute ASPECT, the aspect ratio of the grid map, as follows:

```
%let aspect=%sysevalf((&y_max-&y_min)/(&x_max-&x_min));
```

In the following sections, we will use the two data sets POPGROWTH and MYGRIDMAP as well as the eight macro variables above to make the tile grid map.

GENERATING THE SMALL-MULTIPLE GRAPHS AS IMAGES

THE SMALL-MULTIPLE GRAPH

To derive the small-multiple graph for each state in Figure 2, we define the following macro function using the SGPLOT procedure:

```
%macro small_multiple(symbol=,     /* symbol name like the state code */
                      data=,       /* data set */
                      whereclause= /* where clause for this symbol */);
  ods graphics / reset imagename="&symbol" width=&img_size height=&img_size
                 border=off scale=off;
  proc sgplot data=&data(where=(&whereclause)) nowall noautolegend noborder
              noopaque pad=(left=3px right=3px top=&pad_top bottom=3px);
    /* Hide the axes */
    xaxis display=none;
    /* Use MIN and MAX to get the uniform scale for the response */
    yaxis display=none min=&resp_min max=&resp_max;
    /* The small-multiple time series */
    series x=&time_var y=&resp_var / smoothconnect;
  run;
%mend small_multiple;
```

With the NOOPAQUE option in the SGPLOT statement, the small-multiple graphs are generated as images with a transparent background. The PAD option leaves some extra room at the top for labeling the state in Figure 2.

The above SAS macro function serves as a rubber stamp with which you can generate all the small multiples by subsetting the data for each state. You can customize the code in the macro function to fit your own visualization needs or preference. For example, you can customize the image size and the line attributes, among others. You can also replace the SERIES statement with a NEEDLE statement. Later in the paper, we will use additional examples to create other types of small multiples.

GENERATING ALL SMALL-MULTIPLE GRAPHS

We define another macro function that generates the small-multiple graphs for all states with the WHERE expressions that are built by traversing the list in the SYMBOLS macro variable:

```
%macro all_small_multiples(symbols=, /* list of symbols */
                           data=,    /* data set */
                           wherevar= /* where variable for subsetting */);
  ods _all_ close;
  /* Use higher DPI to get better resolution */
  ods listing gpath="&gpath" image_dpi=&dpi;
  %let word_cnt=%sysfunc(countw(%superq(symbols)));
    %do i = 1 %to &word_cnt;
      %let var&i=%qscan(%superq(symbols),&i,%str( ));
      %small_multiple(symbol=&&var&i, data=&data,
                      whereclause=&wherevar.="&&var&i");
    %end;
  ods listing close;
%mend all_small_multiples;
```

The following code runs the above SAS macro function and produces the small-multiple graphs as images for all states:

```
/* put the generated images in the WORK directory */
%let gpath= %sysfunc(getoption(WORK));
%let time_var=year;
%let resp_var=rate;
%let dpi=200;
%let img_size=200px;
%let pad_top=40px;
%all_small_multiples(symbols=&symbols, data=popgrowth, wherevar=state);
```

A few of the small multiples are shown in Figure 4.

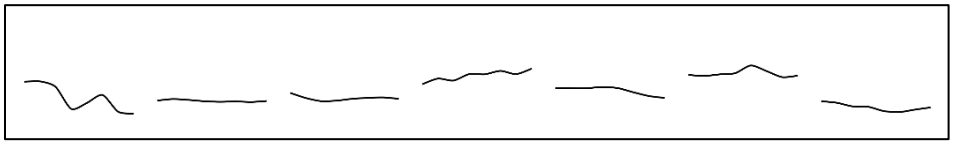

Figure 4. Time Series for AK, AL, AR, AZ, CA, CO, and CT

In the next section we will introduce the SYMBOLIMAGE statement in the SGPLOT procedure and use it to define a list of custom image markers in the tile grid map of Figure 2.

