

Labelling without the Hassle: How to Produce Labeled Stacked Bar Charts Using SGPLOT and GTL Without Annotate

Joe Matise, NORC at the University of Chicago, Chicago, IL

ABSTRACT

PROC SGPLOT and GTL are two powerful weapons in a SAS developer's arsenal in producing high quality graphics. However, there are many times where you try to produce a graph and find that an option isn't available for exactly what you want to do.

Stacked bar charts, and similarly 100% stacked bar charts, are a popular method for displaying data with multiple logical groupings. SAS has made producing these charts easy in SGPLOT with the addition of the `groupstype=stacked` option; adding labels to the charts, however, is a more difficult task, as bar charts only support a single label per bar.

While the annotation facility is one possible method for accomplishing this task, users may find it difficult to use for a variety of reasons. In the case of the clustered stacked bar chart, they may find it nearly impossible. We present two different methods for producing these charts, one using GTL and one using SGPLOT, which do not require the use of the annotation facility, and are extremely flexible. We also direct the reader to additional resources, where similar approaches to other graphical problems often are possible.

This paper assumes the reader has familiarity with the basics of SAS ODS Graphics (SGPLOT in particular). Additionally, some sections assume familiarity with the Graph Template Language which underlies much of the SAS ODS Graphics system. The concepts presented require SAS 9.3 or newer to implement.

INTRODUCTION

PROC SGPLOT and the Graph Template Language (GTL) have become very useful tools for producing publication-quality graphics since their introduction in 9.1.3 (preproduction). As they continue to improve and expand with each successive SAS release, more and more of what SAS developers would like to be able to do with graphics becomes possible.

However, as with any tool, there are always things that simply don't work the way you want. Data labels don't position the way you want, groupings don't work the way you expect them to, or you just can't get that bar chart to do three things at once. Annotations don't position themselves on the screen quite where you want them to without a ton of extra work. For every feature the SAS Institute adds, more questions pop up as to how to make something new work the way a developer needs.

Fortunately, SAS has anticipated this, and has added several flexible plot options to make sure your needs are met. This paper will show you how to use these options to approach a particular problem (the labeled clustered stacked bar chart) by building off of SGPLOT methods with GTL. At the end of this paper, you should not only have a few more tools in your belt, but also have a new way of looking at SAS SGPLOT and GTL, and hopefully be able to use this approach to solve new problems of your own.

Finally, a note on versions. ODS Graphics are relatively new, and as such have significant differences in functionality from version to version. The core of this concept will work in SAS 9.3, but you may see some cosmetic differences from the example plots, which were produced in SAS 9.4 TS1M1.

NEW OPTIONS IN 9.4 M2

SAS version 9.4 Maintenance Release M2 was released in August 2014, and with it came several significant improvements relevant to this paper. While the majority of this paper has been written with a focus on prior versions (9.3 or earlier 9.4), a brief overview of the new options is presented at the end of the paper for those who do have this version. Readers who have the newer version may find the core concepts in the remainder of the paper interesting and useful, even if they have an easier way to accomplish this particular need.

ANNOTATED STACKED BAR CHARTS

The stacked bar chart and its cousin, the 100% stacked bar chart, are one of the more popular ways of displaying grouped data elements. SAS has included an easy way of accomplishing this.

First, we will be using the `sashelp.heart` dataset for our examples here. The dataset comes from the Framingham Heart Study, and consists of data from 5209 subjects indicating various individual indicators (height, weight, cholesterol, etc.). We will primarily concern ourselves with two variables, `sex` which indicates the sex of the respondent, and `status` which indicates whether the respondent is alive or dead.

Then, we will plot it using `vbar`. This will make a 100% stacked bar chart.

```
proc SGPLOT data=heartfreq pctllevel=group;
  vbar sex /group=status groupdisplay=stack;
run;
```

Note that this does make the stacked bar chart we want, but if you add data labels, they do not show up in the middle of the chart; only the top bar can be labeled, and it's labeled on top of the chart. This is not helpful for a stacked bar chart.

