

Paper 5149-2020

## Stepping Up Your SAS® ODS Graphics Game

Sara Richter, Professional Data Analysts

### ABSTRACT

Do you want to create eye-catching and informative data displays, but don't have access to SAS® Viya® or SAS® Visual Analytics? Not to worry! With a little bit of extra work and ingenuity, you can use SAS® Output Delivery System (ODS) Graphics, available in Base SAS®, to turn mediocre data displays into high-impact communication tools. This paper provides ideas for combining tables, charts, and formatting elements to step up your ODS Graphics game. It demonstrates how SGPLOT procedure output can be combined with REPORT procedure output into a single visual element and gives examples of how SG annotate functions can enhance standard graphs. These code examples will empower the applied statistician to create more advanced data visualization elements.

### INTRODUCTION

Effective data visualizations can make it easy to understand trends, variability, and patterns that may not be obvious in tabular data displays. They are also great communication tools for conveying data analysis findings. Creating clear, easy-to-read graphics is becoming an expectation for every analyst, and at the same time fulfilling customer data visualization requests can be challenging. SAS has provided several tools to meet these ever-increasing data visualization needs. For instance, SAS Viya and SAS Visual Analytics are great visualization tools, though they require a separate license from Base SAS. What is a Base SAS user to do?

Thankfully, Base SAS offers powerful, flexible graphing tools within the ODS Graphics procedures, also referred to as Statistical Graphics (SG) procedures or SG graphics. These SG procedures are capable of basic graphing as well as layering different graph types, formatting graph elements, and creating multi-panel graphs.

Even with the many features available within SG procedures, there are times when your data visualization goal cannot be fully achieved. In these cases, there are ways to enhance the ODS Graphics, such as through SG annotate facilities, applying custom formatting and attribute maps, and combining SG output with PROC REPORT output. **Today's analyst** will benefit from having these tools in their data visualization toolbox.

### OVERVIEW

This paper will demonstrate methods to enhance the SG procedures through real world examples. It is geared toward SAS users with a basic knowledge of ODS Graphics. In this paper, I will cover the following topics:

- Creating your graph: A brief review of ODS Graphics and SG procedures
- How to add the graphical "extras": **The SG annotate facilities**
- Where to add the graphical "extras": Choosing the draw space
- Applied examples: Recreations of the Ambulatory Glucose Profile
- A step beyond: Combining output from graphs and tables

Select code is included throughout the paper; full code can be found in the Appendix.

## CREATING YOUR GRAPH: A BRIEF REVIEW OF ODS GRAPHICS AND SG PROCEDURES

ODS Graphics were introduced in SAS® 9.2 as part of SAS/GRAPH®. In SAS 9.3, the ODS Graphics procedures were moved to Base SAS. In SAS 9.4, procedures were added, and enhancements made to the existing procedures. A listing and brief description of the SG procedures available can be found in Table 1. Most of the SG procedures create graphs including the SGPLOT, SGSCATTER, SGPANEL, SGMAP, and SGPIE procedures. Two other procedures, SGRENDER and SGDESIGN, are used to render graphs created with the TEMPLATE procedure. These rendering procedures, PROC SGMAP, and PROC SGPIE are beyond the scope of this paper. For full information on SG procedures, consult the SAS® 9.4 *ODS Graphics: Procedures Guide, Sixth Edition* available at: <https://documentation.sas.com/api/docsets/grstatproc/9.4/content/grstatproc.pdf>.

Procedures that create graphs:		
Procedure	Released in SAS...	Description
SGPLOT	9.3	Creates a single-cell graph and supports basic plots (e.g. line, scatter, band, vector, etc.), fit plots (e.g. regression, loess, ellipse, etc.), distribution plots (e.g. histogram, box, etc.), and categorization plots (e.g. bar charts, dot, etc.) either alone or overlaid
SGSCATTER	9.3	Creates a paneled graph of scatter plots, including a scatterplot matrix, and can overlay fit plots and ellipses
SGPANEL	9.3	Creates a panel of graph cells with each cell containing one plot for each response of a classification variable; supports most of the plot options available in SGPLOT
SGMAP	9.4M5	Creates maps with data overlays (e.g. markers, bubbles, text, etc.)
SGPIE	9.4M6 (pre-production)	Creates pie and donut charts
Procedures that render graphs:		
SGRENDER	9.3	Produces graphs defined in templates from the Graphical Template Language
SGDESIGN	9.3	Produces graphs from user defined ODS Graphics Designer files created in the ODS Graphics Designer application

**Table 1: Description of SG Procedures**

### COMPARISON OF ODS GRAPHICS AND SAS/GRAPH

Even though ODS Graphics started out under SAS/GRAPH, ODS Graphics and SAS/GRAPH are now totally separate graphing tool kits. Both sets of procedures offer customizable graphing tools; however, there are some major differences outlined in Table 2 below. Of note, each has its own set of annotate facilities, allowing you to add embellishments like arrows, shapes, text labels, images, and other annotations to your graphs; however, the annotate facilities are not interchangeable between ODS Graphics and SAS/GRAPH. It is also important to note that options specified on the global GOPTIONS statement have no effect on SG procedures.

<u>ODS Graphics</u>	<u>SAS/STAT</u>
<ul style="list-style-type: none"> <li>• Available in Base SAS</li> <li>• “SG” procedures</li> <li>• Output based on templates and the Output Delivery System</li> <li>• SG ANNOTATE facilities available</li> <li>• ODS GRAPHICS to control graphical environment</li> <li>• Options within SG procedures and attribute maps control visual aspects of the graph</li> <li>• Scales markers and fonts by default</li> </ul>	<ul style="list-style-type: none"> <li>• Requires a separate license</li> <li>• “G” procedures</li> <li>• Output based on devices</li> <li>• ANNOTATE facilities available</li> <li>• GOPTIONS to control graphical environment</li> <li>• Global statements (AXIS, PATTERN, SYMBOL, LEGEND) control visual aspects of the graph</li> <li>• Does not support marker or font scaling</li> </ul>

**Table 2: Comparison of ODS Graphics vs SAS/STAT**

## HOW TO ADD THE GRAPHICAL “EXTRAS”: THE SG ANNOTATE FACILITIES

Before diving into examples, I will begin with a brief overview of the SG annotate facilities. For more details, please refer to the “Annotating ODS Graphs” section of the *SAS® 9.4 ODS Graphics: Procedures Guide, Sixth Edition*.

In brief, the SG annotate facilities allow you to add embellishments like arrows, shapes (lines, rectangles, ovals, etc.), text labels, images, and other annotations to your graphs. To use the annotate facilities, you must first create an SG annotate data set, then you must apply that data set to an SG procedure like SGPLOT, SGSCATTER, or SGPANEL.

**TIP: Make your graph as close as possible to the final product using the features available in SG graph capabilities, then use the ANNOTATE facility to add the “extras”.**

Annotate data sets are created within a DATA step and use specific keywords to create the annotations. Each observation in an annotate data set represents one annotation on the graph as shown in Display 1, except for the custom polygon annotations which need multiple observations. There are two ways to create observations within annotate datasets: annotate functions and, starting in 9.4M1, annotate macros. Annotate macros are shortcuts for calling annotate functions and are an easier way to specify the annotations. A listing of the available SG annotations is presented in Table 3 below. Heath (2016) provides a more thorough summary of both versions.

Annotation to add	SG annotate function	SG annotate macro
Add text	TEXT	%SGTEXT
	TEXTCONT	%SGTEXTCONT
Insert an image	IMAGE	%SGIMAGE
Draw an arrow	ARROW	%SGARROW
Draw a geometric shape	LINE	%SGLINE
	RECTANGLE	%SGRECTANGLE
	OVAL	%SGOVAL
Draw a custom polygon	POLYGON	%SGPOLYGON
	POLYLINE	%SGPOLYLINE
	POLYCONT	%SGPOLYCONT

**Table 3: SG Annotations with Corresponding Annotate Functions and Macros**

To use the annotate functions, you must use the FUNCTION statement in a DATA step to define the annotation you want to add. Depending on the annotation, you can specify labels, colors, starting and ending coordinates, along with many other features.

To call the SG annotate macros, you must first compile the macros by calling %SGANNO. Then all of the SG macros are available to use for the rest of that SAS session. The parameters within the macros are used to set the labels, colors, starting and ending coordinates, etc. and can be specified in any order.

The annotate functions and macros can be used to create the same annotations and they can both be used in the same DATA step. The example code below uses the annotate function TEXT and the %SGTEXT annotate macro in the same DATA step to create the observations in Display 1:

```
DATA MyAnnoData;
  FUNCTION="Text";
  LABEL="This is my example text from an annotate function.";
  X1=10;
  Y1=95;
  ANCHOR="TOPLEFT";
  WIDTH=100;
OUTPUT;
%SGANNO;
%SGTEXT(
  LABEL="This is my example text from the SGTEXT macro.",
  X1=10,
  Y1=85,
  ANCHOR="TOPLEFT",
  WIDTH=100);
RUN;
```

**TIP:** Getting the X- and Y- positioning correct involves trial and error. A good way to start is by creating a mock-up of the data visualization you want in another program, like Microsoft Excel, and then re-creating the graph in SAS.

The resulting data set, MyAnnoData, has two observations and looks like this:

	FUNCTION	LABEL	X1	Y1	ANCHOR	WIDTH
1	Text	This is my example text from an annotate function.	10	95	TOPLEFT	100
2	TEXT	This is my example text from the SGTEXT macro.	10	85	TOPLEFT	100

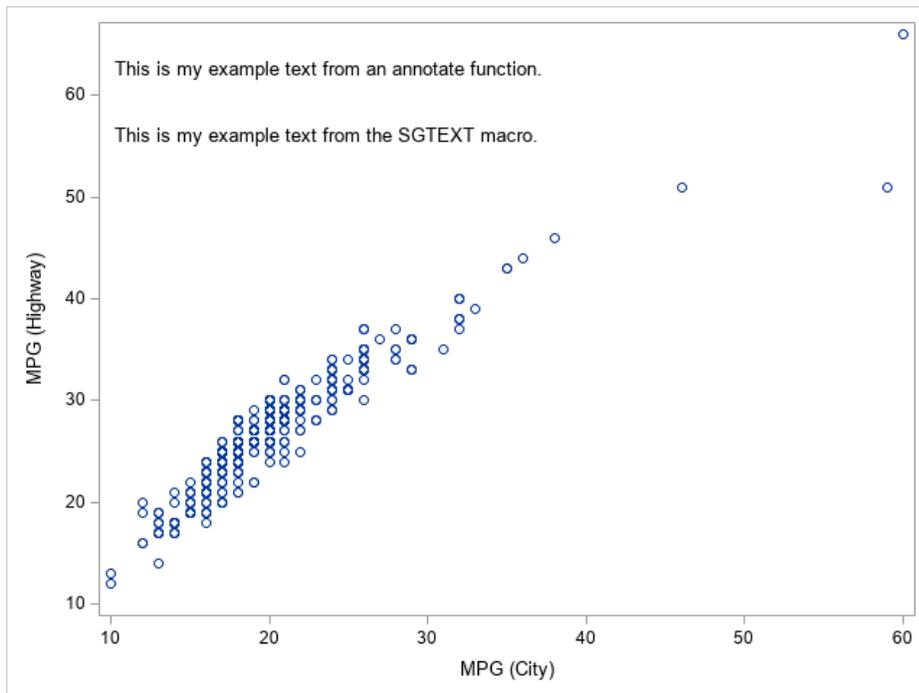
**Display 1: Screenshot of the MyAnnoData Data Set**

**TIP:** SAS adds the annotations in the order they appear in the SG annotation data set. If some annotations get covered up by other annotations, try re-ordering the annotation data set. You can also specify the parameter LAYER=FRONT to bring an annotation to the top layer.