SYMBOLIMAGE IN SGPLOT

The key feature in SGPLOT that is used to create the tile grid map in this paper is the ability to use images to define custom markers with the SYMBOLIMAGE statement:

```
symbolimage name=symbol-name image="file-name" </ options>;
```

We can use the SYMBOLIMAGE statement to define a custom symbol with a name specified by the name=symbol-name option, using an image file on the file system specified in the image="file-name" option. Then, we can use this symbol to draw image markers in the graph with its assigned name just like any other built-in symbol name such as CIRCLE.

Normally, images used for the SYMBOLIMAGE statement can come from many image sources, like the ones taken by your smart phone camera or found on the web (Matange 2017). You can go a different route and use images from the ODS Graphics outputs, such as the ones in Figure 4, to define custom marker symbols and insert them in any plot that supports markers (Li 2019). This is the key feature that enables us to link the small multiples and the grid map.

Let us introduce another macro function that traverses through a list of named symbols and their pre-defined file paths (like the SAS work directory) to define a series of SYMBOLIMAGE statements:

```
%macro symbol_images(symbols /* list of symbols */);
  %let word_cnt=%sysfunc(countw(%superq(symbols)));
  %do i = 1 %to &word_cnt;
     %let var&i=%qscan(%superq(symbols),&i,%str( ));
     symbolimage name=&&&var&i image="&gpath/&&&var&i...png";
  %end;
%mend symbol_images;
```

Now we can proceed to the final step to produce the tile grid map.

PUTTING EVERYTHING TOGETHER

If we can insert the customized image markers shown in Figure 4 at the corresponding squares in Figure 3, then we would be almost there for the tile grid map in Figure 2. The rest of the steps are shading the individual areas behind the images and labeling each square with the state code.

DEFINE A RANGEATTRMAP TO COLOR THE LABELS ON THE TILES

We define a RANGEATTRMAP (range attribute map) data set to color the state-code label red if the average growth rate is negative, and blue if the rate is positive:

```
data myrattrmap;
retain id "myID";
length min $ 5 max $ 5;
input min $ max $ altcolor $;
datalines;
_min_ 0    red
0   _max_ blue
;
run;
quit;
```

This way you can easily identify which states have negative growth rates.

GENERATING THE FINAL TILE GRID MAP

To produce the scatter plot with the inserted small-multiple graphs in Figure 2, we come up with the following SGPLOT code that uses marker symbols defined with the SYMBOLIMAGE statements:

```
%let offset=0.06;
%let markersize=65px;

ods listing gpath="&gpath" image_dpi=&dpi;
ods graphics / scalemarkers=off scale=off noborder attrpriority=none;
title height=14pt 'Population growth rate by state in US from 2011 to
                   2018';
title2 height=10pt color=cxff0000 "Red" color=black " states have negative
       growth and " color=cx0000ff "Blue" color=black " states have
       positive growth";
footnote j=l 'Census data from factfinder.census.gov and US grid map from
              bl.ocks.org/officeofjane/';

/* Use the aspect ratio computed earlier to equate row and column spacings
 */
proc sgplot data=mygridmap rattrmap=myrattrmap aspect=&aspect noborder
            nowall noopaque;

  /* Define new custom markers from the small-multiple graphs */
  %symbol_images(&symbols);

  /* Use the list of custom markers as data symbols for the group */
  styleattrs datasymbols=(&symbols);

  /* Hide the axes and make the axis ranges the same as the grid map */
  xaxis display=none thresholdmin=0 thresholdmax=0
        offsetmin=&offset offsetmax=&offset min=&x_min max=&x_max;
  yaxis reverse display=none thresholdmin=0 thresholdmax=0
        offsetmin=&offset offsetmax=&offset min=&y_min max=&y_max;

  /* Scatter plot that draws the background colors of each tile */
  scatter x=x y=y / colorresponse=rate name='s' filledoutlinedmarkers
                    markerattrs=(symbol=squarefilled size=&markersize);

  /* Main scatter plot that draws the small multiples as custom markers */
  scatter x=x y=y / group=state markerattrs=(size=&markersize);

  /* Text plot that labels the tiles with state codes */
  text x=x y=texty text=state / textattrs=graphlabeltext colorresponse=rate
                               rattrid=myID;

  gradlegend 's' / title='Growth rate' position=bottom
                   outerpad=(top=10px bottom=10px left=200px right=200px);
  format rate percent.;
run;
```