To accomplish this, the traditional way would be to use `annotate`. That choice has its limitations, particularly for bar charts which typically are used with discrete data. If you are using this as part of a panel, it is very challenging to get the position of the annotations correct as you cannot use `datavalue`. Additionally, bar values often may be defined by a character variable; the `annotate` facility has some difficulty using `datavalue` positioning with character variables.

Fortunately, there is a better way to solve this problem by overlaying the plot with another plot type: a scatterplot. First we will present an example of how scatterplots can be used to display annotations, and then we will give a brief explanation of how overlaid plots work in ODS Graphics. Finally, we will show several options for putting this together to display our final result.

BACKGROUND

USING SCATTERPLOTS TO DISPLAY TEXT

Scatterplots are typically used to show distributions in two dimensions, but at their core are simply a straightforward way to graph points with an `x` and a `y` coordinate. In addition to displaying dots, `Xs`, pluses, and circles, SAS allows you to use the `markerchar` option to display any text you wish, located at the `x-y` coordinate. We can use this to display annotations on graphs, subject to a few limitations.

While this is superficially similar to regular annotation (in both cases you provide `X` and `Y` and get text at the location), it has significant advantages when working with grouped data, because it can automatically position correctly in some groupings; and when used with a discrete axis, it is easier to work with than the clunky `annotate`, which often fails to correctly locate `datavalue` labels when used with discrete, character axes. Further, it can be incorporated into a graph easily without having to learn the 'language' of annotation.

OVERLAID PLOTS IN ODS GRAPHICS

ODS Graphics includes the capability to overlay plots on the same graph, with some restrictions. Plot types to be overlaid must be compatible with one another; this makes sense, given SAS would not know how to combine a pie chart with a line chart, for example. It is not always obvious which plots cannot be overlaid, in part because the way SAS generates plots internally has an impact on this.

Bar charts are "categorization" plots, intended for summarizing data, and as such are compatible with other categorization plots, such as line, dot, and waterfall. They are not directly compatible with plots that do not summarize data (as it would be difficult for SAS to know what to do with the non-summarized data points); so you cannot overlay a scatter plot over a `VBAR`, for example. Additionally, even among types of plots that are compatible, you need to use the same axis orientation; so a `VBAR` is compatible with a `VLINE` but not an `HLINE`.

For example, this doesn't work, as you can imagine - Scatter doesn't know what to use for Y, as it does not summarize data:

```
proc SGPLOT data=sashelp.heart;
  vbar sex /stat=percent
    group=status groupdisplay=stack;
  scatter x=sex y=colpercent;
run;
```

But this does work - and shows a brief glimpse into the kind of thing we'll be doing later on. Here you use two summary charts together, which share an axis (both are 'v' charts). Note that the display isn't very nice looking yet, but we'll improve on that later by using chart types with more customization options.

```
proc SGPLOT data=sashelp.heart;
  vbar sex /
    stat=percent
    group=status
    groupdisplay=cluster;
  vline sex /
    stat=percent
    group=status
    groupdisplay=cluster
    lineattrs=(thickness=0px)
    datalabel;
run;
```