Once the annotate dataset is created, you apply it to an SG procedure by specifying the annotate data set you created in the SGANNO option:

```
PROC SGPLOT DATA=SASHELP.Cars SGANNO=MyAnnoData;
  SCATTER X=MPG_City Y=MPG_Highway;
RUN;
```

The resulting annotations can be seen in Figure 1. Even though both methods can create the same annotations, my personal preference is to use the annotate macros.



**Figure 1: Example of Annotate Text Function and Macro**

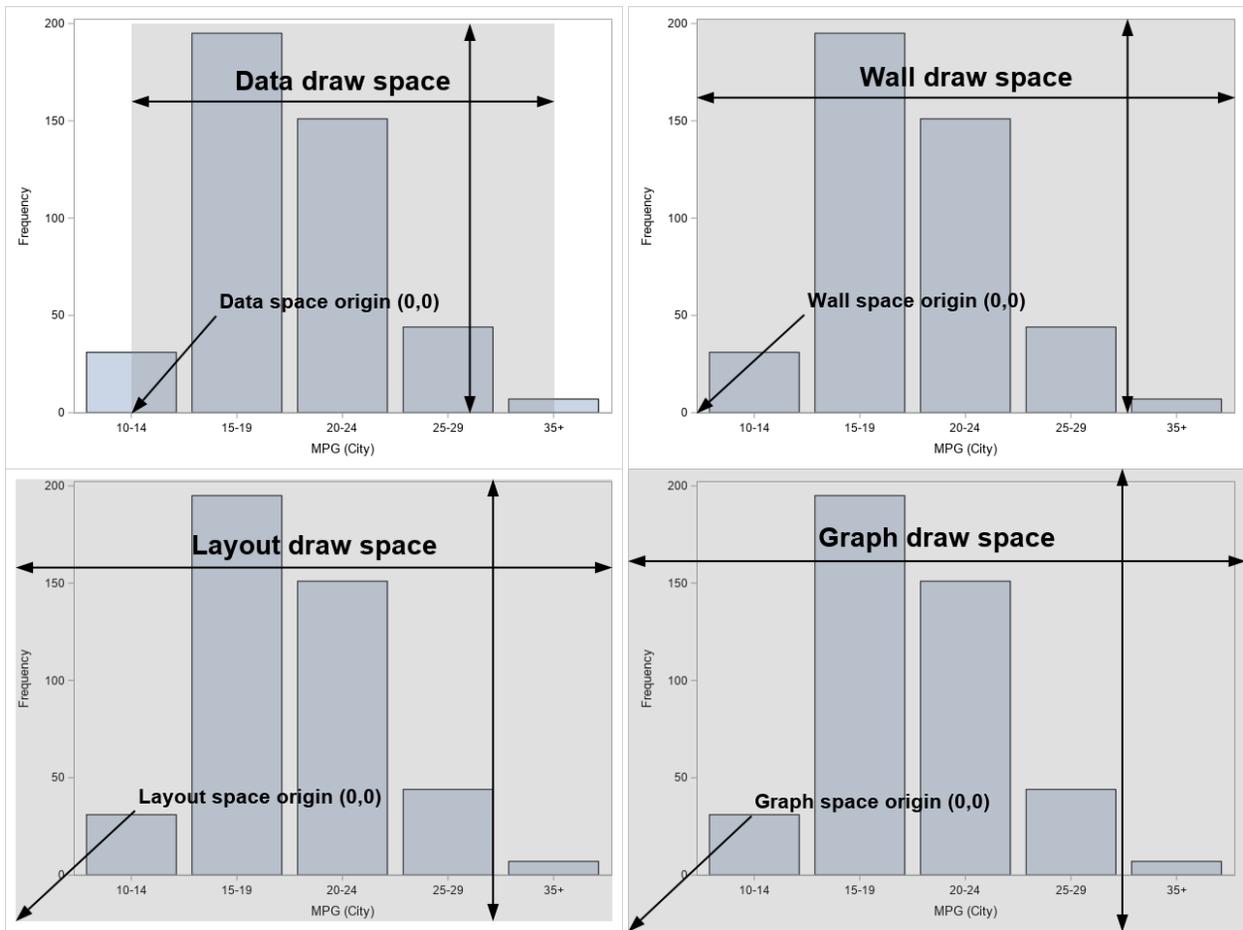
## WHERE TO ADD THE “EXTRAS”: CHOOSING THE DRAW SPACE

In addition to knowing what extra annotations you would like to add, you need to know where you want to add them. Depending on where you want the graphical embellishments, you may need to alter the draw space of the graph and the amount of padding surrounding the graph.

### THE DRAW SPACES

The draw space is the area SAS is using to create the graph and any annotations. The possible draw spaces include data, wall, layout, and graph. All draw spaces can be specified in pixels or percentages; the data draw space can also be defined by values on the plot. The origin of the drawing is the bottom left corner of the draw space. Figure 2 shows the various draw spaces shaded in grey with their origins.

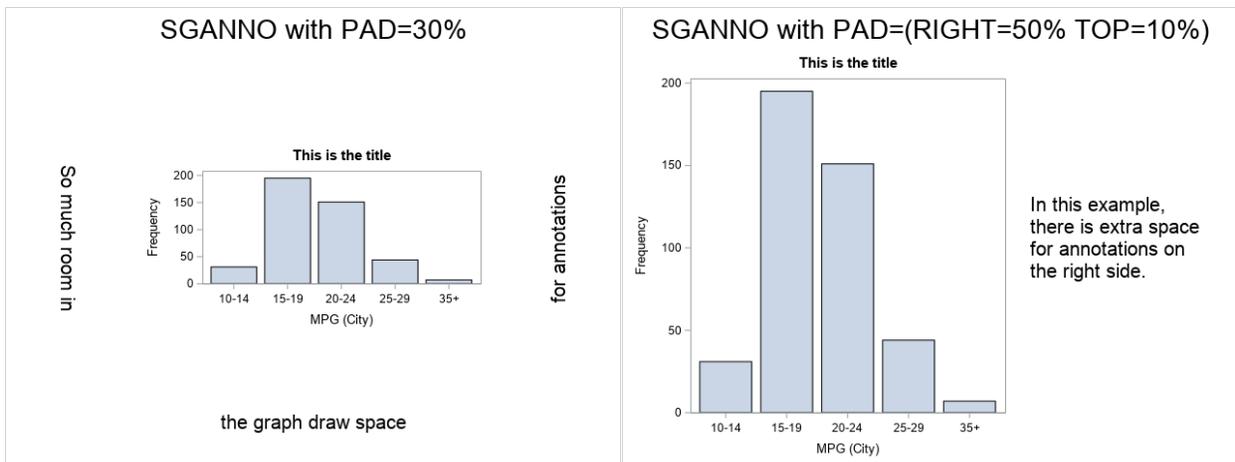
The default draw space is the layout draw space and the default unit is percent. Draw spaces are specified with their units as a single keyword, e.g. LAYOUTPERCENT. The draw space can be changed for both axes using the DRAWSPACE= option; or for the X- and Y- dimensions individually using the XSPACE= and YSPACE= options, respectively.



**Figure 2: Draw Space Examples with Origins**

### PADDING THE GRAPH

By default, SAS produces graphs with little or no padding around them. This means that SG procedures do not reserve any room in the areas around the graph for the annotations, they print the annotations on top of the graph. In cases where you want to add annotations around the graph (not on top of it), you must use the PAD option of the PROC SGPLOT to reserve space for annotations around the graph, as in Figure 3 below. Equal padding can be added around the entire graph, as in Figure 3a, or to any/all sides of the graph as in 3b.



**Figure 3a: Equal Padding for Each Side**

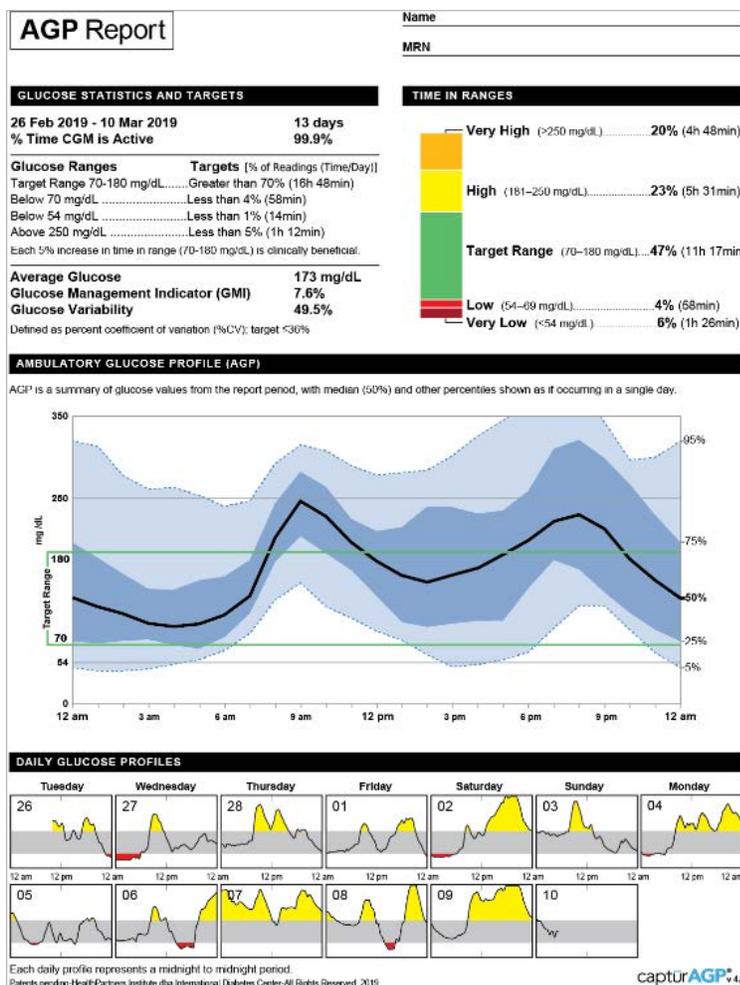
**Figure 3b: Varying Padding Amounts**

## APPLIED SG ANNOTATE EXAMPLES: RECREATIONS OF THE AMBULATORY GLUCOSE PROFILE

Now that we have reviewed ODS graphics and introduced the SG annotate facilities and draw spaces, we give some applied examples showing how SG annotations can enhance ODS Graphics.

### THE DATA VISUALIZATION GOALS

The following examples are recreations of the Ambulatory Glucose Profile (AGP), a one-page clinical report showing blood glucose trends from continuous glucose monitoring data. This report is meant to reflect trends in blood glucose values over the last two weeks to help patients and physicians with shared decision-making in diabetes care.



The full report and content descriptions can be found at <http://www.agpreport.org/agp/>. An example report from that website is shown in

Figure 4 to show the 'final product' we are trying to achieve in the examples.

For the purposes of this paper, the specific content of the report is not important – these examples are meant to demonstrate how the SG annotate facilities can enhance the ODS Graphics. The code will be applicable to many situations.