In the SGPLOT code above, we use the STATE column as the GROUP role on the second scatter plot. Along with the ATTRPRIORITY=NONE option in the ODS GRAPHICS statement and the DATASYMBOLS option, we will draw a different image marker from the DATASYMBOLS list for each GROUP value of the state code. In other words, each tile representing a state in the US map gets its own small multiple.

The use of the MIN, MAX, THRESHOLDMIN, THRESHOLDMAX, OFFSETMIN, and OFFSETMAX options on the XAXIS and YAXIS, along with the ASPECT option of the SGPLOT, make sure that the aspect ratio of the tile grid map matches that of the original US grid map. This way we will ensure that the row spacing is equal to the column spacing in the tile grid map. The TEXTY column with a small vertical offset places the labels in the empty areas of the small multiples.

The code we have seen so far is very general and can be modified easily to accommodate other data and small-multiple visuals. Due to the limited space, we leave out some non-essential parts of the code here. The complete source code and data for all examples in this paper will be available at https://github.com/sascommunities/sas-global-forum-2020. The code is tested in SAS® 9.4M6.

OTHER EXAMPLES

In this section, we will show you more examples of the tile grid map that are produced by SGPLOT, SGPIE, and SGPANEL.

TILE GRID MAP OF PM2.5

Using the same US grid map from NPR but excluding the states of Alaska and Hawaii, we can produce the following tile grid map of the continental US that shows the average PM2.5 levels by state from 2003 to 2011.