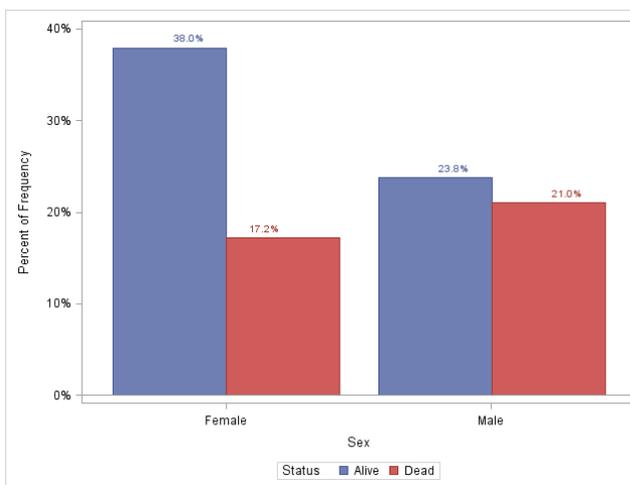


Figure 1. Example Bar Chart with Line-based Annotation

While you cannot overlay scatterplots on VBAR or HBAR, SAS did create a chart type that creates bar charts from summarized data: the BARPARM chart. Using VBARPARM or HBARPARM will create a bar chart from summarized data that is compatible with other basic plot types such as scatter plots. You can also make bar chart-like charts with the HIGHLOW plot type, which can be very useful for getting around some limits of VBARPARM and VBAR and is still compatible with basic plots. Below is an example of VBARPARM overlaid with a simple scatterplot, using a summarized dataset.

```
proc tabulate data=sashelp.heart out=heartfreq;
  class status sex;
  tables sex,status*rowpctn;
run;

proc SGPLOT data=heartfreq;
  vbarparm category=sex response=pctn_01/ group=status;
  scatter x=sex y=pctn_01/group=status
    markerchar=pctn_01 groupdisplay=cluster ;
run;
```

For additional information on overlaying plots, see the SAS ODS Graphics manual page on Plot Type Compatibility, which for 9.4 is available at this link as of the date of publication:
<http://support.sas.com/documentation/cdl/en/grstatproc/66517/HTML/default/viewer.htm#p0yud64khw8fuin1xgr85dgxbb7t.htm>

PRODUCING LABELED STACKED BAR CHARTS

USING HIGH-LOW PLOT PLUS SCATTERPLOT

VBARPARM would be perfect for our needs, except for one detail: it cannot produce stacked bar charts in SGPLOT. The language doesn't include a GROUPDISPLAY option for VBARPARM (or HBARPARM), for some reason, as of SAS version 9.4 TS1M1. This functionality does exist in the Graph Template Language under VBARPARM, which we will cover later, but if you're trying to make a simple plot in SGPLOT, you can't use VBARPARM.

Fortunately, there is a workaround. You can use a High-Low plot to simulate a stacked bar chart directly in SGPLOT, as long as you have pre-summarized and slightly adjusted your data to fit the High-Low plot's needs - and as you'd have to do most of the same work to generate your scatterplot positions, it isn't particularly onerous to do so.

A High-Low plot takes two Y coordinates and one X coordinate (or the reverse), and draws a bar at the X coordinate between the two Y coordinates. So, to produce this chart, we need to take our summarized data and modify it to contain that information. We also need to know where to display our scatterplot labels, similarly needing an X coordinate and a Y coordinate.

```
data heartfreq_highlow;
  set heartfreq;
  by sex status;
  if first.sex then _ybase=0;
  yMin = _ybase;
  yMax = pctn_01+_ybase; *the percentage from PROC TABULATE;
  yScatter = (pctn_01/2)+_ybase;
  output;
  _ybase + pctn_01; *update for next row;
  if pctn_01 < 5 then call missing(pctn_01);
  *less than 5% would not have room to fit - adjust for size of bar;
  format PctN_01 yMin yMax PCTNOMULT.;
  *a percent format for 0-100 %s, details in code in appendix;
run;
```

Then, we request the plots. Here we use group not to create groups (we did that manually), but to allow us to take advantage of the color differentiation by group.

```
proc SGPLOT data=heartfreq_highlow;
  highlow x=sex
  low=yMin high=yMax/
  type=bar barwidth=0.6
  group=status
  groupdisplay=overlay;
  scatter x=sex y=yScatter/
  markerchar=pctn_01
  markercharattrs=
  (size=18pt color=white);
run;
```

