Decorative Infographs using SAS
Sanjay Matange, SAS Institute Inc.

ABSTRACT

The SAS 9.4 SGPLOT procedure is a great tool to create all types of graphs from Business Graphs to complex Clinical Graphs. The goal for such graph is to convey the data in a simple and direct manner with minimal distractions.

But often you need to grab the attention of a reader in the midst of a sea of data and graphs. For such cases, you need a visual that can stand out above the rest of the noise. Such visuals insert a “Decorative” flavor into the graph to attract the eye of the reader, and encouraging them to spend more time studying the visual. This presentation will discuss how you can create such attention grabbing visuals using the SGPLOT procedure.

INTRODUCTION

The SGPLOT procedure is your tool of choice to create graphs that effectively communicate the information to your readers in a clear and effective manner. You can create graphs from the simple bar chart to complex clinical graphs. For the most part, such graphs avoid unnecessary glitz or decorations, also referred to as “Chart Junk” by Edward Tufte.

In this presentation, we will depart from the quest for “Effective Graphs” and delve into the arena of decorative infographs: Graphs that utilize a decorative twist to grab the eye of the reader. Often, information is densely packed on the page of a magazine or brochure. Different articles on the page vie with each other for the attention of the reader. How does one stand out amongst the crowd? One way is by use of Decorative Infographs.

What, you may ask, is “Decorative Infographs”? Let us take a look at the two graphs shown in Figure 1 and Figure 2 below.

Figure 1 shows a traditional bar chart of the “Revenues by Year”. The chart uses a “Skin” to create the appearance of a cylindrical bar, but other than that it is very standard. The graph in Figure 2 displays the same data, but using a “real world” stack of coins to represent the revenue. On a page crowded with text and graphics, the graph in Figure 2 may stand out and grabs the attention of a reader. I call the graph in Figure 2 a “Decorative Infograph”.

In this graph, it is possible the stack of coins may distort the magnitude of the data a little bit, and the height of each stack may not be exactly proportional to the data. But those details are secondary in this display. The goal is to grab the attention of the reader.

In this presentation, we will see how you can create decorative infographs like the one in Figure 2 using the SGPLOT procedure. The graphs are created using data sets and standard procedure syntax. So, anything you do is scalable to different data with minimal customization. No annotation of any kind is used in the creation of the graphs discussed in this paper.
KEY FEATURE FOR CREATING DECORATIVE INFOGRAPHS

The key feature of the SGPLOT procedure that is used to create the decorative graphs discussed in this paper is the ability to use images as markers using the SYMBOLIMAGE statement. This statement was introduced with the SAS 9.40M1 release.

SymbolImage name=symbolname image="file-name" </ options>;

Supported options are:

- HOffset=offset. The horizontal offset to the anchor point to be placed at the (x, y) location.
- Rotate=number. Angle of rotation in degrees.
- Scale=number. Scale factor for the image. This is used to map to default size of the marker.
- VOffset=offset. The vertical offset to the anchor point to be placed at the (x, y) location.

We can use this statement to define a symbol with a name specified in the name=symbolname parameter using an image file on the file system specified in the image='file-name' parameter. Then, we can use this symbol to draw markers in the graph using its assigned name just like any other symbol, like "circle" or "circleFilled".

Normally, images found on the web or those taken using a smartphone camera are rectangular in shape, and every pixel in the image is opaque, like the example shown in Figure 3. The image shown in Figure 4 has been processed to make all the background pixels (not part of the object) as transparent. This can be done by using an image processing software such as Photoshop.

The "red" pixels from Figure 3 are transparent in Figure 4, so when the image is pasted on a background, the background will show through the transparent pixels. We will always use transparent images in these examples for reasons shown below.

![Figure 3 – Image with fully opaque background](image1)

![Figure 4 – Image with transparent background](image2)

Now, let us define a symbol called "coinOpaque" using the images from the file system using the image for the image in Figure 3. Then, we create a scatter plot from sashelp.class data using this as the symbol. This is shown in Figure 5 below.

```
proc sgplot data=sashelp.class noborder noautolegend;
    symbolimage name=coinOpaque image="&coinOpaque";
    scatter x=age y=height / markerattrs=(symbol=coinOpaque size=100);
    xaxis offsetmin=0.15 offsetmax=0.15;
    yaxis offsetmin=0.15 offsetmax=0.15;
run;
```

We do the same graph using the transparent image from Figure 4. This graph is shown in Figure 6 below.
Note in the graphs in Figure 5 and 6, markers are drawn for every observation in the data set. I have set the marker size to 100 pixels. The graph in Figure 5 uses the opaque image for the symbol. So we see the full rectangular image for each marker, and where the markers are close together, the markers are obscured by the background.