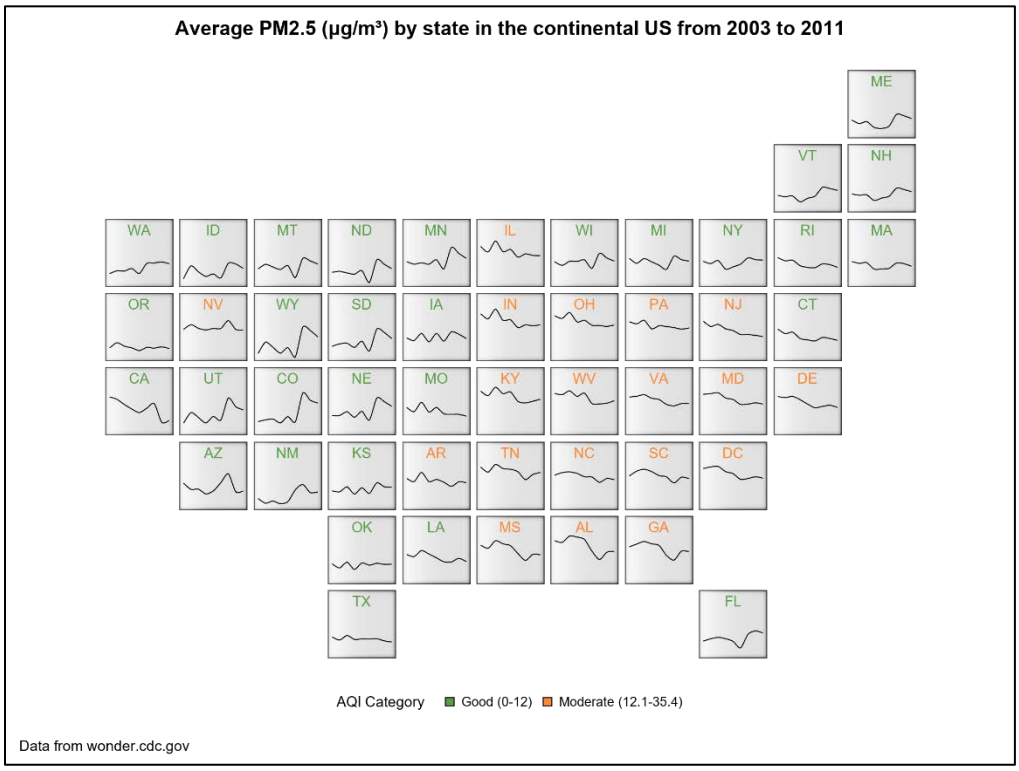

Figure 5. Tile Grid Map of PM2.5

To generate the tile grid map in Figure 5, we mostly follow the steps outlined earlier for Figure 2. The time series shows the average PM2.5 levels from 2003 to 2011 for each state. The color of the state code reflects the AQI level for "good" or "moderate" when the overall PM2.5 average is taken into consideration. The graph easily tells you which states have cleaner air over the years. It also shows that the air quality is improving for most states with declining PM2.5 levels.

TILE GRID MAP OF PRESIDENTIAL ELECTIONS FROM 1976 TO 2016

Many news organizations use tile grid maps to analyze geopolitical data. The tile grid map in Figure 6 shows the US presidential election results over recent years. The small multiples of Figure 6 are generated by modifying the SAS macro function %SMALL_MULTIPLE to use the HEATMAPPARM statement in SGPLOT. The graph highlights the voting patterns in the last eleven elections for different US regions.

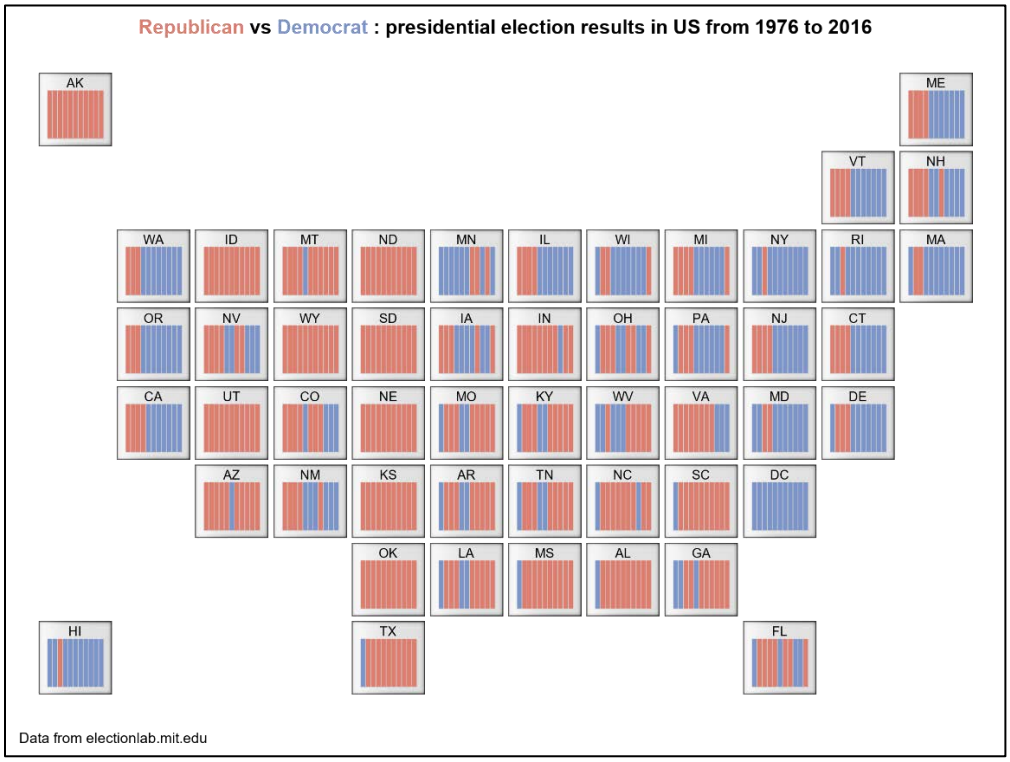

Figure 6. Tile Grid Map of Presidential Election Results

TILE GRID MAP OF US MEDIAN HOUSEHOLD INCOME

Let us use the Decorative Infographs (Matange, 2017) for our next example. Figure 7 uses a stack of coins to show the 2017 US median household income by state on a tile grid map.