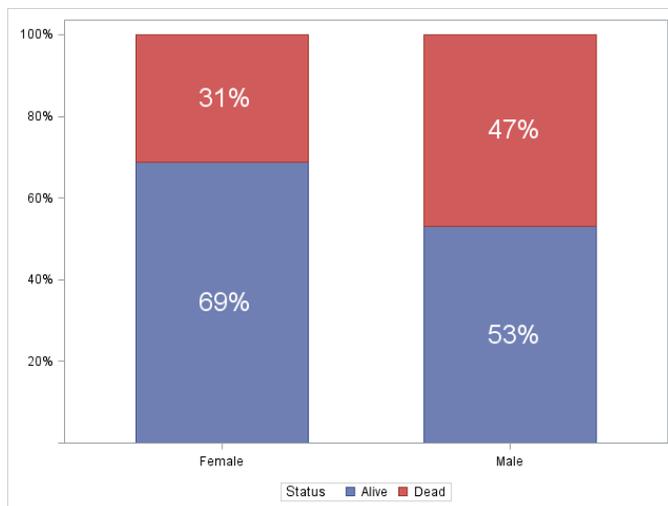


Figure 2. Stacked Bar Charts with Scatterplot Annotation

Further explanation of this method can be found in Sanjay Matange's post, "G100 with SGPLOT", in his blog *Graphically Speaking*.

GRAPH TEMPLATE LANGUAGE

Another tool that can be used to produce labeled stacked bar charts is the Graph Template Language (GTL). This is a somewhat more complex solution, but it has its advantages both in allowing significantly greater customization and in allowing the production of the clustered stacked bar chart later on.

If you have not used the Graph Template Language yet, go read one of the several papers in the Recommended Reading section below that provide an introduction to the topic; we will not going cover it in detail here beyond the basic description. However, it is definitely worth learning, as it is much more powerful than SGPLOT. GTL is the back end of the ODS Graphics engine (SGPLOT and similar); anything that you do in SGPLOT has an equivalent in GTL code. Additionally, some features are available in GTL that are not available in SGPLOT, including VBARPARM stacked bar charts (as of 9.4 M1 – see below for an update to this for 9.4 M2).

USING GTL WITH VBARPARM

In GTL, the code to make the graph is made in a PROC TEMPLATE step, and then played using SGRENDER to combine the graph template with a dataset. Using the same dataset as the high-low plot, we create the template for the BARCHARTPARM with a scatterplot for labels below. GTL code can appear complex due to the need to specify some normally default options (such as the axis options below), but the code that creates the key parts of the graph is very simple, and highlighted for clarity. The code for the BarChartParm and the Scatterplot are nearly identical to the SGPLOT code above; we add the name option to allow the legend to be created (which was automatic with SGPLOT).

```
proc template;
  define statgraph work.heartdata;
    begingraph /;
      layout overlay / xaxisopts=(label=' ')
        yaxisopts=( Label=' ' offsetmin=0);
      BarChartParm X=sex Y=PctN_01 / primary=true Group=Status
        name="Status" groupdisplay=stack barwidth=0.6;
      Scatterplot x=sex y=yScatter /
        group=Status markercharacter=Pctn_01 labelstrip=true
        markercharacterattrs=(size=18 weight=bold color=white);
      DiscreteLegend "Status"/ title="Patient Status";
    endlayout;
  endgraph;
end;
run;
```

Then you call it using SGRENDER.

```
proc sgrender
  template=work.heartdata
  data=work.heartfreq_highlow;
run;
```