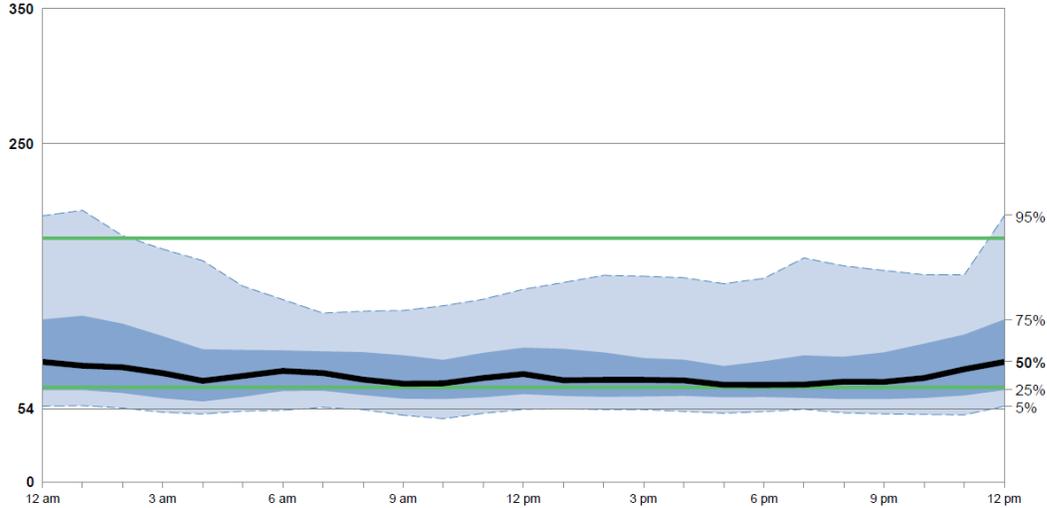
Example 1 will cover the line graph; example 2 will cover the stacked bar chart; and example 3 will cover the "Glucose Statistics and Targets." For each example, the desired enhancement is described and the before and after graphs shown. The data used in these examples is simulated.

Figure 4: The AGP Report Taken from agpreport.org

### EXAMPLE #1: ENHANCING THE Y-AXIS OF A SERIES PLOT

This first example recreates the Ambulatory Glucose Profile graph from the middle of the AGP report. This graph shows aggregated trends over a 24-hour period. The SG PLOT procedure with SERIES, BAND, and REFLINE statements produced a graph similar to what we need (see Figure 5), however the y-axis needs some additional annotations including y-axis labels and extended green bars (see Figure 6).

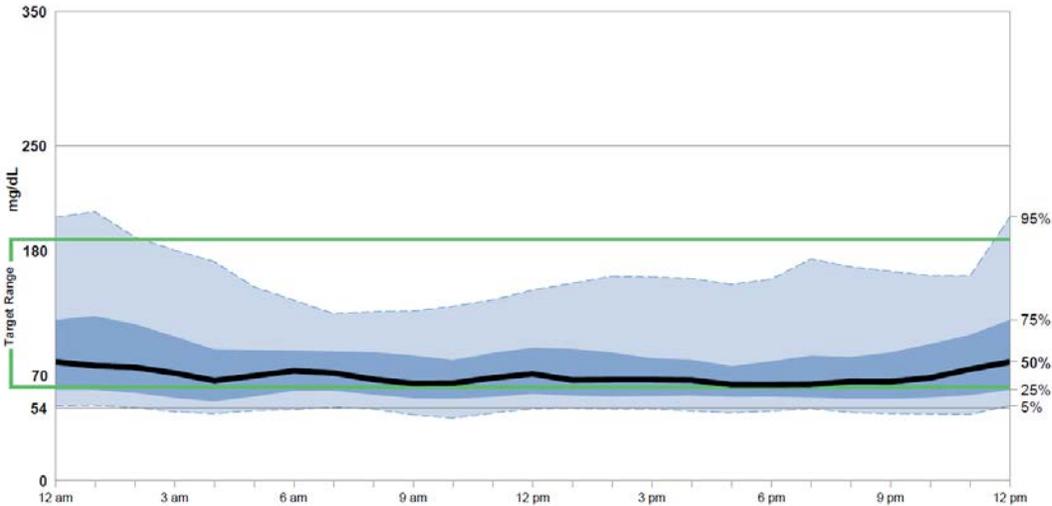
Result from PROC SGPLOT before annotations:



**Figure 5: The AGP Graph from PROC SGPLOT Before SG Annotation**

The SG annotate goal: Add formatting enhancements to the y-axis including extending the green lines into the left side layout space, modifying the y-axis label placement for the 70 and 180 lines, and adding vertical text labels.

Result from PROC SGPLOT after annotations:



**Figure 6: The AGP Graph from PROC SGPLOT with SG Annotation**

How to approach it: To extend the green bars, we will need to draw two horizontal green lines (one at 70 mg/dL and one at 180 mg/dL) and one vertical green line connecting the horizontal green lines. We will then need to add four text annotations – “70”, “180”, “Target Range”, and “mg/dL”. All of these annotations fit within the LAYOUT space and are compact enough that we do not need to pad the graph.

Full code for the graph, including the options specified in the <OPTIONS> placeholders below can be found in the Appendix. The SG annotation code to accomplish each of the needed embellishments is provided below along with an abbreviated version of the PROC SGPLOT code:

```
1 /* Compile SG annotate macros and then create the SG annotate data set */
   %SGANNO;
```

```

DATA AGPAnno;
  /* Add green lines that extend into the layout space */
  /* Adds the lower horizontal green line */
  2 %SGLINE(X1=3, Y1=28, X2=8, Y2=28, LINECOLOR="GREEN", LINETHICKNESS=8);
  /* Adds the upper horizontal green line */
  %SGLINE(X1=3, Y1=56, X2=8, Y2=56, LINECOLOR="GREEN", LINETHICKNESS=8);
  /* Adds the vertical green line */
  %SGLINE(X1=3, Y1=28, X2=3, Y2=56, LINECOLOR="GREEN", LINETHICKNESS=8);
  /* Add the 'Target Label' text over the green line */
  %SGTEXT(LABEL="Target Range", X1=3, Y1=44,
  3 ANCHOR="CENTER", ROTATE=90, FILLCOLOR="WHITE",
  TEXTCOLOR="BLACK", TEXTFONT="Arial", TEXTSIZE=7, TEXTWEIGHT="BOLD");
  /* Adjust y-axis labels (70 & 180) so they sit inside the green lines */
  %SGTEXT(LABEL=" 70", X1=6.5, Y1=28, ANCHOR="BOTTOMRIGHT",
  TEXTCOLOR="BLACK", TEXTFONT="Arial", TEXTSIZE=8, TEXTWEIGHT="BOLD");
  %SGTEXT(LABEL="180", X1=6.5, Y1=56, ANCHOR="TOPRIGHT",
  TEXTCOLOR="BLACK", TEXTFONT="Arial", TEXTSIZE=8, TEXTWEIGHT="BOLD");
  /* Add the 'mg/dL' label */
  %SGTEXT(LABEL="mg/dL", X1=3, Y1=65,
  ANCHOR="CENTER", ROTATE=90, FILLCOLOR="WHITE",
  TEXTCOLOR="BLACK", TEXTFONT="Arial", TEXTSIZE=8, TEXTWEIGHT="BOLD");
RUN;
  /* Create a format for the x-axis */
PROC FORMAT;
  4 VALUE XAxisFMT 0,24="12am" 3="3am" 6="6am" 9="9am" 12="12pm" 15="3pm"
    18="6pm" 21="9pm" OTHER=" ";
RUN;
  /* Use ODS GRAPHICS statement to set the size of the graph and to suppress
  the graph border */
  ODS GRAPHICS/ RESET=ALL WIDTH=7.5in HEIGHT=3.75in BORDER=OFF;
  /* Generate the plot */
PROC SGPLOT DATA = AGPData SGANNO=AGPAnno NOAUTOLEGEND;
  FORMAT Hour XAxisFMT.;
  5 BAND X=Hour LOWER=AGP5thPercentile UPPER=AGP95thPercentile/<OPTIONS>
  BAND X=Hour LOWER=AGP25thPercentile UPPER=AGP75thPercentile/<OPTIONS>;
  REFLINE 54 250/AXIS=Y <OPTIONS>;
  REFLINE 70 180/AXIS=Y <OPTIONS>;
  SERIES X=Hour Y=AGP50thPercentile/<OPTIONS>;
  XAXIS LABEL=" " VALUES=(0 TO 24 BY 1) <OPTIONS>;
  YAXIS LABEL=" " VALUES=(0 54 250 350) <OPTIONS>;
  6 Y2AXIS <OPTIONS>;
RUN;

```

Here is an inside look at a few code lines:

- 1 %SGANNO only needs to be called one time in each SAS session. It is repeated in each of the examples here so each example can run independently of the others.
- 2 Getting the X- and Y- positioning, line thickness, and text size correct involves lots of trial and error.
- 3 The ROTATE=90 option is used to turn the text 90 degrees. The FILLCOLOR="WHITE" gives the text box a white background, allowing it to cover the green vertical line drawn previously.
- 4 User-defined formats are useful tools to supplement SG annotations! In this case, the format is being used to apply custom labels to some values of the x-axis and to suppress labels for others.
- 5 Order is important. SAS graphs the statements in the order they are specified.
- 6 The Y2AXIS is only being used to make the tick marks on the right side of the y-axis.

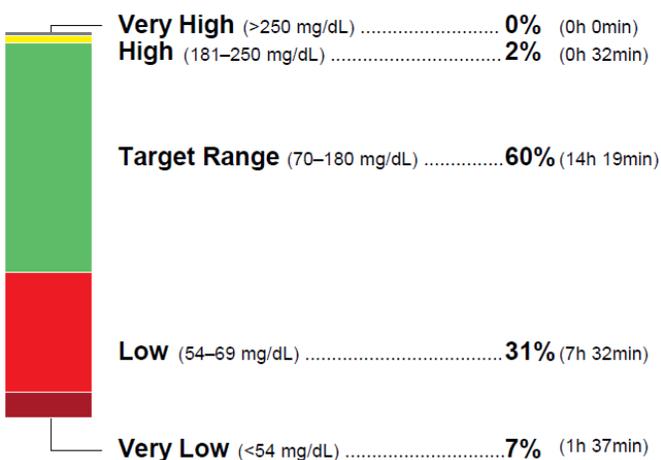
## EXAMPLE #2: ADDING DESCRIPTIVE TEXT NEXT TO A STACKED BAR CHART

The next example is the stacked bar graph taken from the “Time in Range” section of the AGP report. To the right side of the stacked bar are text descriptions giving each bar segment a name, citing its clinical definition, and reporting the percent and amount of time spent in that range. PROC SGPLOT with VBAR statement can generate the stacked bar chart, though we need SG annotations to add the text labels as shown in Figure 7.

Result from PROC SGPLOT before annotations:



Result from PROC SGPLOT after annotations:



**Figure 7: PROC SGPLOT with VBAR Statement Before (Left) and After (Right) SG Annotation**

The SG annotate goal: Add the descriptive text labels. The leading lines from the top and bottom of the graph also need to be added by annotation, but that code will only be presented in the Appendix.

How to approach it: In this case, we use a VBAR statement within PROC SGPLOT to create the stacked bar chart. As there is not a way to add that much text to the plot with PROC SGPLOT features, these text descriptions are a good candidates for annotation. In this case, we create an annotate dataset and apply it to a blank graph that sits next to the stacked bar chart. In this method, we can use the default layout draw space.