In the graph in Figure 6, we have used the transparent image for creating the symbol. Now, only the non-transparent pixels of the symbol are drawn. Whatever was in the graph before a symbol was placed shows through the transparent portion of the symbol. This creates a nice overlap between the coins, including the short stack of 3 coins in the middle.

Using transparent images as symbols for markers allows us to use these as shapes or masks or skins.

**USING MARKERS AS SHAPES**

When we use transparent images where the pixels outside the shape of the object in the image are transparent, as in Figure 6, we can use markers as shapes for our graphs. Let us review a couple of examples. Some options in the code below are thinned to fit the space. See “Resources” for link to download the program zip file.

**COIN STACK BAR CHART**

Let us go back to the Coin Stack Bar Chart to see how we can create this graph using the feature we described above. To start, we have a data set of Revenues by Year as shown in Figure 7. The bar chart shown in Figure 8 can be easily created using the VBAR statement of the SGPLOT procedure as follows:

```sas
PROC SGPLOT DATA=Bar NOORDER NOAUTOLEGEND;
  VBAR year / RESPONSE=resp DATASKIN=pressed DATALABEL;
   XAXIS DISPLAY=(NOTICKS NOLINE NOLABEL) INTEGER;
   YAXIS DISPLAY=(NOLABEL NOTICKS NOLINE) INTEGER GRID;
RUN;
```

Now, in order to use the image of the coin as a marker to create the coin stack, we need to use the SCATTER plot statement instead of the VBAR statement.

```sas
PROC SGPLOT DATA=bar NOORDER NOAUTOLEGEND;
  SCATTER X=year Y=resp / MARKERATTRS=(symbol=circlefilled size=60)
     FilledOutlinedMarkers DATASKIN=gloss;
  TEXT X=year Y=resp TEXT=resp / STRIP POSITION=top BACKLIGHT=0.75;
  XAXIS DISPLAY=(NOTICKS NOLINE NOLABEL) OFFSETMIN=0.15 OFFSETMAX=0.15;
  YAXIS DISPLAY=NONE OFFSETMIN=0.2 OFFSETMAX=0.2;
RUN;
```

Since we use the same data shown in Figure 7, we will only get one marker for each observation as shown in Figure 9. We also used the TEXT plot to display the response value on top of each marker per year. X and y-axis options are used to clean up the graph. The results are shown in Figure 9.
What we really need is a stack of markers for each year, each stack having the number of markers that add up to the revenue. So, first stack needs 8 markers at values of 1 to 8, and so on. So, we need to process the data to create multiple observations per year as shown in Figure 10.

```
data Coins;
  set bar;
  YVal=resp+1;
  do Val=1 to Resp by 1;
    output;
    YVal=.;
  end;
run;
```

In the code above, we also compute another variable YVAL that has a value just above the revenue for each year. Only one value per year is generated, and all other values are set to missing. The variable YVAL is used later to display the response value for each year above the stack.

We use this new data set shown in Figure 10 with the scatter plot to create the stack of markers as shown in Figure 11.

```
title 'Revenues (Millions) by Year';
footnote j=1 h=7pt italic 'Indian 2 Annas Coin from 1942';
proc sgplot data=Coins noborder noautolegend;
  symbolimage name=Indian image="&Indian";
  scatter x=year y=val / markerattrs=(symbol=Indian size=110);
  text x=year y=YVal text=resp / strip position=top backlight=0.75;
  xaxis display=(noticks noline nolabel) offsetmin=0.15 offsetmax=0.15;
  yaxis display=none offsetmin=0.2 offsetmax=0.2;
run;
```

Note, for this to work, the image of the coin needs to have a bit of a perspective. It is best the image itself has a perspective view. If not, you could distort the image a bit to suit your needs.
Finally, we use the JITTER option in the SCATTER statement to create stacks that are not perfectly stacked by area a bit more realistic as shown in Figure 13. It is also necessary to create the number of observations in the stack that correspond to the apparent thickness of the coin in the image. Note, the Somalian Silver Shilling is a bigger and thicker coin. Using that coin, we need more spacing between each marker to look right, as shown in Figure 14.