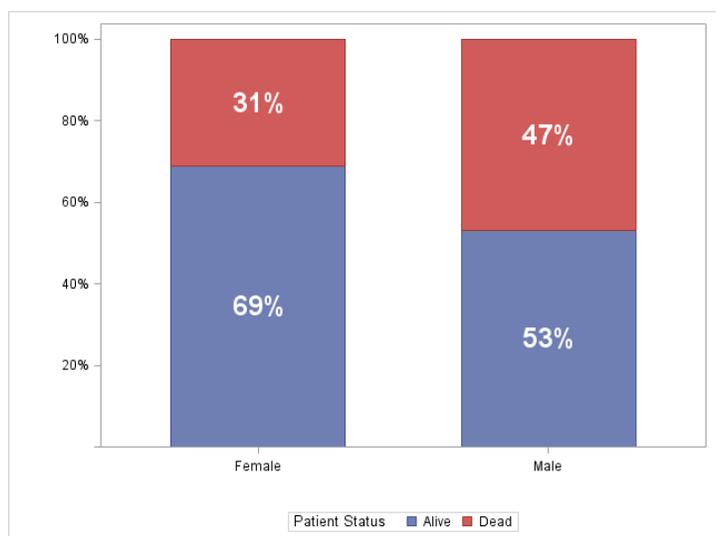


Figure 3. Stacked Bar Chart with Annotations from GTL

CREATING STACKED/CLUSTERED BAR CHARTS

Creating a stacked and clustered bar chart seems at first glance to be a substantially complicated task, as none of the bar charts allow for combining stacking and clustering at the same time; further, the scatterplot will have difficulty properly labelling the bars even if they did do both, as using the GROUPDISPLAY=CLUSTER option would cause them to no longer line up vertically. One might imagine the complexity of writing many high-low charts and carefully assigning each one the correct color; to write this would be a nightmare, not to mention dealing with data changes.

However, with a little bit of flexibility in the layout of the chart, we can take advantage of a GTL feature that allows us to cluster items together based on data elements: the data lattice. If you have used PROC SG PANEL before, you have probably written a set of graphs that used the DATALATTICE element behind the scenes; here, we use it in GTL to give us the ability to cluster our bars while retaining the data labels we developed in the previous section.

Consider this new table, adding in the cholesterol level of the patient:

```
proc tabulate data=sashelp.heart out=heartfreq_chol;
class sex chol_status status ;
table sex, chol_status, status*rowpctn;
run;
```

Then you have a similar post-processed dataset to the previous section, except for adding the new variable (and making it numeric, so it sorts properly - see the appendix for details) and adjusting the name of the PCT variable.

```
data heartfreq_chol_forhl;
set heartfreq_chol;
by sex chol_status status;
if first.chol_status then _ybase=0;
yMin = _ybase;
yMax = pctn_110+ _ybase; *the percentage from PROC TABULATE;
yScatter = (pctn_110/2)+_ybase;
output;
_ybase + pctn_110; *update for next row;
format PctN_110 yMin yMax PCTNOMULT.;
run;
```

Here, we have a GTL template that is fairly similar to the earlier template. You will see the datalattice layout type wrapping the charting code, causing our separate panels to be created (one per value of sex). The prototype layout is nearly identical to the single panel GTL earlier, other than the variable name changes (again, the code that produces the bar chart and scatterplot is highlighted) and a slightly smaller marker font. We also need to move the legend to its own block (or put it in a sidebar).

```
proc template;
define statgraph work.heartdata_stackcluster;
begingraph /;
layout datalattice columnvar=sex/
rowaxisopts=(Label=' ' offsetmin=0)
columnaxisopts=(offsetmin=0.2 offsetmax=0.2 display=(ticks tickvalues));
layout prototype /;
BarChartParm X=chol_status_n Y=PctN_110 / primary=true Group=status
NAME="Status" groupdisplay=stack barwidth=0.6;
Scatterplot x=chol_status_n y=yScatter / group=status
markercharacter=PctN_110 labelstrip=true
markercharacterattrs=(size=16 weight=bold color=white);
endlayout;
endlayout;
layout globallegend;
DiscreteLegend "Status"/ title="Patient Status";
endlayout;
endgraph;
end;
run;
```

The usual SGRENDER code follows. We do not show it here, but you could easily use dynamic variables in place of the hardcoded variables if you want to be able to flexibly use this template with any dataset containing three variables to use in this fashion.

```
proc sgrender template=work.heartdata_stackcluster
  data=work.heartfreq_chol_forhl;
run;
```