In the code below, the two graphs are placed within neighboring ODS REGIONS of a PDF so they appear to be part of the same figure. The leading lines are created in a separate data set and applied to the bar chart (see Appendix). To create the text descriptions within an SG annotate data set, we use four %SGTEXT statements to generate each label. The full code for the “Very High” bar segment is included here and the rest of the code in the Appendix:

```

/* Stacked bar chart */
/* Compile SG annotate macros */
%SGANNO;
/* Use ODS GRAPHICS statement to set the size of the stacked bar chart and
to suppress the graph border */
ODS GRAPHICS/ RESET=ALL WIDTH=0.6in HEIGHT=2.5in BORDER=OFF;
/* Make the stacked bar chart */
PROC SGPLOT DATA=StackedBarData NOWALL NOBORDER NOAUTOLEGEND
SGANNO=BarLeadLinesAnno;
/* Here we assign the colors to each of the bar segments */
STYLEATTRS DATACOLORS=( "SCARLET" "RED" "GREEN" "YELLOW" "DARKGREY" );
VBAR Visit / RESPONSE=TIRPercent
GROUP=RangeData GROUPDISPLAY=STACK GROUPORDER=DATA
BARWIDTH=1 BASELINEATTRS=( COLOR="WHITE" )

```

1

```

        OUTLINEATTRS=(COLOR="WHITE" THICKNESS=1.25pt);
XAXIS DISPLAY=(NOTICKS NOLINE NOVALUES) LABEL=" " OFFSETMIN=0;
YAXIS DISPLAY=NONE;
RUN;

/* Descriptive labels for the stacked bar chart */
ODS GRAPHICS/ RESET=ALL WIDTH=3in HEIGHT=3in BORDER=OFF;
DATA BarTextAnno;
    /* Text label for the "Very High" segment */
    %SGTEXT(LABEL="Very High", X1=0, Y1=97, WIDTH=100, WIDTHUNIT="PERCENT",
    ② ANCHOR="BOTTOMLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=10, TEXTWEIGHT="BOLD");
    %SGTEXT(LABEL="(>250 mg/dL) .....", X1=23, Y1=97,
    ③ ANCHOR="BOTTOMLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=8, TEXTWEIGHT="NORMAL");
    %SGTEXT(LABEL="0%", X1=71, Y1=97,
        ANCHOR="BOTTOMLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=10, TEXTWEIGHT="BOLD");
    %SGTEXT(LABEL="(0h 0min)", X1=81, Y1=97,
        ANCHOR="BOTTOMLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=8, TEXTWEIGHT="NORMAL");

<NOTE: The descriptions for each of the other segments are
generated using the same set of %SGTEXT macros, replacing labels
and Y-axis positions as needed.>

RUN;
/* Now create a SERIES plot where all graph elements are suppressed,
/* not visible, or forced to be white to blend into the background. */
PROC SGPLOT DATA=AGPData NOAUTOLEGEND NOWALL NOBORDER SGANNO=BarTextAnno;
    WHERE Hour <= 10;
    SERIES X=Hour Y=AGP50thPercentile/ LINEATTRS=(COLOR="WHITE");
    XAXIS DISPLAY=NONE;
    YAXIS DISPLAY=NONE;
RUN;

```

Here are a few items to note in these code lines:

- ① When creating a bar chart with PROC SGPLOT, the baseline axis is not suppressed with the DISPLAY=NONE option. To suppress the x-axis, we color it white so it blends in with the background and seems invisible.
- ② The WIDTH=100 AND WIDTHUNIT=PERCENT set the text box region to span the entire width of the layout draw space. Once this parameter is set, it will be applied to all of the other observations in the annotate data set, so it only needs to be set at the start and any time you want it changed. This can be verified by checking the annotate data set as shown in Display 2.
- ③ The code could have been created with a combination of %SGTEXT and %SGTEXTCONT statements – the reason for only using %SGTEXT was personal preference.

	ANCHOR	FUNCTION	LABEL	TEXTCOLOR	TEXTFONT	TEXTSIZE	TEXTWEIGHT	WIDTH	WIDTHUNI	X1	Y1
1	BOTTOMLEFT	TEXT	Very High	BLACK	Arial	10	BOLD	100	PERCENT	0	97
2	BOTTOMLEFT	TEXT	(>250 mg/dL) .....	BLACK	Arial	8	NORMAL	100	PERCENT	23	97
3	BOTTOMLEFT	TEXT	0%	BLACK	Arial	10	BOLD	100	PERCENT	71	97
4	BOTTOMLEFT	TEXT	(0h 0min)	BLACK	Arial	8	NORMAL	100	PERCENT	81	97

**Display 2: The First Four Observations of the BarTextAnno Data Set**

### EXAMPLE #3: CREATING A CUSTOM TEXT TABLE

In the last AGP example, we recreate the “Glucose Statistics and Targets” section of the report. Typically, the ODSTEXT procedure or PROC REPORT might come to mind when making tables; though this table is text heavy and the format of the table varies row-by-row. This non-standard formatting makes it a good candidate for an annotate approach even though there is no corresponding graph.

The SG annotate goal: Create a text table with varying row formats and different columns widths.

<b>01 JAN 2020-15 JAN 2020</b>	<b>14 days</b>
<b>% Time CGM is Active</b>	<b>98.9%</b>
<hr/>	
<b>Glucose Ranges</b>	<b>Targets [% of Readings (Time/Day)]</b>
Target Range 70–180 mg/dL.....	Greater than 70% (16h 48min)
Below 70 mg/dL.....	Less than 4% (58min)
Below 54 mg/dL.....	Less than 1% (14min)
Above 180 mg/dL.....	Less than 25% (6h)
Above 250 mg/dL.....	Less than 5% (1h 12min)
Each 5% increase in time in range (70–180 mg/dL) is clinically beneficial.	
<hr/>	
<b>Average Glucose</b>	<b>86 mg/dL</b>
<b>Glucose Management Indicator (GMI)</b>	<b>5.4%</b>
<b>Glucose Variability</b>	<b>37.8%</b>
Defined as percent coefficient of variation (%CV); target $\leq 36\%$	

**Figure 8: Custom Table Made Entirely of SG Annotate Statements**

How to approach it: Even though there is not a graph in this section, we utilize the SG annotations to get the custom look of the table. Specifically, we will use the %SGTEXT and %SGLINE macros. To get an annotate data set to print without a graph, we will use the same trick as in Example 2 and create an empty graph in which to apply the annotate data set. In this case, we need to change the draw space to the graph draw space.

**TIP:** Consider using SG annotations for non-traditional uses such as creating custom tables or designing a custom image or figure (like a flow chart) & overlaying them onto an empty graph.

In brief, each line of text within the table will be constructed of multiple %SGTEXT statements similar to Example 2:

```

/* Create the table from top to bottom */
DATA StatsAnno;
  /* Days of CGM Use */
  %SGTEXT(LABEL="01 JAN 2020-15 JAN 2020", X1=0, Y1=99,
    DRAWSPACE="GRAPHPERCENT", WIDTH=100, WIDTHUNIT="PERCENT",
    ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
    TEXTSIZE=10, TEXTWEIGHT="BOLD");
  %SGTEXT(LABEL="14 days", X1=75, Y1=99,
    ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
    TEXTSIZE=10, TEXTWEIGHT="BOLD");
  <MORE %SGTEXT statements for the Percent of Time CGM is Active row>
  /* Section break */
  %SGLINE(X1=0, Y1=84, X2=100, Y2=84, LINECOLOR="BLACK",
    LINETHICKNESS=1);

```

```

    <MORE %SGTEXT statements creating the rest of the rows>
RUN;
/* Now create a SERIES plot where all graph elements are suppressed,
/* not visible, or forced to be white to blend into the background. This
/* is the same blank graph as in Example 2. */
PROC SGPLOT DATA=AGPData NOAUTOLEGEND NOWALL NOBORDER SGANNO=BarTextAnno;
  WHERE Hour <= 10;
  SERIES X=Hour Y=AGP50thPercentile/ LINEATTRS=(COLOR="WHITE");
  XAXIS DISPLAY=NONE;
  YAXIS DISPLAY=NONE;
RUN;

```

## A STEP BEYOND: COMBINING OUTPUT FROM GRAPHS AND TABLES

Despite the level of customization that can be achieved with the features of SG procedures and the SG annotate facility, you may want to create data visualizations that require integrating tabular data with graphs in addition to using these features. This last section provides two examples of such a situation.

Both examples come from an automated SAS report used to monitor the number of registrations to a state tobacco cessation program and to monitor which combinations of services people are registering for. Two of the services, counseling calls and nicotine replacement therapy (NRT), have a body of research that support their effectiveness in helping people quit using tobacco products. Other services, such as an online web program, cessation emails, supportive text messages, and brochures, give people more information and support, but have not yet shown to increase quit rates. While program participants can sign up for any and all of the services, it is important that most people receive the evidence-based services.

### EXAMPLE #4: EMBEDDING SPARKLINES IN A TABLE

One of the items being monitored is the number of registrations for the program. In tracking the number of registrations by time, program, and areas of the state, it was difficult to see overall trends and any areas that had unique trends. To aid in seeing the trends, we decided to embed sparklines, or small line graphs, into the table.

The data visualization goal: Embed tiny line graphs into the cells of the table.

How to approach it: In this visualization, we create two elements – a table and a “stacked” line chart – and then lay the line chart on top of the table using overlapping ODS REGIONS within a PDF. The key to getting this to work is getting the dimensions of the two elements the same and then being able to layer the two elements on top of one another.

The table was created with a PROC REPORT. The underlying data set looks very similar to the finished table. PROC REPORT in this situation is being used as a fancy printing tool more than a reporting/calculation tool. Using a COMPUTE block in PROC REPORT, we add a blank column that is large enough to hold the plots (see Figure 9 left side).

For the “stacked” line chart, we first convert the number of registrations to percents of total within each area-program combination. That way, all the data points are bounded by [0, 1] and all the line graphs can take up the same amount of vertical space. Then, to get all the lines stacked, we create a new y-axis variable by adding consecutive integers to the percentages for each area-program combination. For instance, we add 7 to the percentages from the Western area program AAAA, so all of its values are bounded by [7, 8]. Similarly, we add 6 to the percentages from the Western area program BBB. This approach allows us to “stack” all the line charts in one SGPLOT as shown in the right side of Figure 9.

Output from PROC REPORT with a blank column for the line graphs

DMA	Program	FY17		FY18		
		Q4	Q1	Q2	Q3	Q4
Western area	AAAA	1,167	817	868	1,050	998
	BBBB	1,262	896	983	1,084	950
	CCCC	603	467	505	585	425
	DDDD	1,845	1,597	1,728	2,097	1,925
Eastern area	AAAA	157	119	97	111	129
	BBBB	237	158	179	162	154
	CCCC	91	89	82	90	61
	DDDD	135	157	121	180	170

“Stacked” line graph from SGPLOT with same dimensions as blank cells

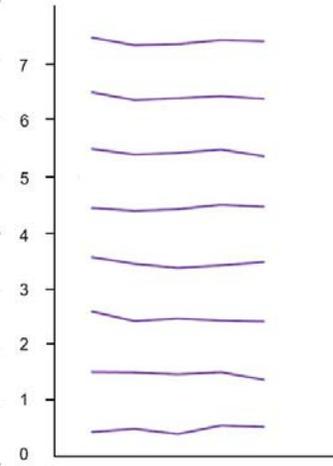


Figure 9: Output from PROC REPORT and PROC SGPLOT Separately

Then, to finish the visualization, we use overlapping ODS REGIONS to layer the two elements as shown in Figure 10.

