

Annotate your SGPLOT Graphs

Sanjay Matange, SAS Institute Inc.

ABSTRACT

The SG procedures provide you multiple plot statements to create many different kind of graphs. These plot statements can be used together in creative ways to build your graph. However, even with this ability to customize, there are times when you need more than what you can get using just the plot statements. You need a way to add custom information anywhere on the graph.

With SAS 9.3, the SG procedures support the ability to annotate the graph using data set based information. This annotation functionality is designed in a way similar to the annotate facility available with the SAS/GRAPH procedures. There are a few differences and enhancements. If you already know annotation from SAS/GRAPH, or if you are new to it, this paper will show you how to add custom annotations to your graphs.

INTRODUCTION

The SGPLOT procedure provides new, exciting way to create clinical graphs included SAS. Starting with SAS 9.3, all Statistical Graphics procedures are included in the BASE SAS. The SGPLOT procedure supports multiple plot statements that can be combined in numerous ways to create most of the graphs commonly used in the Clinical Research domain.

The plot statements supported by the SGPLOT procedure can be combined (with some restrictions) with other plot statements to create effective graphs. Here is an example of the Survival Plot, one of the most frequently used graphs in the clinical domain.

SGPLOT Procedure code for Survival Plot:

```
title 'Product-Limit Survival Estimates';
proc sgplot data=SurvivalPlotData;
  step x=time y=survival / group=stratum lineattrs=(pattern=solid) name='s';
  scatter x=time y=censored / markerattrs=(symbol=plus) name='c';
  scatter x=time y=censored / markerattrs=(symbol=plus) group=stratum;
  keylegend 'c' / location=inside position=topright;
  keylegend 's';
run;
```

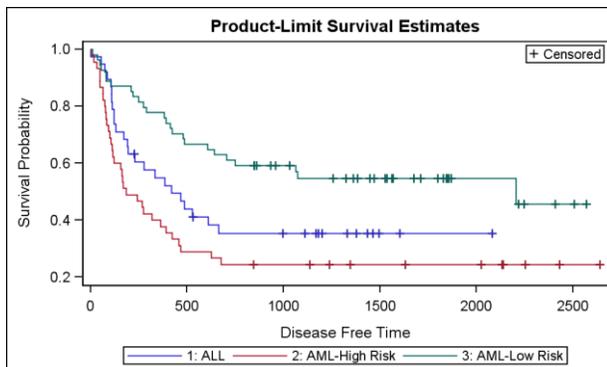


Figure 1 - Survival Plot using SGPLOT procedure

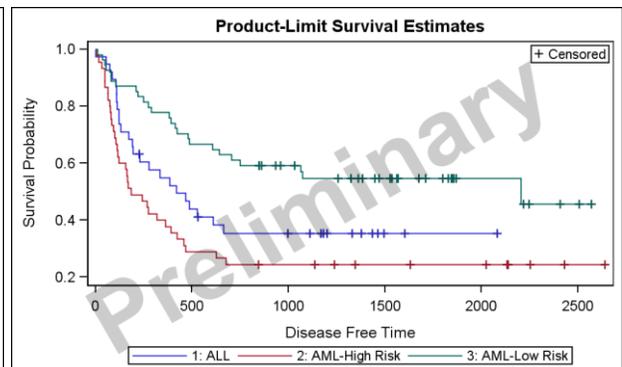


Figure 2 – Survival Plot with annotation.

THE NEED FOR ANNOTATION

Figure 1 above shows the result of the program. This graph uses the survival data set created by the LIFETEST procedure as shown in the Appendix. Often, there is a need to embellish the graph for cosmetic reasons. Or, there is a need to add some important information to the graph which cannot be done using the plot statements supported by the procedures. The graph in Figure 2 shows one such a use case. We have placed a simple watermark on the graph. Such embellishment of the graph can be done by using the SG annotation facility starting with SAS 9.3.

OVERVIEW OF SG ANNOTATION

Annotations with the SG procedures work in a way very similar to the SAS/GRAPH annotate facility. So, if you are familiar with annotating with SAS/GRAPH, learning SG annotation will be easy. Now, clearly there are some differences, necessitated by the underlying architecture of the new system, but the general idea is the same.