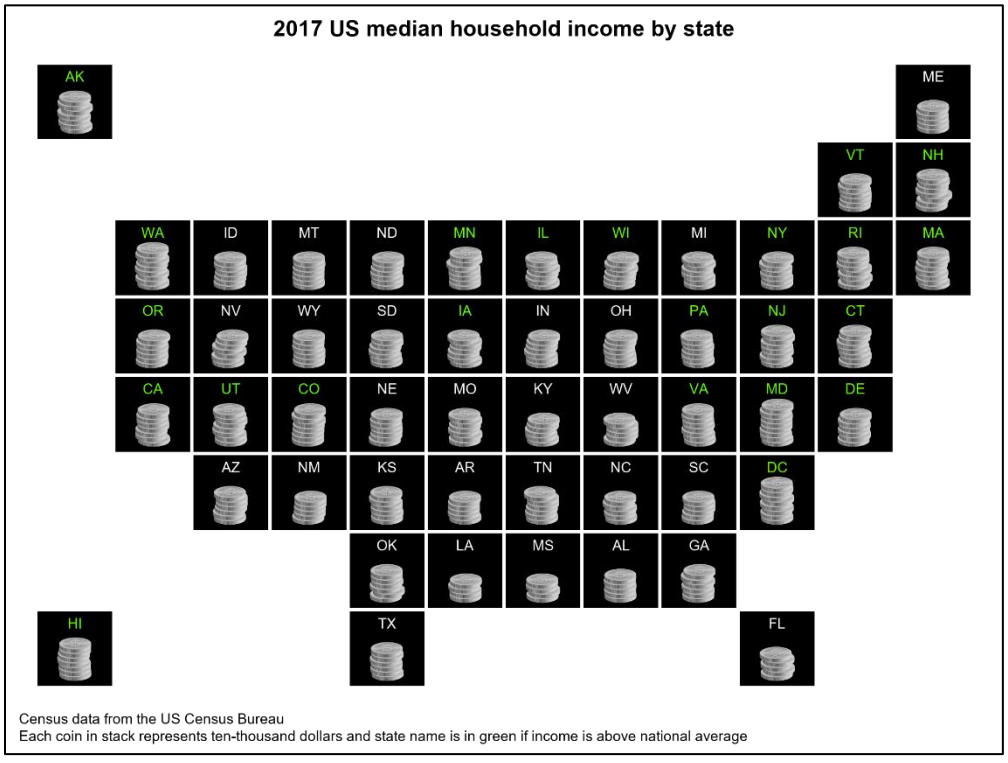

Figure 7. Tile Grid Map of US Median Household Income

In this graph, the small multiples are generated by a scatter plot that displays the coin images as vertically stacked markers. The coins are also jittered horizontally to make the stack look more realistic. In addition to the coin stacks, the color of the label indicates which states have income levels above the national average. Although an infographic like this does not give you a very precise representation of the data, what it brings to the table is the wow factor for presentation!

TILE GRID MAP OF 2016 PRESIDENTAIL ELECTION

Figure 8 uses the SGPIE procedure instead of SGPLOT in the SAS macro function %SMALL_MULTIPLE to generate the small multiples.