DMA	Program	FY17		FY18			
		Q4	Q1	Q2	Q3	Q4	
Western area	AAAA	1,167	817	868	1,050	998	
	BBBB	1,262	896	983	1,084	950	
	CCCC	603	467	505	585	425	
	DDDD	1,845	1,597	1,728	2,097	1,925	
Eastern area	AAAA	157	119	97	111	129	
	BBBB	237	158	179	162	154	
	CCCC	91	89	82	90	61	
	DDDD	135	157	121	180	170	

Figure 10: The Final Table with Embedded Sparklines Using PROC REPORT with PROC SGPLOT

The code to generate the REPORT and SGPLOT are more heavily focused on the formatting:

```

/* NOTE: Overlapping ODS REGIONS are used to print the SGPLOT on top of the
PROC REPORT. */
/* Make the report table */
ODS REGION WIDTH=100pct HEIGHT=8.25in x=0in y=1.25in;
PROC REPORT DATA = RegistrationCounts NOWINDOWS NOCENTER SPLIT='*'
CONTENTS="" SPANROWS
STYLE(report)=[RULES=NONE FRAME=VOID CELLSPACING=0 CELLPADDING=3.5pt]
STYLE(header)=[BACKGROUND="PURPLE" FOREGROUND="WHITE" FONT_SIZE=11pt];

```

```

COLUMN DMACat ProgramType ("FY17" FY17Q4) ("FY18" FY18Q1 FY18Q2 FY18Q3
    FY18Q4) BlankCol;
DEFINE DMACat /ORDER ORDER=INTERNAL "DMA" LEFT FORMAT=DmaFmt.
    STYLE(COLUMN)=[FONT_SIZE=10pt] STYLE(HEADER)=[FONT_SIZE=11pt];
DEFINE ProgramType /ORDER ORDER=INTERNAL "Program" LEFT FORMAT=ServFmt.
    STYLE(COLUMN)=[FONT_SIZE=10pt VJUST=C]
    STYLE(HEADER)=[FONT_SIZE=11pt];
DEFINE FY17Q4 /DISPLAY "Q4" RIGHT FORMAT=COMMA8.0
    STYLE(COLUMN)=[FONT_SIZE=10pt CELLHEIGHT=0.45in VJUST=C]
    STYLE(HEADER)=[FONT_SIZE=11pt];
<REPEAT THIS DEFINE STATEMENT FOR EACH OF THE QUARTERS OF DATA. FULL
CODE CAN BE FOUND IN THE APPENDIX. >
/* Here we are inserting a blank column with white background and
enough space for our sparklines */
DEFINE BlankCol /COMPUTED " " STYLE=[CELLWIDTH=2in COLOR=WHITE
    CELLHEIGHT=0.45in];

/* Add borders and blank column */
<ADDITIONAL COMPUTE STATEMENTS TO ADD BORDERS. SEE CODE IN APPENDIX.>
COMPUTE BlankCol;
    BlankCol = .;
ENDCOMP;
RUN;

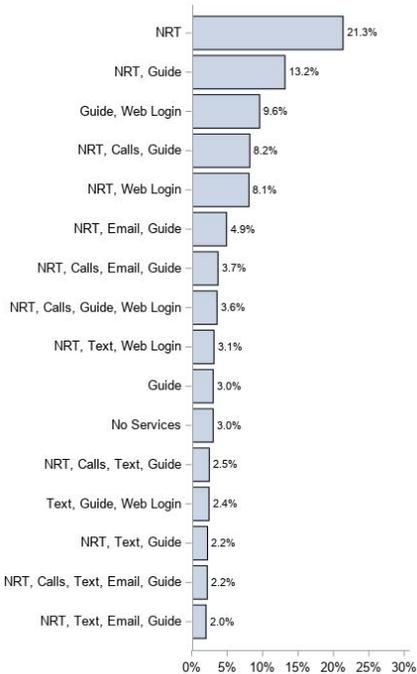
/* SPARKLINES CODE */
/* Create an ODS REGION that overlaps the REPORT region */
ODS REGION WIDTH=2.2in HEIGHT=7.25in X=5.8in Y=1.8in;
/* Set the height and width of sparkline area */
ODS GRAPHICS / RESET=ALL HEIGHT=7.25in WIDTH=2.25in NOBORDER;
/* Create the stacked line chart, suppressing all borders, axes, & labels*/
PROC SGPLOT DATA=SparklinesPercents NOAUTOLEGEND NOBORDER NOOPAQUE NOWALL
    PAD=0pt;
    SERIES X=Calc_Perc_X Y=Calc_Y /GROUP=DMAProgGrp
        LINEATTRS=(PATTERN=SOLID COLOR="PURPLE" THICKNESS=2pt);
    XAXIS DISPLAY=NONE OFFSETMAX=0.25;
    YAXIS VALUES=(0 to 8) DISPLAY=NONE;
RUN;

```

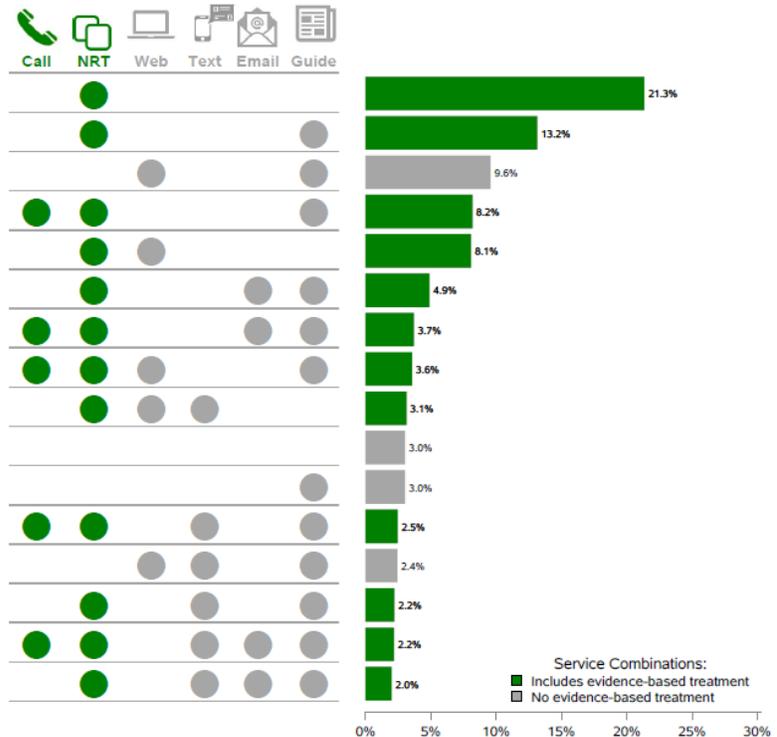
## EXAMPLE #5: ENHANCING AN ORDERED BAR CHART – USING ALL THE TOOLS

In this last example, we demonstrate how PROC REPORT and SG annotations can be used to add visual labels to a horizontal bar chart created from PROC SGPLOT. This example demonstrates the service combination monitoring mentioned above. Using the HBAR statement of PROC SGPLOT produces a basic horizontal bar graph showing combinations of services participants are choosing in decreasing order (Figure 11 left side). In this chart it is hard to get a sense of which services are the most common, and there's nothing to distinguish services that are evidence-based with those that are not.

Basic bar chart from PROC SGPLOT:



Final data visualization from PROC REPORT, PROC SGPLOT, and SG annotate:



**Figure 11: Bar Chart of Most Common Service Combinations for All Registrants Using SGPLOT Before (Left) and After (Right) a Data Visualization Make-Over**

The data visualization goal: Add enhancements to the bar graph to make the service combinations more obvious, including a way to easily distinguish between evidence-based (calls, NRT) and non-evidence based (all other) interventions.

How to approach it: Using a combination of the formatting features within PROC SGPLOT, SG annotations, and PROC REPORT, we can make drastic enhancements to this visual. Specifically, we will first turn the listing of services into a PROC REPORT table. Putting each service in its own column/in a tabular form makes it easier to see which service combinations are most popular. Then we will add icons across the top through an SG annotate data set to give readers a visual cue of the service being described. Additionally, we will order the columns from the most intense service to the least intense and add color coding based on evidence-based services along with a legend.

To prepare the data for this visualization, we create a data set such that each person has an indicator variable (0=did not sign up for this service/1=signed up for this service) for each service. Next, we use the FREQ procedure with LIST option on the TABLES statement to create a count of all existing service combinations. Save the PROC FREQ output to its own dataset and sort it in descending order. To focus on the most common combinations, we limit to any combinations with >2% of registrations.

Once that dataset is created, use PROC REPORT with a user-defined format, SG annotations with a blank graph, and PROC SGPLOT together like this:

```

/* CREATE TABLE OF SERVICES USING PROC REPORT */
/* Create a format that will apply an empty label to indicator (XXXXFlag)
/* values of 0 and a dot in green or grey for evidence-based service or
/* not, respectively, for indicator values of 1. */
PROC FORMAT;
1 VALUE Evfmt 0=" " 1="^S={COLOR=green FONT_FACE='Webdings'}n";

```

```

VALUE NoEvfmt 0=" " 1="^S={COLOR=cxa6a6a6 FONT_FACE='Webdings'}n";
RUN;
/* Generate table of service combinations with PROC REPORT, applying the
custom format */
PROC REPORT DATA = ServCombo_v3 NOWINDOWS NOCENTER SPLIT='*' CONTENTS=""
STYLE(REPORT)=[RULES=none FRAME=void CELLSPACING=0 CELLSPACING=3pt]
STYLE(COLUMN)=[BORDERBOTTOMCOLOR=cxa6a6a6 BORDERBOTTOMWIDTH=0.5pt]
STYLE(HEADER)=[BORDERBOTTOMCOLOR=cxa6a6a6 BORDERBOTTOMWIDTH=0.5pt];
COLUMN CallFlag NRTFlag WebFlag TextFlag EmailFlag QGFlag;
DEFINE CallFlag /DISPLAY "Call" CENTER FORMAT=EvFmt.
STYLE(COLUMN)=[FONTSIZE=20.75pt CELLWIDTH=0.55in]
STYLE(HEADER)=[COLOR=GREEN FONTSIZE=12pt];
DEFINE NRTFlag /ANALYSIS "NRT" CENTER FORMAT=EvFmt.
STYLE(COLUMN)=[FONTSIZE=20.75pt CELLWIDTH=0.55in]
STYLE(HEADER)=[COLOR=GREEN FONTSIZE=12pt];
DEFINE WebFlag /ANALYSIS "Web" CENTER FORMAT=NoEvFmt.
STYLE(COLUMN)=[FONTSIZE=20.75pt CELLWIDTH=0.55in]
STYLE(HEADER)=[COLOR=cxa6a6a6 FONTSIZE=12pt];
DEFINE TextFlag /ANALYSIS "Text" CENTER FORMAT=NoEvFmt.
STYLE(COLUMN)=[FONTSIZE=20.75pt CELLWIDTH=0.55in]
STYLE(HEADER)=[COLOR=cxa6a6a6 FONTSIZE=12pt];
DEFINE EmailFlag /ANALYSIS "Email" CENTER FORMAT=NoEvFmt.
STYLE(COLUMN)=[FONTSIZE=20.75pt CELLWIDTH=0.55in]
STYLE(HEADER)=[COLOR=cxa6a6a6 FONTSIZE=12pt];
DEFINE QGFlag /ANALYSIS "Guide" CENTER FORMAT=NoEvFmt.
STYLE(COLUMN)=[FONTSIZE=20.75pt CELLWIDTH=0.6in]
STYLE(HEADER)=[COLOR=cxa6a6a6 FONTSIZE=12pt];
RUN;

/* ADD THE ICONS USING SGANNOTATE OVER A BLANK GRAPH */
/* Set the region above the REPORT output */
ODS REGION WIDTH=3.5in HEIGHT=0.54in X=0in Y=1.42in;
/* Compile SG annotate macros */
%SGANNO;
/* Create the annotate data set with the images */
DATA IconAnno;
%SGIMAGE (IMAGE="&iconpath\PhoneIconGREEN.jpg", X1=0, Y1=0,
DRAWSPACE="GRAPHPERCENT", WIDTH=14, WIDTHUNIT="PERCENT",
HEIGHT=85, HEIGHTUNIT="PERCENT", ANCHOR="BOTTOMLEFT",
BORDER="FALSE");
%SGIMAGE (IMAGE="&iconpath\NRTIconGREEN.jpg", X1=18, Y1=0,
WIDTH=11, WIDTHUNIT="PERCENT", HEIGHT=66,
HEIGHTUNIT="PERCENT", ANCHOR="BOTTOMLEFT", BORDER="FALSE");
%SGIMAGE (IMAGE="&iconpath\WebIconGREY.jpg", X1=34, Y1=0,
WIDTH=15, WIDTHUNIT="PERCENT", HEIGHT=94,
HEIGHTUNIT="PERCENT", ANCHOR="BOTTOMLEFT", BORDER="FALSE");
%SGIMAGE (IMAGE="&iconpath\TextIconGREY.jpg", X1=52, Y1=0,
WIDTH=17, WIDTHUNIT="PERCENT", HEIGHT=100,
HEIGHTUNIT="PERCENT", ANCHOR="BOTTOMLEFT", BORDER="FALSE");
%SGIMAGE (IMAGE="&iconpath\EmailIconGREY.jpg", X1=66.5, Y1=0,
WIDTH=14, WIDTHUNIT="PERCENT", HEIGHT=85,
HEIGHTUNIT="PERCENT", ANCHOR="BOTTOMLEFT", BORDER="FALSE");
%SGIMAGE (IMAGE="&iconpath\GuideIconGREY.jpg", X1=83, Y1=0,
WIDTH=17, WIDTHUNIT="PERCENT", HEIGHT=100,
HEIGHTUNIT="PERCENT", ANCHOR="BOTTOMLEFT", BORDER="FALSE");
RUN;
/* Set the size of the graph */
ODS GRAPHICS / RESET=ALL WIDTH=3.4in HEIGHT=0.53in NOBORDER;