COLLEGE READINESS GRAPH

Let us review another example of using shapes to create decorative infographs.

Recently, there was a request to create a graph comparing College Readiness of students from a local school with state average. The data for the graph is shown in Figure 15 below. From a perspective of providing the most effective way to make the comparison, the cluster grouped graph shown in Figure 16 below would be best. It is very easy to compare the performance of students from the local schools with state averages side by side.
Having addressed the issue of “effective graphs” let us focus on the needs of this particular user to create a graph that would catch the eye of the reader. The desired graph is shown in Figure 18. How would we create a graph like this using the technique discussed so far?

The first step is to note that the response values are in percent, from 0 to 100. So, we can create an observation for every 10% of the response value. An observation is created for “This School” value and one for “State” value. If the value for the marker is a multiple of 10%, we will display it with the full colored marker (group value A1 or B1). If the marker represents a portion of the 10%, then we use the partial markers with group value of A2 and B2. For the values > response and < 100%, we use the gray marker with group value of C1.

The icons for the symbols for each of the group values A1, A2, B1, B2 and C1 are shown in Figure 17. As in the previous example, we use five SYMBOLIMAGE statements to define the symbols for each group. Now, we use the STYLEATTRS statement to use these symbols as the group symbols to draw the graph. A separate XAXISTABLE statement is used to display the headers and TEXT plot is used to display the values on left, middle and right of the graph. The SGPLOT code snippet is shown below. See “Resources” for link to download the program zip file. The %rgbhex macro created by Perry Watts is used for setting colors.

```
proc sgplot data=Readiness4 noborder noautolegend;
symbolimage name=A1 image="&file1";
symbolimage name=B1 image="&file2";
symbolimage name=C1 image="&file3";
symbolimage name=A2 image="&file4";
symbolimage name=B2 image="&file5";
styleattrs datasymbols=(A1 A2 C1 B1 B2)
datacontrastcolors=(%rgbhex(254, 145, 104)
%rgbhex(130, 109, 146));
xaxistable text / x=xlbl1 colorgroup=text location=inside position=top;
scatter y=level x=val / group=group markerattrs=(size=40);
text x=x0 y=level text=valueLblA / textattrs=(color=%rgbhex(254,145,104));
text x=x270 y=level text=valueLblB /textattrs=(color=%rgbhex(130,109,146));
text x=x130 y=level text=level / splitchar='.' splitpolicy=splitalways;
xaxis display=none max=2.7 offsetmin=0.05 offsetmax=0.0;
yaxis display=none reverse splitchar='.' offsetmin=0.1
fitpolicy=splitalways splitjustify=center;
run;
```

**USING MARKERS AS MASK AND SHAPE**

When we use transparent images where the pixels outside the shape of the object in the image are transparent, as in Figure 6, we can use markers as shapes for our graphs. We did that in the graphs shown in Figures 13, 14 and 18.
When the outer pixels are opaque (usually of a color that matches the graph background) and the inner pixels are transparent, then the marker can be used as a "Mask". Figure 19 shows an image of a cup, where the pixels inside the cup are all transparent. When this image is pasted on a red background, only the red pixels in the middle show through, while the other pixels are overdrawn by the opaque pixels of the cup image, as shown in Figure 20. We can use this technique to shape the features of a graph.

Figure 19 – Image used as a mask

Figure 20 – Image mask on a red background

COFFEE RECIPES GRAPH

Let us use the "mask" technique to create a graph showing the ingredients of cup of coffee. In this example, I want to show you the recipes for 4 different popular coffee beverages, Expresso, Macchiato, Café Latte and Café Mocha.

We start with a simple bar chart showing the ingredients and proportions for each beverage. Starting with Expresso only, for Macchiato, we mix Expresso plus an equal portion of Milk Foam. Then, for Café Latte, we use 1 part Expresso, 2 parts Steamed Milk topped with 1 part Milk Foam. Lastly for Café Mocha, we use 1 part Expresso, 1 part Chocolate, 1 part Steamed Milk topped with 1 part Milk Foam.