Annotation objects to be placed in the graph are defined in a special SG annotation data set. Each annotation, except POLYGON and POLYLINE, is defined by one observation in the data set. The annotation data set uses specific column names such as FUNCTION, X1, Y1, TEXTSIZE and so on to specify the annotation object.

The key column is FUNCTION. This determines the type of the annotation object placed on the graph. Each function normally needs the location in the graph where the annotation is to be placed. This information is usually provided using the X1, Y1, X2 and Y2 columns that are of type numeric.

Some functions like TEXT need only one set of coordinates, while other functions may need two sets, one for start point and one for the end point. Additional information to draw the object is needed based on the type of the object. This can be provided using other column names such as TEXTSIZE, TEXTCOLOR, LINETHICKNESS and so on. Most of these have default values in the system, so they are only necessary if you want to customize the object. The system supports the ten different functions listed in Figure 3 below.

ARROW	Draws an arrow from (x1, y1) to (x2, y2).
IMAGE	Draws an image at (x1, y1) location from an image file.
LINE	Draws a line from (x1, y1) to (x2, y2).
OVAL	Draws an oval or circle centered at (x1, y1).
POLYGON	Specifies the beginning point (x1, y1) of a polygon.
POLYLINE	Specifies the beginning point (x1, y1) of a polyline.
POLYCONT	Continues adding points (x1, y1) to a polygon or polyline.
RECTANGLE	Draws a rectangle or square annotation centered at (x1, y1).
TEXT	Places text in the graph output starting at (x1, y1)
TEXTCONT	Continues a text string from the end of the previous string..

Figure 3

GETTING STARTED WITH TEXT ANNOTATION

Let us examine the example shown in Figure 2 above. This is one of the simplest and useful case of annotation. In order to create the annotation of a watermark on the graph, we have to follow these steps:

1. Create a data set that contains the necessary observations to define the annotation.
2. Use the SGANNO option on the SGPLOT procedure statement to specify the annotation data set.

We create a data set called Anno_Watermark_Data as shown in Figure 4 below.

function	width	rotate	textsize	textweight	transparency	label
text	100	30	60	bold	0.8	Preliminary

Figure 4 – Annotation data set for simple annotation

The only required columns are FUNCTION, and LABEL. However, since we want a large, bold and transparent font, we provide the TEXTSIZE, TEXTWEIGHT and TRANSPARENCY columns. Since we want to rotate the text string, we provide the ROTATE column. The WIDTH column is the wrapping width within which the text is to be drawn, and we set it to 100 percent (default). Now, we provide the name of this data set on the procedure statement as follows:

```
proc sgplot data=SurvivalPlotData sganno= Anno_Watermark_Data;
```

Now, the procedure will draw the requested objects in the graph as shown in Figure 2 using various default settings. In this case, we have only one rotated text string to be drawn, so we need only one observation in the data set. In this simple example, the position of the text was not provided, and the default value of (50, 50) was used.

DRAWSPACE

One of the key concepts to understand for drawing of the annotations is the DRAWSPACE. This is a combination of the location where the annotation is being drawn and also the units of measurement. The system supports four regions as listed below and illustrated in Figure 5.

1. GRAPH - The whole graph (default).
2. LAYOUT - The region inside the Graph excluding the padding around the edge.
3. WALL - The region bounded by the axes lines.
4. DATA - The region occupied by the data, excluding the axis offsets.

The coordinate origin (0, 0) for each of these regions is the lower left corner of the corresponding rectangular region, with positive x values going to the right and positive y values going up. Coordinates can be provided in one of the following units

1. PERCENT - For all regions (default).
2. PIXEL - For all regions.
3. VALUE - For DATA region only.

The DRAWSPACE is a text string that is the combination of the region and the units. The default draw space is GRAPHPERCENT. Other draw spaces can be any of the combinations like LAYOUTPIXEL, DATAPERCENT and so on. When using PERCENT or PIXEL, the coordinates are always numeric, and provided by the X1, Y1, X2 and Y2.

The unit VALUE can only be used in the DATA region. So, DATAVALUE is allowed, but not GRAPHVALUE. The DATAVALUE draw space allows you to draw in the data region using axis coordinates in terms of the actual data on each axis. This is useful to align your annotations with the plot data.