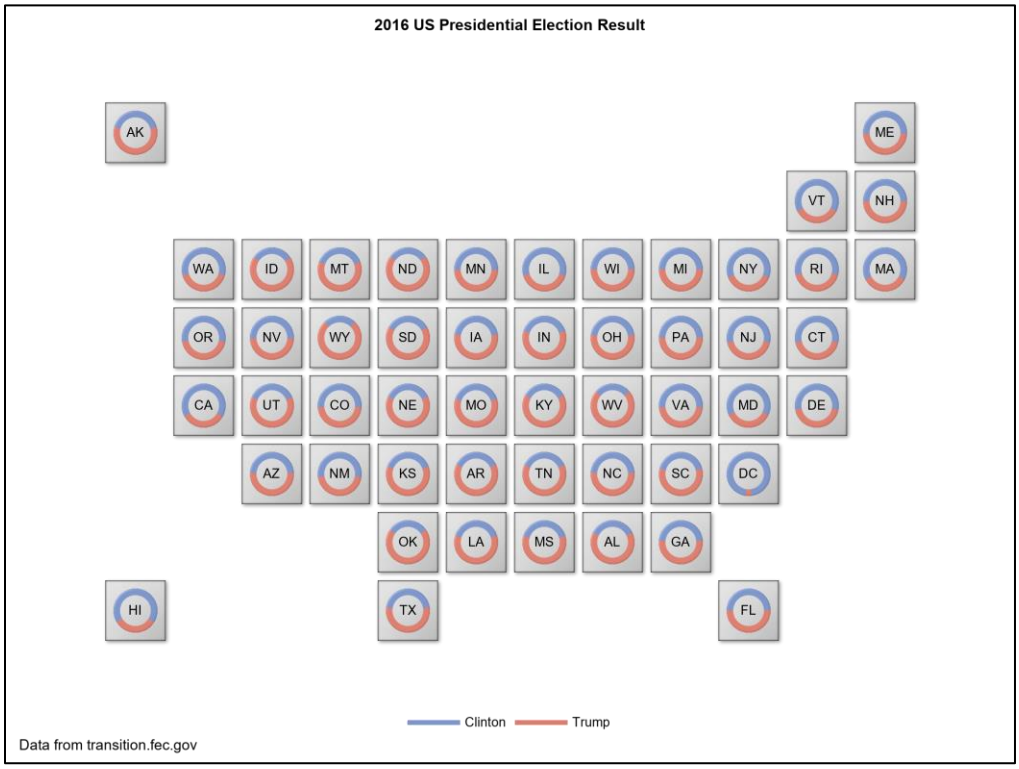

Figure 8. Tile Grid Map of the 2016 US Presidential Election

This example demonstrates that you can use any procedure that is based on ODS Graphics to produce the small multiples. This opens all kinds of possibilities for you as an ODS Graphics user.

TILE GRID MAP OF GDP GROWTH IN MAINLAND CHINA

The use of SGPLOT with a SCATTER statement to assemble the final tile grid map offers you great flexibility in terms of shading the tile background, adding skins, coloring the label, and using any ODS Graphics output for the small multiples. But if you do not need that ultimate flexibility, you can still create an impressive tile grid map with the SGPANEL procedure using considerably less code.