We create a data set as shown in Figure 21. Each beverage (Name) uses the required ingredients (Group) with a value (0.25). Starting from Y=0, these are added to create the Low and High values that will be used by a HighLow plot to draw the bars as shown in Figure 22.

```
title j=l h=1 'Coffee Recipes';
proc sgplot data=Coffee dattrmap=attrmap;
    highlow x=name low=low high=high / group=group type=bar nooutline
        barwidth=0.7 name='a' attrid=Coffee;
    text x=name y=mid text=group / backlight=0.4 textattrs=(color=white);
    xaxis display=( nolabel noticks) offsetmin=0 offsetmax=0.12;
    yaxis display=none min=0 max=1 offsetmin=0.15 offsetmax=0.42;
run;
```

Figure 21 – Data for Coffee Graph

Figure 22 – High Low graph of ingredients

Figure 23 – Graph of coffee recipes.
Now, comes the key step that turns this ordinary graph into a cool infographic shown in Figure 23. We use the image of the cup as a mask to sculpt the simple bar chart into one that looks like a coffee cup.

To do this, we use the image of the coffee cup as a symbol using the SYMBOLIMAGE statement. Then, we simply draw a scatter plot with this symbol as its marker right on top of each highlow bar. Recall that in this case, the image is fully opaque, with white pixels along the outside and brown pixels for the part of the cup. However, the pixels in the middle of the cup are transparent. So, the stacked bars previously drawn show through this middle part of the markers.

As a final touch, we display a small marker using an image of steaming vapors on top of the coffee. The code snippet is shown below. Note the use of an attribute map for the color of each ingredient and “Backlight” option on the text. See “Resources” for link to download the program zip file.

```sas
title j=1 h=1 'Coffee Recipes';
proc sgplot data=Coffee dattrmap=attrmap pad=(bottom=20pct);
  symbolimage name=Cup image="&file1" / voffset=0.15;
  symbolimage name=Steam image="&file2" / voffset=0.0;
  highlow x=name low=low high=high / group=group type=bar nooutline
    barwidth=0.7 name='a' attrid=Coffee;
  scatter x=name y=y / markerattrs=(symbol=Cup size=100);
  scatter x=name y=ys / markerattrs=(symbol=Steam size=20);
  text x=name y=mid text=group / backlight=0.4 textattrs=(color=white);
  xaxis display=( nolabel noticks) offsetmin=0.12 offsetmax=0.12;
  yaxis display=( nolabel noticks noline) min=0 max=1 offsetmin=0.15 offsetmax=0.44;
run;
```

**USING MARKERS FOR MASK, SHAPE AND SKIN**

For the final example, we will use image markers as masks, skins and shapes as shown in Figure 24. For this example, we will create a graph of Soda Sales by Region. The number are made up for illustration only. The idea is to build a decorative infograph of the sales.

```sas
Obs Region Value  Low  Mid  Bubble Y YLbl
1  USA  0.7  0.01  0.35  0.63  0  0.1
2  Canada  0.6  0.01  0.30  0.54  0  0.1
3  Mexico  0.5  0.01  0.25  0.45  0  0.1
4  Brazil  0.6  0.01  0.30  0.54  0  0.1
5  Chile  0.3  0.01  0.15  0.27  0  0.1
```