For DATAVALUE, the coordinates provided depend on the axis, and can be numeric or discrete (character). If the axis is numeric, then the value is read from the X1, Y1, X2 and Y2 (numeric) columns. If the axis is discrete, with values such as 'A', 'B', then the coordinates for the annotation location are read from the XC1, XC2, YC1 and YC2 columns, using A or B as the value.

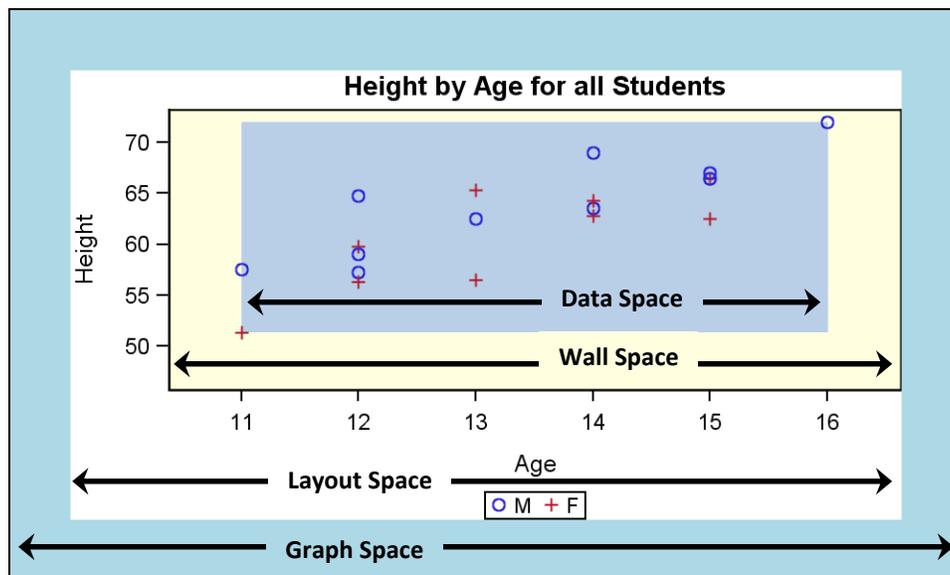


Figure 5 – Draw spaces.

For the annotation shown in Figure 2, the default DRAWSPACE of GRAPHPERCENT is used since it is not provided in the annotation data set. Also, the default ANCHOR point for the text string is in the center of the text bounding box. Hence the text is placed in the center of the graph, rotated 30 degrees counterclockwise around the center.

DRAWSPACE value applies to both X and Y axes. However, it is not necessary the both X and Y must use the same draw space setting. You can have the x values in DATAVALUE space and y values in GRAPHPERCENT space. Or any such combination is allowed. As a matter of fact, one end of the line can be in a different draw space from the other. For such cases, you can use the X1SPACE, X2SPACE, Y1SPACE or Y2SPACE options. We will see the use of such combinations later in the presentation.

SURVIVAL PLOT WITH SUBJECTS AT RISK

The survival plot shown in Figure 1 is incomplete, and hence marked "Preliminary". An important feature of the survival plot is the display of the subjects at risk at intervals along the time axis. There are multiple ways of adding the at-risk information to the graph. The "SurvivalPlotData" data set produced from the LIFETEST procedure (see Appendix) is shown below. We have already used the TIME, SURVIVAL and STRATUM columns to create the graph shown in Figure 1. Then we added the CENSORED points to the curves, and displayed the legends. Now, let us use the TATRISK and ATRISK columns to display the subjects at-risk in the graph, using SG annotation

Time	Survival	AtRisk	Event	Censored	tAtRisk	Stratum	StratumNum
0	1.00000	38	0	.	.	1: ALL	1
0	.	38	.	.	0	1: ALL	1
1	0.97368	38	1	.	.	1: ALL	1
55	0.94737	37	1	.	.	1: ALL	1
74	0.92105	36	1	.	.	1: ALL	1