```

```

/* Create the blank graph and apply the annotate data set */
PROC SGPLOT DATA = ServCombo_v1 SGANNO=IconAnno
  NOWALL NOBORDER NOAUTOLEGEND NOOPAQUE;
  WHERE Percent >= 10;
  HBAR NRTFlag / RESPONSE=Percent NOFILL
    OUTLINE OUTLINEATTRS=(COLOR="WHITE")
    BASELINEATTRS=(COLOR="WHITE");
  XAXIS DISPLAY=NONE;
  YAXIS DISPLAY=NONE;
RUN;

/* CREATE BAR CHART USING SGPLOT AND CUSTOM FORMATS */
/* Set the region to be adjacent to the REPORT output */
ODS REGION WIDTH=4.6in HEIGHT=7.45in x=3.43in y=1.95in;
/* Set the size of the bar chart */
ODS GRAPHICS / RESET=ALL BORDER=OFF HEIGHT=7.45in WIDTH=4.7in;
/* Create the horizontal bar chart */
PROC SGPLOT DATA = ServCombo_v3 NOBORDER NOWALL;
  FORMAT NoEvBasedPerc EvBasedPerc PERCENT7.1;
  FORMAT TextString $35.;
  HBAR TextString /RESPONSE=NoEvBasedPerc CATEGORYORDER=RESPDESC
    NOOUTLINE DATALABEL=NoEvBasedPerc
    FILLATTRS=(COLOR=cxa6a6a6) BASELINEATTRS=(THICKNESS=0)
    LEGENDLABEL="No evidence-based treatment" NAME="B";
  HBAR TextString /RESPONSE=EvBasedPerc NOOUTLINE DATALABEL=EvBasedPerc
    FILLATTRS=(COLOR="GREEN") BASELINEATTRS=(THICKNESS=0)
    LEGENDLABEL="Includes evidence-based treatment" NAME="A";
  YAXIS DISPLAY=NONE;
  XAXIS DISPLAY=(NOLABEL) VALUES=(0 to 0.3 BY 0.05);
  KEYLEGEND "A" "B"/TITLE="Service Combinations:" NOBORDER ACROSS=1
  POSITION=BOTTOMRIGHT LOCATION=INSIDE;
RUN;

```

There are two items to call attention to in this example:

- ❶ To achieve the filled in circles in the REPORT output, we use a user-defined format that assigns a filled circle to the value 1 (Webding character n) and an empty cell to the value 0. Depending on whether or not the service combination includes an evidence-based service, we include the green or grey color.
- ❷ In this SG annotate dataset, each icon has slightly different dimensions, leading to the different widths and heights.

After all these pieces have been applied and the combined output sent to PDF for easier viewing, we see that the resulting data visualization makes it easy to see most service combinations include an evidence-based treatment and that almost all of the combinations involve NRT.

## CONCLUSIONS

In conclusion, SAS offers many powerful data visualization tools to the Base SAS user. This means all SAS users can create visually engaging graphs and charts that communicate results effectively. Knowing what features are available within each of the SG procedures is a great start to creating these high-impact visualizations. When those features are not enough to reach your data visualization goal, consider enhancing the ODS Graphics with SG annotations, by applying custom formatting and attribute maps, and combining SG output with REPORT output.

## REFERENCES

SAS Institute Inc. 2016. *SAS® 9.4 ODS Graphics: Procedures Guide, Sixth Edition*. Cary, NC: SAS Institute Inc.

Heath, D. 2016. "Annotating the ODS Graphics Way!" Proceedings of the Midwest SAS User Group Conference, Cincinnati, OH: SAS Institute, Inc. Available at:

<http://www.mwsug.org/proceedings/2016/DV/MWSUG-2016-DV10-SAS.pdf>

<http://www.agpreport.org/agp/>

## RECOMMENDED READING

- SAS Graphically Speaking blog: <https://blogs.sas.com/content/graphicallyspeaking/>

## CONTACT INFORMATION

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

Sara Richter  
Professional Data Analysts  
SRichter@pdastats.com

## APPENDIX

### CODE FOR FIGURE 1

```
DATA MyAnnoData;
  FUNCTION="Text";
  LABEL="This is my example text from an annotate function.";
  X1=10;
  Y1=95;
  ANCHOR="TOPLEFT";
  WIDTH=100;
OUTPUT;
%SGANNO;
%SGTEXT(
  LABEL="This is my example text from the SGTEXT macro.",
  X1=10,
  Y1=85,
  ANCHOR="TOPLEFT",
  WIDTH=100);
RUN;
PROC SGPLOT DATA=SASHELP.Cars SGANNO=MyAnnoData;
  SCATTER X=MPG_City Y=MPG_Highway;
RUN;
```

### CODE FOR FIGURE 2

```
/* Compile SG annotate macros */
%SGANNO;
/* Define categories for the bar chart */
PROC FORMAT;
  VALUE MPGFmt
    10-14='10-14'
    15-19='15-19'
    20-24='20-24'
    25-29='25-29'
    30-34='20-24'
    35-high='35+';
RUN;
/* Create annotate data set for data draw space */
DATA DataAnno;
  /* Create a grey rectangle that fills the entire draw space */
  %SGRECTANGLE(X1=0, Y1=0, HEIGHT=100, WIDTH=100,
    ANCHOR="BOTTOMLEFT", DRAWSPACE="DATAPERCENT",
    DISPLAY="FILL", FILLCOLOR="GREY", FILLTRANSPARENCY=0.75);
  /* Add the text label */
  %SGTEXT(LABEL="Data draw space", X1=50, Y1=85,
    ANCHOR="CENTER", DRAWSPACE="DATAPERCENT", FILLTRANSPARENCY=1,
    TEXTSIZE=20, TEXTWEIGHT="BOLD");
  /* Draw a horizontal, double-sided arrow pointing to the right and left
  edges of the draw space */
  %SGARROW(X1=0, X2=100, Y1=80, Y2=80,
    DRAWSPACE="DATAPERCENT",
    DIRECTION="BOTH", LINECOLOR="BLACK", SHAPE="FILLED");
  /* Draw a vertical, double-sided arrow pointing to the top and bottom
  edges of the draw space */
  %SGARROW(X1=80, X2=80, Y1=0, Y2=100,
    DRAWSPACE="DATAPERCENT",
    DIRECTION="BOTH", LINECOLOR="BLACK", SHAPE="FILLED");
```

```

/* Add a diagonal arrow pointing to the origin of the draw space and
label it */
%SGARROW(X1=0, X2=20, Y1=0, Y2=25,
         DRAWSPACE="DATAPERCENT",
         DIRECTION="IN", LINECOLOR="BLACK", SHAPE="FILLED");
%SGTEXT(LABEL="Data space origin (0,0)", X1=20, Y1=25,
        ANCHOR="BOTTOMLEFT", DRAWSPACE="DATAPERCENT", FILLTRANSPARENCY=1,
        TEXTSIZE=15, TEXTWEIGHT="BOLD");
RUN;
/* Create vertical bar chart, adding the annotation data set */
PROC SGPLOT DATA=SASHELP.Cars SGANNO=DataAnno;
  FORMAT MPG_City MPGFmt.;
  VBAR MPG_City;
RUN;
/* Code for the wall, layout, and graph draw spaces follows the data draw
space, changing the DRAWSPACE= option to: WALLPERCENT, LAYOUTPERCENT, and
GRAPHPERCENT, as appropriate. */

```

### CODE FOR FIGURE 3

```

/* Compile SG annotate macros */
%SGANNO;
/* Define categories for the bar chart */
PROC FORMAT;
  VALUE MPGFmt
    10-14='10-14'
    15-19='15-19'
    20-24='20-24'
    25-29='25-29'
    30-34='20-24'
    35-high='35+';
RUN;
DATA PadAnno;
  /* Create custom title */
  %SGTEXT(LABEL="SGANNO with PAD=30%", X1=50, Y1=95, WIDTH=100,
         ANCHOR="CENTER", DRAWSPACE="GRAPHPERCENT", FILLTRANSPARENCY=1,
         TEXTSIZE=20);
  /* Create text annotations around each side of the graph */
  %SGTEXT(LABEL="So much room in", X1=10, Y1=50, WIDTH=100,
         ANCHOR="CENTER", DRAWSPACE="GRAPHPERCENT", FILLTRANSPARENCY=1,
         TEXTSIZE=14, ROTATE=270);
  %SGTEXT(LABEL="the graph draw space", X1=50, Y1=10, WIDTH=100,
         ANCHOR="CENTER", DRAWSPACE="GRAPHPERCENT", FILLTRANSPARENCY=1,
         TEXTSIZE=14, ROTATE=0);
  %SGTEXT(LABEL="for annotations", X1=90, Y1=50, WIDTH=100,
         ANCHOR="CENTER", DRAWSPACE="GRAPHPERCENT", FILLTRANSPARENCY=1,
         TEXTSIZE=14, ROTATE=90);
RUN;
PROC SGPLOT DATA=SASHELP.Cars SGANNO=PadAnno PAD=30%;
  FORMAT MPG_City MPGFmt.;
  VBAR MPG_City;
  TITLE "This is the title";
RUN;
DATA PadAnno2;
  /* Create custom title */
  %SGTEXT(LABEL="SGANNO with PAD=(RIGHT=50% TOP=10%)", X1=50, Y1=95,
         WIDTH=100, ANCHOR="CENTER", DRAWSPACE="GRAPHPERCENT",