**Figure 24 – Mask, Skin and Shape**

**Figure 25 – Data for graph**

The first step in creating this graph is to create a bar chart of soda sales by region, as shown in Figure 26. The code for this is shown below. Instead of using a VBAR statement, we have used the HighLow statement. The main reason is to be able to overlay markers later and also use the HighLabel option.

```sas
title j=1 h=1 'Soda Sales by Region (in Millions $)';
proc sgplot data=coke noborder noautolegend nocycleattrs;
  highlow x=Region low=low high=high / type=bar highlabel=high;
  xaxis display=( nolabel noticks) offsetmin=0.12 offsetmax=0.12;
  yaxis display=( nolabel noticks noline) min=0 max=1 offsetmin=0 offsetmax=0;
run;
```
The next step is to sculpt each bar in the shape of the soda bottle. Here I have used an image of a classic coke bottle as a mask. This mask is shown in the left in Figure 24. The image has opaque white pixels everywhere, except in the middle of the bottle defining the shape. These pixels in the middle are fully transparent. So, when a marker using this image as a symbol is displayed on top of the bar, the color of the bar shows through the transparent pixels, but the rest of the bar is masked. This results in the graph shown in Figure 27. The code snippet is shown below:

```plaintext
title j=l h=1 'Soda Sales by Region (in Millions $)';
proc sgplot data=coke noborder noautolegend nocycleattrs;
  symbolimage name=In image="&In" / voffset=0.5 hoffset=0.01;
  highlow x=Region low=low high=high / type=bar nooutline;
  scatter x=Region y=y / markerattrs=(symbol=in size=200);
  xaxis display=( nolabel noticks) offsetmin=0.12 offsetmax=0.12;
  yaxis display=(nolabel noticks noline) min=0 max=1 offsetmin=0 offsetmax=0;
run;
```

Figure 26 – Bar Chart

Figure 27 – Bar Chart with mask

Now, finally, we use the middle image of the coke bottle in Figure 24 as a "skin". This image has the reverse transparency as the mask. The outer pixels are transparent. The pixels making up the shape of the bottle with the highlights and reflections in the glass are opaque. We use another scatter plot to layer this image on top of the previous shape, with a partial transparency. The last layer acts like a skin. The result is shown in Figure 28.

Figure 28 – Bar chart with mask and skin.
If you look closely, you will also see some bubbles in the liquid, placed there using image of the bubbles in Figure 24. The SGPlot code snippet is shown below. See “Resources” for link to download the program zip file.

title j=l h=l 'Soda Sales by Region (in Millions $)';
proc sgplot data=coke noborder noautolegend nocycleattrs;
  symbolimage name=In image=&In / voffset=0.5 hoffset=0.01;
  symbolimage name=Out image=&Out / voffset=0.5;
  symbolimage name=bubbles image=&Bubbles;
  scatter x=Region y=y / markerattrs=(symbol=Out size=200) transparency=0;
  highlow x=Region low=low high=high / type=bar nooutline barwidth=0.7
    fillattrs=(color=cx3f0f00);
  scatter x=Region y=y / markerattrs=(symbol=In size=200);
  scatter x=Region y=y / markerattrs=(symbol=Out size=200) transparency=0.4;
  scatter x=Region y=bubbles / markerattrs=(symbol=bubbles size=20);
  highlow x=Region low=low high=high / type=bar nooutline barwidth=0.7
    fillattrs=(transparency=1) highlabel=hi labelattrs=(color=cx5f3f00);
  reline 0.2 0.4 0.6 0.8 / transparency=0.8;
  xaxis display= nolabel noticks noline offsetmin=0.12 offsetmax=0.12;
  yaxis display= nolabel noticks noline min=0 max=1
    offsetmin=0 offsetmax=0 values=(0 to 1 by 0.2);
run;

CONCLUSION

The SGPlot procedure has new features with SAS 9.4M1 that support creation of decorative infographs. In particular, we can use the SYMBOLIMAGE statement that allows us to use any image as a symbol for markers in the graph. When transparent images are used as markers, this allows us to use these as shapes or masks or skins.

Creative combination of markers as shapes, masks or skins can be used to make decorative graphs using procedure syntax and data sets. No annotation is required. Now, when you feel the need for a visual that is outside the box, the SGPlot procedure can get you there.

RESOURCES

The PDF file of this paper and the SAS code for all the programs is available at:


REFERENCES

Graphically Speaking by Sanjay Matange. http://blogs.sas.com/content/graphicallyspeaking

ACKNOWLEDGMENTS

Thanks to Perry Watts for the %rgbhex macro for creating a CX formatted string from RGB values.

RECOMMENDED READING


CONTACT INFORMATION

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

Sanjay Matange  
SAS Institute Inc., 
100 SAS Campus Dr.  
Cary, NC 27513  
Sanjay.Matange@sas.com  
http://blogs.sas.com/content/graphicallyspeaking

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration.

Other brand and product names are trademarks of their respective companies.