Figure 6 – Survival Plot Data

Traditionally, the survival plots commonly seen in the industry include an axis aligned table of the subjects at risk at the bottom of the graph, below the stratum legend. As you can see in Figure 1, there is no space in the graph to display this information. So, the first step will be to create some room for these numbers. We do this by adding a pad to the bottom of the graph. We do this by specifying a bottom pad of 15% by using the PAD option on the procedure statement:

```
proc sgplotdata=SurvivalPlotData pad=(bottom=15pct);
```

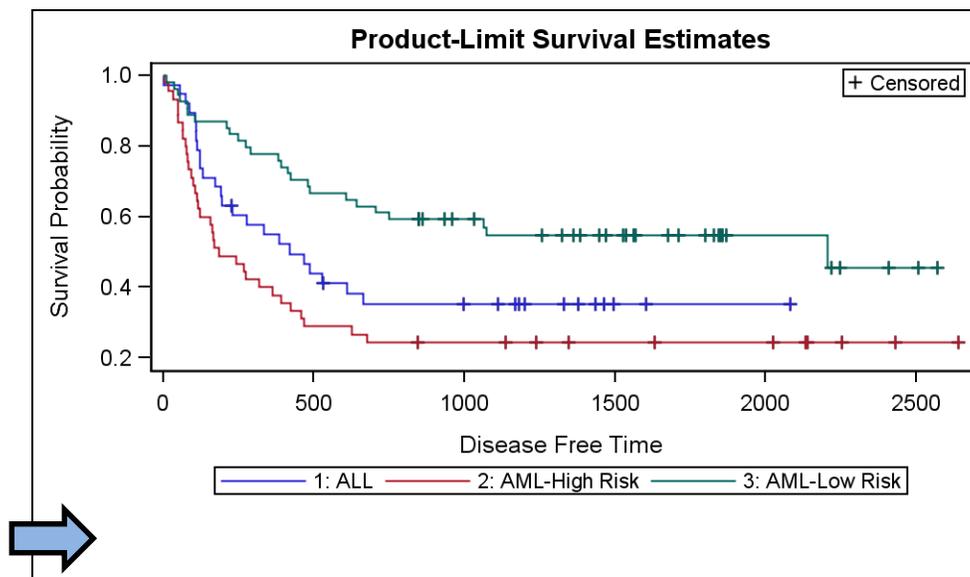


Figure 7 – Creating space for annotation

Here is the annotation data set needed to place a text string below the legend horizontally centered on the wall region.

function	x1space	y1space	x1	y1	width	textsize	textweight	label
text	wallpercent	graphpercent	50	8	100	12	bold	We will add the At-Risk table here

Figure 8 – Data set Anno_Text

Here is the code and the graph created by using the annotation data set shown in Figure 9. Note, now the annotated text string is displayed horizontally centered on the wall area, and vertically below the legend.

```
proc sgplot data=SurvivalPlotData pad=(bottom=15pct) sganno=Anno_Text;
```

This is done by using X1SPACE=WALLPERCENT and Y1SPACE=GRAPHPERCENT. The Y1 setting is not strictly required as it will be the default data space if not specified. The coordinate X1=50% (of the wall) and Y1=8% (of the graph height). The default ANCHOR is still center, so this annotated text string is located where we want it.