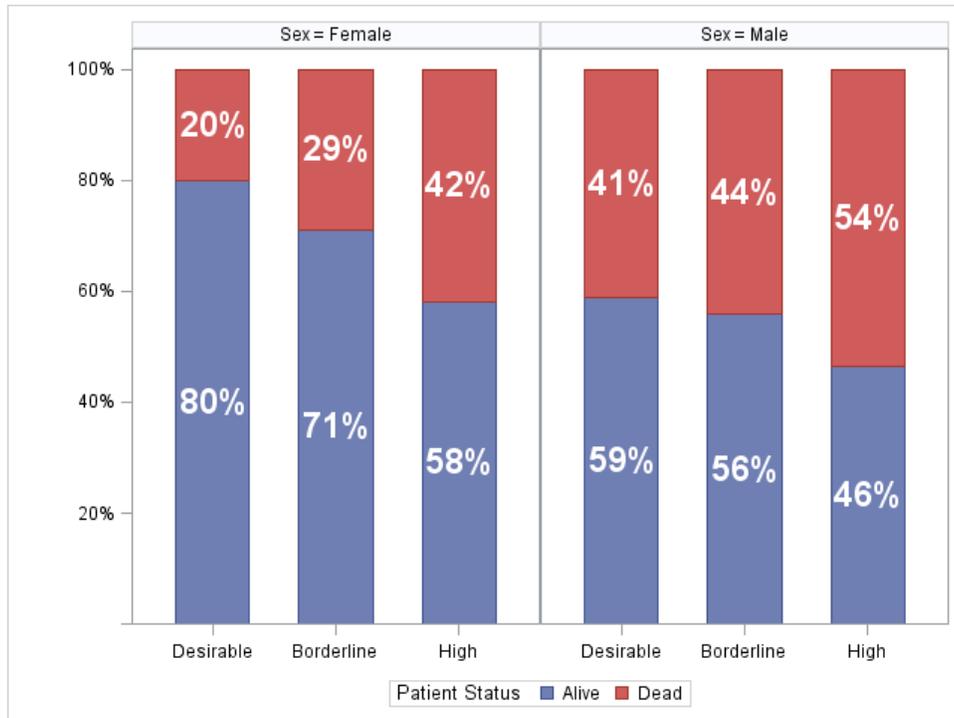


Figure 4. Initial Stacked/Clustered Bar Chart

This is not entirely ideal, however; while it does display our data nicely, it is not as neatly packed as the clustered bar charts that come directly from SGPLOT, and it has extra borders that are not really needed.

To do that, we have to control a few specific elements. First, we expand the bar widths to 0.9 instead of 0.6, and reduce the space available using the offsetmin and offsetmax, which effectively define the padding on the axis (in this case, the column axis, which is the x axis). 0.2 on either side matched the bar padding (0.6 means 0.2 on each side); upping that to 0.3 on either side combined with the increase to 0.9 of the bar width creates a nice spacing between our clusters of bars, and tightly grouped bars.

Second, we remove the unnecessary border by adding the line WALLDISPLAY=NONE to the layout prototype. (These borders come from the individual panels, which are defined by the prototype.) We also move the panel column header to the bottom, remove the "Sex=", and turn the opacity off, which makes the background color go away from the headers. We cannot remove the borders from the panel column headers without modifying the style template, which is beyond the scope of this paper, but even with the header borders we have a nice looking clustered graph now.

The updated code is listed below, with the changes highlighted.

```

proc template;
  define statgraph work.heartdata_stackcluster;
    beginingraph /;
      layout datalattice columnvar=sex/
        rowaxisopts=( Label=' ' offsetmin=0)
        columnaxisopts=(offsetmin=0.3 offsetmax=0.3 display=(ticks tickvalues))
          headerlabeldisplay=value headeropaque=false columnheaders=bottom;
      layout prototype /walldisplay=none;
      BarChartParm X=chol_status_n Y=PctN_110 / primary=true Group=status
        NAME="Status" groupdisplay=stack barwidth=0.9;
      Scatterplot x=chol_status_n y=yScatter / group=status
        markercharacter=PctN_110 labelstrip=true
        markercharacterattrs=(size=16 weight=bold color=white);
      endlayout;
    endlayout;
  layout globallegend;
    DiscreteLegend "Status"/ title="Patient Status";
  endlayout;
endgraph;
end;
run;

```

This produces our final chart, which clearly conveys the information without too much difficulty coding the chart – and is data-driven, so any changes in our data will automatically be properly reflected.