```

```

        FILLTRANSPARENCY=1, TEXTSIZE=20, TEXTCOLOR="BLACK");
    /* Add annotation text in the right graph space */
    %SGTEXT(LABEL="In this example, there is extra space for annotations on
the right side.", X1=80, Y1=50, WIDTH=30,
        ANCHOR="CENTER", DRAWSPACE="GRAPHPERCENT", FILLTRANSPARENCY=1,
        TEXTSIZE=14);
RUN;
PROC SGPLOT DATA=SASHELP.Cars SGANNO=PadAnno2 PAD=(RIGHT=50% TOP=10%);
    FORMAT MPG_City MPGFmt.;
    VBAR MPG_City;
    TITLE "This is the title";
RUN;

```

## CODE FOR FIGURE 6

```

/* Compile SG annotate macros */
%SGANNO;
/* Create the annotate data set */
DATA AGPAnno;
    /* Add green lines that extend into the layout space */
    /* Adds the lower horizontal green line */
    %SGLINE(X1=3, Y1=28, X2=8, Y2=28, LINECOLOR="GREEN", LINETHICKNESS=8);
    /* Adds the upper horizontal green line */
    %SGLINE(X1=3, Y1=56, X2=8, Y2=56, LINECOLOR="GREEN", LINETHICKNESS=8);
    /* Adds the vertical green line */
    %SGLINE(X1=3, Y1=28, X2=3, Y2=56, LINECOLOR="GREEN", LINETHICKNESS=8);
    /* Add the 'Target Label' text over the green line */
    %SGTEXT(LABEL="Target Range", X1=3, Y1=44,
        ANCHOR="CENTER", ROTATE=90, FILLCOLOR="WHITE",
        TEXTCOLOR="BLACK", TEXTFONT="Arial", TEXTSIZE=7, TEXTWEIGHT="BOLD");
    /* Adjust the y-axis labels at 70 and 180 so they sit inside the green
lines */
    %SGTEXT(LABEL=" 70", X1=6.5, Y1=28, ANCHOR="BOTTOMRIGHT",
        TEXTCOLOR="BLACK", TEXTFONT="Arial", TEXTSIZE=8, TEXTWEIGHT="BOLD");
    %SGTEXT(LABEL="180", X1=6.5, Y1=56, ANCHOR="TOPRIGHT",
        TEXTCOLOR="BLACK", TEXTFONT="Arial", TEXTSIZE=8, TEXTWEIGHT="BOLD");
    /* Add the 'mg/dL' label */
    %SGTEXT(LABEL="mg/dL", X1=3, Y1=65,
        ANCHOR="CENTER", ROTATE=90, FILLCOLOR="WHITE",
        TEXTCOLOR="BLACK", TEXTFONT="Arial", TEXTSIZE=8, TEXTWEIGHT="BOLD");
RUN;
/* Create a format for the x-axis */
PROC FORMAT;
    VALUE XAxisFMT
        0,24="12am" 3="3am" 6="6am" 9="9am" 12="12pm" 15="3pm" 18="6pm"
        21="9pm" OTHER=" ";
RUN;
/* Use ODS GRAPHICS statement to set the size of the graph and to suppress
the graph border */
ODS GRAPHICS/ RESET=ALL WIDTH=7.5in HEIGHT=3.75in BORDER=OFF;
/* Generate the AGP plot */
PROC SGPLOT DATA = AGPData SGANNO=AGPAnno NOAUTOLEGEND;
    FORMAT Hour XAxisFMT.;
    BAND X=Hour LOWER=AGP5thPercentile UPPER=AGP95thPercentile/
    FILL FILLATTRS=(COLOR="CXCAD7EA")
    OUTLINE LINEATTRS=(COLOR="CX4289C0" PATTERN=3)
    CURVELABELLOC=OUTSIDE CURVELABELLOWER="5%" CURVELABELUPPER="95%"
    CURVELABELATTRS=(FAMILY="Arial" SIZE=8pt);
    BAND X=Hour LOWER=AGP25thPercentile UPPER=AGP75thPercentile/

```



```

/* Text label for the "Very High" segment */
%SGTEXT(LABEL="Very High", X1=0,Y1=99, WIDTH=100, WIDTHUNIT="PERCENT",
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=10, TEXTWEIGHT="BOLD");
%SGTEXT(LABEL="(>250 mg/dL) .....", X1=23, Y1=98,
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=8, TEXTWEIGHT="NORMAL");
%SGTEXT(LABEL="0%", X1=71, Y1=99,
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=10, TEXTWEIGHT="BOLD");
%SGTEXT(LABEL="(0h 0min)", X1=81, Y1=98,
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=8, TEXTWEIGHT="NORMAL");

/* Text label for the "High" segment */
%SGTEXT(LABEL="High", 1=0, 1=96, WIDTH=100, WIDTHUNIT="PERCENT",
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=10, TEXTWEIGHT="BOLD");
%SGTEXT(LABEL="(181-250 mg/dL) .....",
        X1=12, Y1=95, ANCHOR="TOPLEFT", TEXTCOLOR="BLACK",
        TEXTFONT="Arial", TEXTSIZE=8, TEXTWEIGHT="NORMAL");
%SGTEXT(LABEL="2%", X1=71,Y1=96,
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=10, TEXTWEIGHT="BOLD");
%SGTEXT(LABEL="(0h 32min)", X1=81, Y1=95,
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=8, TEXTWEIGHT="NORMAL");

/* Text label for the "In Target" segment */
%SGTEXT(LABEL="Target Range", X1=0, Y1=68, WIDTH=100,
        WIDTHUNIT="PERCENT", ANCHOR="TOPLEFT", TEXTCOLOR="BLACK",
        TEXTFONT="Arial", TEXTSIZE=10, TEXTWEIGHT="BOLD");
%SGTEXT(LABEL="(70-180 mg/dL) .....", X1=31, Y1=67,
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=8, TEXTWEIGHT="NORMAL");
%SGTEXT(LABEL="60%", X1=71,Y1=68,
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=10, TEXTWEIGHT="BOLD");
%SGTEXT(LABEL="(14h 19min)", X1=81, Y1=67,
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=8, TEXTWEIGHT="NORMAL");

/* Text label for the "Low" segment */
%SGTEXT(LABEL="Low", X1=0,y1=22, WIDTH=100, WIDTHUNIT="PERCENT",
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=10, TEXTWEIGHT="BOLD");
%SGTEXT(LABEL="(54-69 mg/dL) .....",
        X1=12, Y1=21, ANCHOR="TOPLEFT", TEXTCOLOR="BLACK",
        TEXTFONT="Arial", TEXTSIZE=8, TEXTWEIGHT="NORMAL");
%SGTEXT(LABEL="31%", X1=71,Y1=22,
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=10, TEXTWEIGHT="BOLD");
%SGTEXT(LABEL="(7h 32min)", X1=81, Y1=21,
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial", TEXTSIZE=8,
        TEXTWEIGHT="NORMAL");

/* Text label for the "Very Low" segment */
%SGTEXT(LABEL="Very Low", X1=0, Y1=21, WIDTH=100, WIDTHUNIT="PERCENT",

```

```

        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=10, TEXTWEIGHT="BOLD");
%SGTEXT(LABEL="(<54 mg/dL) .....",
        X1=12, Y1=20, ANCHOR="TOPLEFT", TEXTCOLOR="BLACK",
        TEXTFONT="Arial", TEXTSIZE=8, TEXTWEIGHT="NORMAL");
%SGTEXT(LABEL="7%", X1=71, Y1=21,
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial",
        TEXTSIZE=10, TEXTWEIGHT="BOLD");
%SGTEXT(LABEL="(1h 37min)", X1=81, Y1=20,
        ANCHOR="TOPLEFT", TEXTCOLOR="BLACK", TEXTFONT="Arial", TEXTSIZE=8,
        TEXTWEIGHT="NORMAL");

RUN;
/* Now create a SERIES plot where all graph elements are suppressed,
/* not visible, or forced to be white to blend in to the background. */
PROC SGPLOT DATA=AGPData NOAUTOLEGEND NOWALL NOBORDER SGANNO=BarTextAnno;
    WHERE Hour <= 10;
    SERIES X=Hour Y=AGP50thPercentile/ LINEATTRS=(COLOR="WHITE");
    XAXIS DISPLAY=NONE;
    YAXIS DISPLAY=NONE;
RUN;

```

## CODE FOR FIGURE 10

```

/* NOTE: Overlapping ODS REGIONS are used to print the SGPLOT on top of the
PROC REPORT. */
/* Make the report table */
ODS REGION WIDTH=100pt HEIGHT=8.25in x=0in y=1.25in;
PROC REPORT DATA = RegistrationCounts NOWINDOWS NOCENTER SPLIT='*'
    CONTENTS="" SPANROWS
    STYLE(report)=[RULES=NONE FRAME=VOID CELLSPACING=0 CELLPADDING=3.5pt]
    STYLE(header)=[BACKGROUND="PURPLE" FOREGROUND="WHITE" FONT_SIZE=11pt];
    COLUMN DMACat ProgramType ("FY17" FY17Q4) ("FY18" FY18Q1 FY18Q2 FY18Q3
        FY18Q4) BlankCol;
    DEFINE DMACat /ORDER ORDER=INTERNAL "DMA" LEFT FORMAT=DmaFmt.
        STYLE(COLUMN)=[FONT_SIZE=10pt] STYLE(HEADER)=[FONT_SIZE=11pt];
    DEFINE ProgramType /ORDER ORDER=INTERNAL "Program" LEFT FORMAT=ServFmt.
        STYLE(COLUMN)=[FONT_SIZE=10pt VJUST=C]
        STYLE(HEADER)=[FONT_SIZE=11pt];
    DEFINE FY17Q4 /DISPLAY "Q4" RIGHT FORMAT=COMMA8.0
        STYLE(COLUMN)=[FONT_SIZE=10pt CELLHEIGHT=0.45in VJUST=C]
        STYLE(HEADER)=[FONT_SIZE=11pt];
    DEFINE FY18Q1 /DISPLAY "Q1" RIGHT FORMAT=COMMA8.0
        STYLE(COLUMN)=[FONT_SIZE=10pt CELLHEIGHT=0.45in VJUST=C]
        STYLE(HEADER)=[FONT_SIZE=11pt];
    DEFINE FY18Q2 /DISPLAY "Q2" RIGHT FORMAT=COMMA8.0
        STYLE(COLUMN)=[FONT_SIZE=10pt CELLHEIGHT=0.45in VJUST=C]
        STYLE(HEADER)=[FONT_SIZE=11pt];
    DEFINE FY18Q3 /DISPLAY "Q3" RIGHT FORMAT=COMMA8.0
        STYLE(COLUMN)=[FONT_SIZE=10pt CELLHEIGHT=0.45in VJUST=C]
        STYLE(HEADER)=[FONT_SIZE=11pt];
    DEFINE FY18Q4 /DISPLAY "Q4" RIGHT FORMAT=COMMA8.0
        STYLE(COLUMN)=[FONT_SIZE=10pt CELLHEIGHT=0.45in VJUST=C]
        STYLE(HEADER)=[FONT_SIZE=11pt];
    /* Here we are inserting a blank column with white background and enough
    space for our sparklines */
    DEFINE BlankCol /COMPUTED " " STYLE=[CELLWIDTH=2in COLOR=WHITE
        CELLHEIGHT=0.45in];