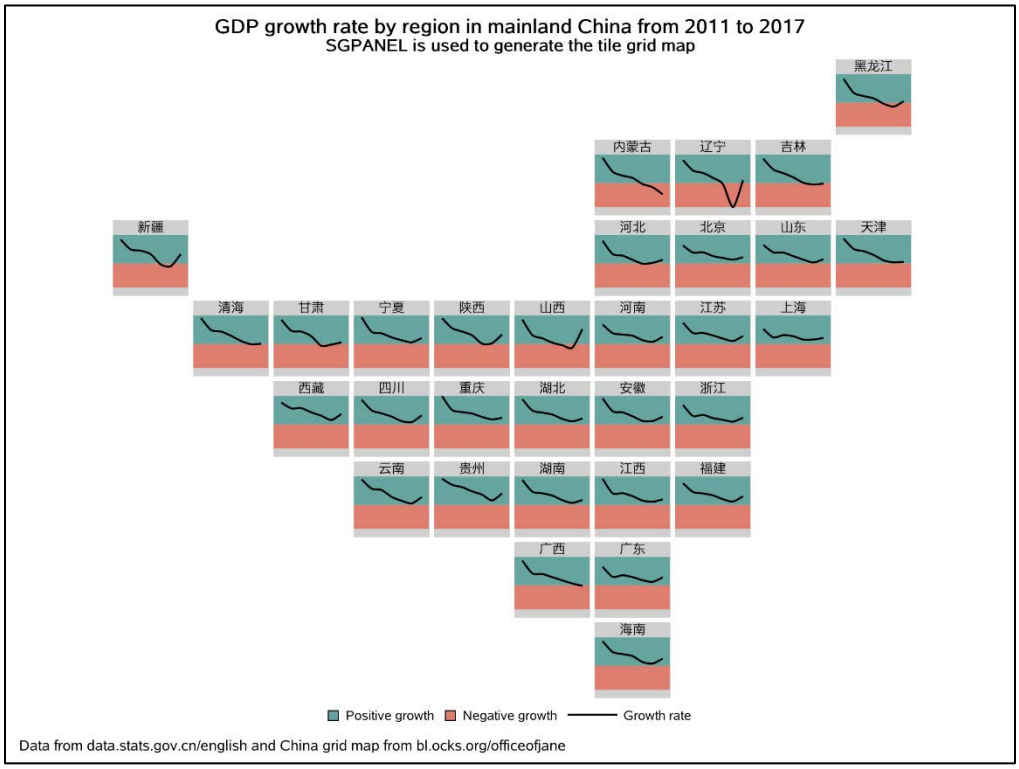

Figure 9. Tile Grid Map of GDP Growth in Mainland China

Figure 9 illustrates a tile grid map of mainland China to show the GDP growth rate by region. The output is created by the SGPANEL procedure with the small-multiple graphs generated by a block plot, two band plots, and a series plot. This example uses a China grid map found on the web (Pone, 2019). It demonstrates that you can display a tile grid map of any country or region if you can get your hands on the grid map data (PolicyViz).

CONCLUSION

The introduction of the SYMBOLIMAGE statement in SAS® 9.4M1 allows us to use any image output generated by ODS Graphics as a marker symbol in a subsequent graph. The techniques shown in this paper demonstrated that, with a combination of SAS language features and the ODS Graphics system, you can come up with an automated way to produce a very popular and fully customizable mapping tool: the tile grid map.

REFERENCES

Keating, Dan, and Park, Katie. 2015. "Raising the lowest wages." The Washington Post. Available at https://www.washingtonpost.com/graphics/national/minimum-wage/

Ribecca, Severino. 2018. "Chart Combinations: Tile Grid Maps." The Data Visualization Catalogue Blog. Available at https://datavizcatalogue.com/blog/chart-combinations-tile-grid-maps/.

Pong, Jane. 2019. "US grid maps." Jane Pong's Block. Available at https://bl.ocks.org/officeofjane/2c3ed88c4be050d92765de912d71b7c4.

Matange, Sanjay. 2017. "Decorative Infographs Using SAS®." *Proceedings of the SAS Global 2017 Conference*. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings17/SAS0315-2017.pdf

Li, Lingxiao. 2019. "Identify Seasonality and Trend with a Cycle Plot in SAS®." *Proceedings of the SAS Global 2019 Conference*. Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2019/3214-2019.pdf

PolicyViz. "Tile Grid Map." Blog Collections in PolicyViz. Available at https://policyviz.com/?s=tile+grid+map

ACKNOWLEDGMENTS

Thanks to my colleagues Prashant Hebbar, Dan Heath, and Ravi Devarajan, SAS Institute, for their help and comments on the paper.

CONTACT INFORMATION

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

Lingxiao Li
SAS Institute Inc.
919-531-5858
L.LI@SAS.COM