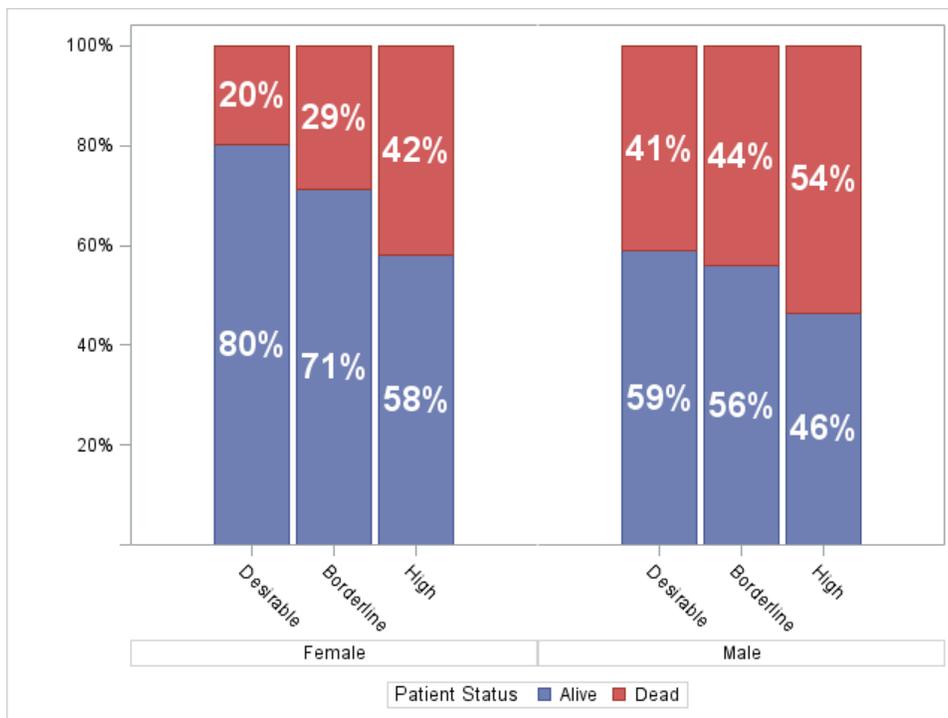


Figure 5. Final Stacked/Clustered Bar Chart

CONCLUSION AND FURTHER THOUGHTS

This just breaks the surface of what you can do with combining charts with scatterplots to annotate details onto the charts. Many other plot types can include scatterplots for annotation; any plot that can be made with a “basic” plot type will do. For more information, see Graphically Speaking (<http://blogs.sas.com/content/graphicallyspeaking>), Sanjay Matange’s blog that covers many interesting uses of SAS ODS graphics, and was very helpful in working out how to do many of the topics covered in this paper.

NEW OPTIONS IN 9.4 M2 IN DETAIL

- **SEGLABEL**

9.4 M2 introduces the SEGLABEL option to PROC SGPLOT and GTL, which allows the user to specify a variable to use as a label on each segment of a stacked bar chart. This applies to both VBAR and VBARPARM (and their H equivalents), and is specified as an option on the plot statement. In addition, SEGLABELFITPOLICY, SEGLABELATTRS, and SEGLABELFORMAT were added to improve the look of the labels.

Example:

```
proc SGPLOT data=sashelp.heart;  
  vbar sex /stat=percent group=status groupdisplay=stacked seglabel;  
run;
```

That would generate by itself a similar plot to Figure 2, with substantially less work.

- **VBARPARM GROUP=STACKED**

While we noted that VBARPARM did not allow GROUP=STACKED as an option above, as of 9.4 M2 this is no longer true; that is now an option. As such, the HIGHLOW plot is not needed, and if you need to overlay a stacked bar chart with text, you now can use VBARPARM just as you would VBAR to generate a stacked bar chart.