```

```

/* Add borders and blank column */
COMPUTE BEFORE ProgramType;
    Count+1;
ENDCOMP;
COMPUTE DMACat;
/* Adds the light grey bottom cell border */
CALL DEFINE (_row_, 'style', 'style={BORDERBOTTOMWIDTH=0.5pt
    BORDERBOTTOMCOLOR=cxe2e2e2}');
/* Adds the black border between DMAs */
IF DMACat IN(1,2) THEN CALL DEFINE
    (_COL_, 'style', 'style={BORDERBOTTOMWIDTH=0.5pt
    BORDERBOTTOMCOLOR=black}');
IF Count IN(4,8) THEN CALL DEFINE
    (_ROW_, 'style', 'style={BORDERBOTTOMWIDTH=0.5pt
    BORDERBOTTOMCOLOR=black}');
ENDCOMP;
COMPUTE BlankCol;
    BlankCol = .;
ENDCOMP;
RUN;

/* SPARKLINES CODE */
/* Create an ODS REGION that overlaps the REPORT region */
ODS REGION WIDTH=2.2in HEIGHT=7.25in X=5.8in Y=1.8in;
/* Set the height and width of sparkline area */
ODS GRAPHICS / RESET=ALL HEIGHT=7.25in WIDTH=2.25in NOBORDER;
/* Create the stacked line chart, suppressing all borders, axes, and labels
*/
PROC SGPLOT DATA=SparklinesPercents NOAUTOLEGEND NOBORDER NOOPAQUE NOWALL
    PAD=0pt;
    SERIES X=Calc_Perc_X Y=Calc_Y /GROUP=DMAProgGrp
        LINEATTRS=(PATTERN=SOLID COLOR="PURPLE" THICKNESS=2pt);
    XAXIS DISPLAY=NONE OFFSETMAX=0.25;
    YAXIS VALUES=(0 to 8) DISPLAY=NONE;
RUN;

```

## CODE FOR FIGURE 11

```

/* ----- */
/* DATA PREPARATION */
/* Use the LIST option of a PROC FREQ and send the output to a dataset
/* called ServCombo_v1 */
PROC FREQ DATA = FinalEnroll;
    TABLES NRTFlag*CallFlag*TextFlag*EmailFlag*WebFlag*QGFlag/LIST MISSING
    OUT=ServCombo;
RUN;
/* Sort and flag combinations with > 2% */
PROC SORT DATA = ServCombo_v3;
    BY DESCENDING Percent;
    WHERE Percent >= 2;
RUN;
/* In a separate DATA step (not shown), use the Percent and the service
/* flags to create 2 percent columns - 1 column with percentages for
/* evidence-based service combinations (EvBasedPerc) and 1 column for the
/* percentages for non-evidence-based service combinations (NoEvBasedPerc).
/* Also add a unique text string (TextString) with the service combinations
/* to act as the grouping variable for the bars.
/* ----- */
/* ----- Set options and start the PDF. ----- */

```

```

/* Set the escape character - used for styling throughout the report */
ODS ESCAPECHAR="^";

/* Set document properties */
OPTIONS NONUMBER NODATE ORIENTATION=portrait PAPERSIZE=letter
      TOPMARGIN=0.25in BOTTOMMARGIN=0.15in LEFTMARGIN=0.25in
      RIGHTMARGIN=0.25in;

/* Close other output destinations */
ODS _ALL_ CLOSE;
ODS NORESULTS;

/* Clear title */
TITLE;

ODS PDF FILE="P:\Portfolio\SAS Global Forum 2020\test page.pdf" NOTOC
      NOGFOOT NOGTITLE DPI=600;
ODS LAYOUT START;

/* Set the region for the REPORT output */
ODS REGION WIDTH=3.6in HEIGHT=7.4in X=0in Y=1.92in;
/* Create a format that will apply an empty label to indicator (XXXXFlag)
/* values of 0 and a dot in green or grey for evidence-based service or
/* not, respectively, for indicator values of 1.*/
PROC FORMAT;
  VALUE Evfmt 0=" " 1="^S={COLOR=green FONT_FACE='Webdings'}n";
  VALUE NoEvfmt 0=" " 1="^S={COLOR=cxa6a6a6 FONT_FACE='Webdings'}n";
RUN;
/* Generate table of service combinations with PROC REPORT, applying the
custom format */
PROC REPORT DATA = ServCombo_v3 NOWINDOWS NOCENTER SPLIT='*' CONTENTS=""
  STYLE(REPORT)=[RULES=none FRAME=void CELLSPACING=0 CELLPADDING=3pt]
  STYLE(COLUMN)=[BORDERBOTTOMCOLOR=cxa6a6a6 BORDERBOTTOMWIDTH=0.5pt]
  STYLE(HEADER)=[BORDERBOTTOMCOLOR=cxa6a6a6 BORDERBOTTOMWIDTH=0.5pt];
  COLUMN CallFlag NRTFlag WebFlag TextFlag EmailFlag QGFlag;
  DEFINE CallFlag /DISPLAY "Call" CENTER FORMAT=EvFmt.
    STYLE(COLUMN)=[FONTSIZE=20.75pt CELLWIDTH=0.55in]
    STYLE(HEADER)=[COLOR=GREEN FONTSIZE=12pt];
  DEFINE NRTFlag /ANALYSIS "NRT" CENTER FORMAT=EvFmt.
    STYLE(COLUMN)=[FONTSIZE=20.75pt CELLWIDTH=0.55in]
    STYLE(HEADER)=[COLOR=GREEN FONTSIZE=12pt];
  DEFINE WebFlag /ANALYSIS "Web" CENTER FORMAT=NoEvFmt.
    STYLE(COLUMN)=[FONTSIZE=20.75pt CELLWIDTH=0.55in]
    STYLE(HEADER)=[COLOR=cxa6a6a6 FONTSIZE=12pt];
  DEFINE TextFlag /ANALYSIS "Text" CENTER FORMAT=NoEvFmt.
    STYLE(COLUMN)=[FONTSIZE=20.75pt CELLWIDTH=0.55in]
    STYLE(HEADER)=[COLOR=cxa6a6a6 FONTSIZE=12pt];
  DEFINE EmailFlag /ANALYSIS "Email" CENTER FORMAT=NoEvFmt.
    STYLE(COLUMN)=[FONTSIZE=20.75pt CELLWIDTH=0.55in]
    STYLE(HEADER)=[COLOR=cxa6a6a6 FONTSIZE=12pt];
  DEFINE QGFlag /ANALYSIS "Guide" CENTER FORMAT=NoEvFmt.
    STYLE(COLUMN)=[FONTSIZE=20.75pt CELLWIDTH=0.6in]
    STYLE(HEADER)=[COLOR=cxa6a6a6 FONTSIZE=12pt];
RUN;

/* ADD THE ICONS USING SGANNOTATE OVER A BLANK GRAPH */
/* Set the region above the REPORT output */
ODS REGION WIDTH=3.5in HEIGHT=0.54in X=0in Y=1.42in;

```

```

/* Compile SG annotate macros */
%SGANNO;
/* Create the annotate data set with the images */
DATA IconAnno;
  %SGIMAGE (IMAGE="&iconpath\PhoneIconGREEN.jpg", X1=0, Y1=0,
    DRAWSPACE="GRAPHPERCENT", WIDTH=14, WIDTHUNIT="PERCENT",
    HEIGHT=85, HEIGHTUNIT="PERCENT", ANCHOR="BOTTOMLEFT",
    BORDER="FALSE");
  %SGIMAGE (IMAGE="&iconpath\NRTIconGREEN.jpg", X1=18, Y1=0,
    WIDTH=11, WIDTHUNIT="PERCENT", HEIGHT=66,
    HEIGHTUNIT="PERCENT", ANCHOR="BOTTOMLEFT", BORDER="FALSE");
  %SGIMAGE (IMAGE="&iconpath\WebIconGREY.jpg", X1=34, Y1=0,
    WIDTH=15, WIDTHUNIT="PERCENT", HEIGHT=94,
    HEIGHTUNIT="PERCENT", ANCHOR="BOTTOMLEFT", BORDER="FALSE");
  %SGIMAGE (IMAGE="&iconpath\TextIconGREY.jpg", X1=52, Y1=0,
    WIDTH=17, WIDTHUNIT="PERCENT", HEIGHT=100,
    HEIGHTUNIT="PERCENT", ANCHOR="BOTTOMLEFT", BORDER="FALSE");
  %SGIMAGE (IMAGE="&iconpath\EmailIconGREY.jpg", X1=66.5, Y1=0,
    WIDTH=14, WIDTHUNIT="PERCENT", HEIGHT=85,
    HEIGHTUNIT="PERCENT", ANCHOR="BOTTOMLEFT", BORDER="FALSE");
  %SGIMAGE (IMAGE="&iconpath\GuideIconGREY.jpg", X1=83, Y1=0,
    WIDTH=17, WIDTHUNIT="PERCENT", HEIGHT=100,
    HEIGHTUNIT="PERCENT", ANCHOR="BOTTOMLEFT", BORDER="FALSE");
RUN;
/* Set the size of the graph */
ODS GRAPHICS / RESET=ALL WIDTH=3.4in HEIGHT=0.53in NOBORDER;
/* Create the blank graph and apply the annotate data set */
PROC SGPLOT DATA = ServCombo_v1 SGANNO=IconAnno
  NOWALL NOBORDER NOAUTOLEGEND NOOPAQUE;
  WHERE Percent >= 10;
  HBAR NRTFlag / RESPONSE=Percent NOFILL
    OUTLINE OUTLINEATTRS=(COLOR="WHITE")
    BASELINEATTRS=(COLOR="WHITE");
  XAXIS DISPLAY=NONE;
  YAXIS DISPLAY=NONE;
RUN;
/* CREATE BAR CHART USING SGPLOT AND CUSTOM FORMATS */
/* Set the region to be adjacent to the REPORT output */
ODS REGION WIDTH=4.6in HEIGHT=7.45in x=3.43in y=1.95in;
/* Set the size of the bar chart */
ODS GRAPHICS / RESET=ALL BORDER=OFF HEIGHT=7.45in WIDTH=4.7in;
/* Create the horizontal bar chart */
PROC SGPLOT DATA = ServCombo_v3 NOBORDER NOWALL;
  FORMAT NoEvBasedPerc EvBasedPerc PERCENT7.1;
  FORMAT TextString $35.;
  HBAR TextString /RESPONSE=NoEvBasedPerc CATEGORYORDER=RESPDESC
    NOOUTLINE DATALABEL=NoEvBasedPerc
    FILLATTRS=(COLOR=cxa6a6a6) BASELINEATTRS=(THICKNESS=0)
    LEGENDLABEL="No evidence-based treatment" NAME="B";
  HBAR TextString /RESPONSE=EvBasedPerc NOOUTLINE DATALABEL=EvBasedPerc
    FILLATTRS=(COLOR="GREEN") BASELINEATTRS=(THICKNESS=0)
    LEGENDLABEL="Includes evidence-based treatment" NAME="A";
  YAXIS DISPLAY=NONE;
  XAXIS DISPLAY=(NOLABEL) VALUES=(0 to 0.3 BY 0.05);
  KEYLEGEND "A" "B"/TITLE="Service Combinations:" NOBORDER ACROSS=1
    POSITION=BOTTOMRIGHT LOCATION=INSIDE;
RUN;
/* Close the PDF */ ODS LAYOUT END; ODS PDF END;

```