- **TEXT PLOT TYPE**

This is a plot type functionally similar to the SCATTER plot we use here, but focused on text; it gives significantly greater control over the text you display, including options like backlighting and rotated text. While it was not available to test prior to this paper going to press, you should be able to use either SCATTER or TEXT in a similar fashion and achieve similar results. See the SAS documentation on SGPLOT in 9.4 for more details.

REFERENCES

- Matange, Sanjay, "Bar charts with Stacked and Cluster Groups"; September 7, 2013;
<http://blogs.sas.com/content/graphicallspeaking/2013/09/07/bar-charts-with-stacked-and-cluster-groups/>
- Matange, Sanjay, "Stacked Bar Chart with Segment Labels", September 20, 2013;
<http://blogs.sas.com/content/graphicallspeaking/2013/09/20/stacked-bar-chart-with-segment-labels/>
- Matange, Sanjay, "G100 with SGPLOT", April 6, 2014;
<http://blogs.sas.com/content/graphicallspeaking/2014/04/06/g100-with-SGPLOT/>

All URLs in References and Recommended Reading verified as of September 5, 2014.

ACKNOWLEDGMENTS

First, I'd like to thank Sanjay Matange of SAS Institute, both for providing many of the concepts for this paper and for assistance with the original project that was the genesis of this paper in the fall of 2013.

I'd also like to thank Richard Hebel, John Vidmar, Gordon Hanson, and the team at Ipsos, who provided me an opportunity to learn SAS in an excellent environment, and the room to grow in my own direction.

Finally, I'd like to thank Jeff Vose, Al Crego, Felicia LeClere, and the team at NORC at the University of Chicago for supporting me while I wrote this paper and for continuing to provide me the resources and opportunities I need to further my SAS development.

RECOMMENDED READING

Some additional papers showing alternate methods to do some of the things in this paper. Also, we recommend following Sanjay's blog, Graphically Speaking (<http://blogs.sas.com/content/graphicallyspeaking/>) if you are interested in learning more about using ODS Graphics and data visualization.

- Matange, Sanjay, "Tips and Tricks for Clinical Graphs using ODS Graphics", PharmaSUG 2011; <http://www.pharmasug.org/proceedings/2011/SAS/PharmaSUG-2011-SAS-AD01.pdf>
- Wicklin, Rick, "Construct a stacked bar chart in SAS where each bar equals 100%", April 8, 2014; <http://blogs.sas.com/content/iml/2014/04/08/construct-a-stacked-bar-chart-in-sas-where-each-bar-equals-100/>

If you need a quick start using the Graph Template Language, here are a few options for getting started.

- Matange, Sanjay, 2008. "Introduction to the Graph Template Language", <http://support.sas.com/resources/papers/sgf2008/gtl.pdf>
- Cartier, Jeff, 2006. "A Programmer's Introduction to the Graphics Template Language," <http://www2.sas.com/proceedings/sugi31/262-31.pdf>
- Matange, Sanjay, 2013. *Getting Started with the Graph Template Language in SAS*, SAS Institute.

For more information about the new options in SAS 9.4 M2, Sanjay explains the new features in two blog posts:

- Matange, Sanjay, "New Graphics Features in SAS 9.4M2 – Part 1", August 24, 2014; <http://blogs.sas.com/content/graphicallyspeaking/2014/08/24/new-graphics-features-in-sas-9-4m2-part-1/>
- Matange, Sanjay, "New Graphics Features in SAS 9.4M2 – Part 2", August 29, 2014; <http://blogs.sas.com/content/graphicallyspeaking/2014/08/29/new-graphics-features-in-sas-9-4m2-part-2/>

CONTACT INFORMATION

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

Joe Matise
NORC at the University of Chicago
65 E Monroe
Chicago, IL 60604
(312) 759-4296
matisejoe@gmail.com

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.