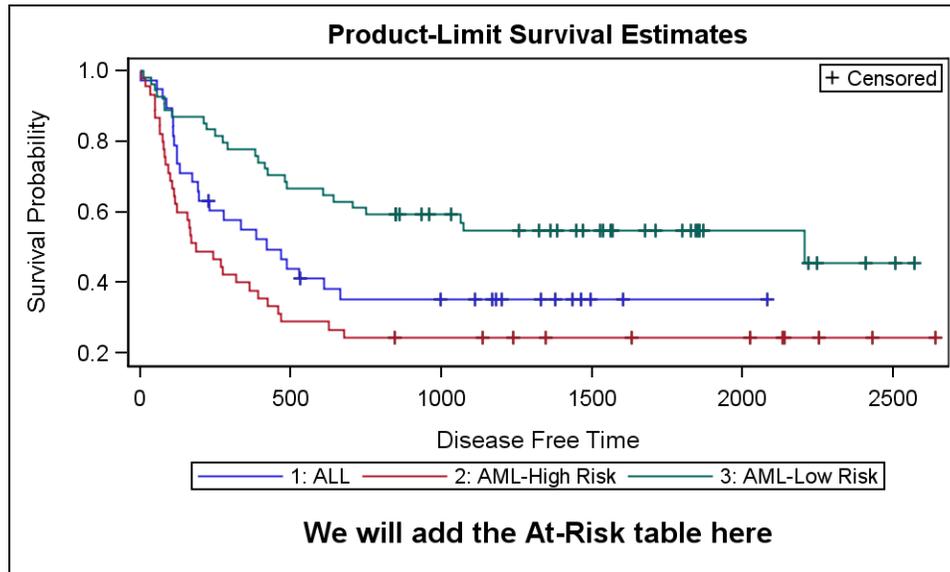


Figure 9 – Writing text at the bottom of the graph

Now, that we know how to place the text string where we want, let us display the actual at-risk numbers in this space. This time, we will create an annotation data set to place three rows of numbers from the ATRISK columns of the original data, along the x axis at the TATRISK location. The three rows are one per value of the STRATUM variable, at different levels from the bottom of the graph.

Here is the code to create this data set and the data set itself.

```
data Anno_At_Risk;
  length function $10;
  set SurvivalPlotData;
  by stratum;
  retain id 0 function x1space y1space textsize textweight width y1;

  if first.stratum then do;
    id+1;
    function='text'; x1space='datavalue'; y1space='graphpercent';
    textsize=8; textweight='normal'; width='10'; y1=(id-1)*4+3;
  end;

  if tatrisk ne . then do;
    x1=tatrisk; label=put(atrisk, 3.0); output;
  end;
run;
```

In the program above, we set SurvivalPlotData by stratum. For each value of stratum, we change the y location of the annotation. For each ATRISK value, we set the x location to TATRISK with X1SPACE=DATAVALUE.

function	x1space	y1space	x1	y1	width	textsize	textweight	label
text	datavalue	graphpercent	0	3	10	8	normal	38
text	datavalue	graphpercent	500	3	10	8	normal	16
text	datavalue	graphpercent	1000	3	10	8	normal	11
text	datavalue	graphpercent	1500	3	10	8	normal	2
text	datavalue	graphpercent	2000	3	10	8	normal	1
text	datavalue	graphpercent	2500	3	10	8	normal	0
text	datavalue	graphpercent	0	7	10	8	normal	45

Figure 10 – Data Set Anno_At_Risk

The data set is shown in Figure 10 above. Note the draw space for x values is DATAVALUE, and each X1 value is the TATRISK value, the same as was used for the x axis for the survival curves. The draw space for Y values is GRAPHPERCENT, and the values are 3%, 7% and 11%. The text LABEL is the number of subjects at risk from the ATRISK column. We have written to data to the Anno_At_Risk data set only when the TATRISK values are non-missing. Now, we run the same code by specifying Anno_At_Risk data set to create this graph:

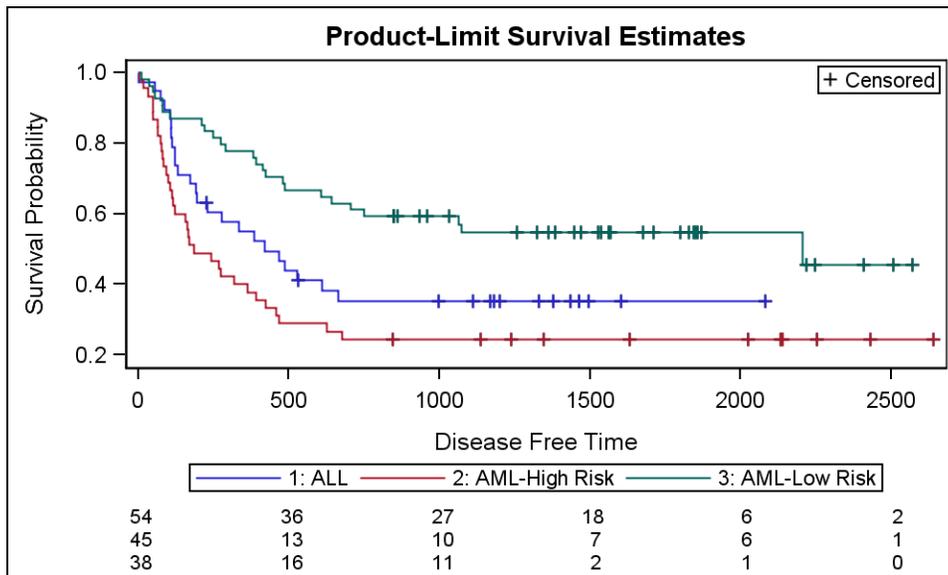


Figure 11 – Survival plot with subjects at risk values

In Figure 11, the at-risk values are displayed at the right locations below the legend, but we can't correlate the values with the stratum levels. To do that, we are going to add the display of stratum level, at the same y value as the at risk values but positioned horizontally just to the left of the Y axis. Here is the additional SAS data step code to generate the stratum labels:

```

if last.stratum then do;
  function='text'; x1space='wallpercent'; y1space='graphpercent';
  textsize=7; textweight='bold'; anchor='right';
  width=30; widthunit='percent';
  x1=-2; y1=(id-1)*4+3; label=stratum; output;
end;

```

Note, in the code snippet, the x1 space is WALLPERCENT, X1=-2% and ANCHOR=RIGHT. Coordinate values can be less than 0 or more than 100. Such values are not clipped, but are still drawn relative to the origin of the region. So, in this case the STRATUM labels will be drawn with their right edge of the text is 2% to the left of the Y axis.

The data set is shown in Figure 12, and the graph in Figure 13. We have also set a PAD=(LEFT=15pct) to provide space for the STRATUM level labels.

function	x1space	y1space	x1	y1	width	widthunit	textsize	textweight	anchor	label
text	datavalue	graphpercent	0	3	20	percent	7	normal	center	38
text	datavalue	graphpercent	500	3	20	percent	7	normal	center	16
text	datavalue	graphpercent	1000	3	20	percent	7	normal	center	11
text	datavalue	graphpercent	1500	3	20	percent	7	normal	center	2
text	datavalue	graphpercent	2000	3	20	percent	7	normal	center	1
text	datavalue	graphpercent	2500	3	20	percent	7	normal	center	0
text	wallpercent	graphpercent	-2	3	30	percent	7	bold	right	1: ALL
text	datavalue	graphpercent	0	7	20	percent	7	normal	center	45

Figure 12 – Annotation data set with STRATUM labels

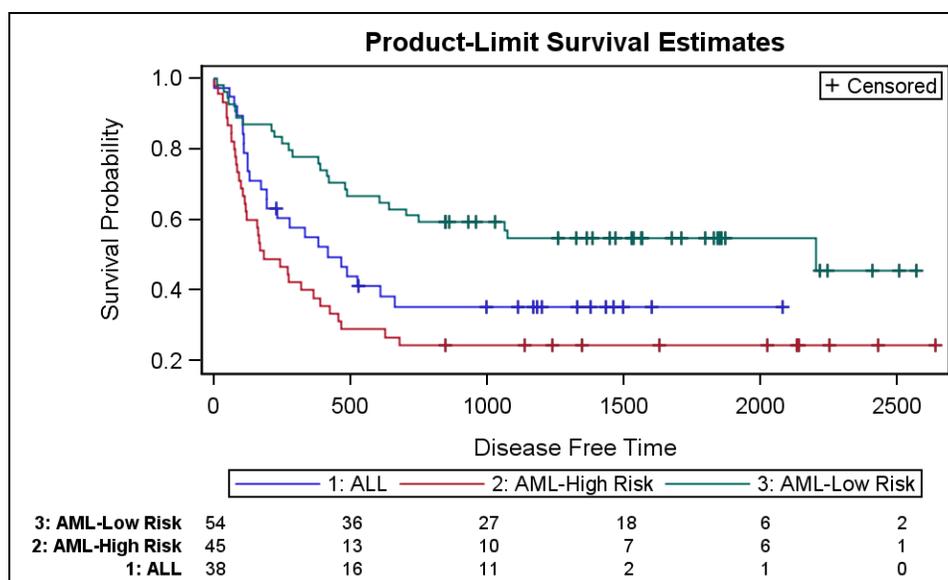


Figure 13 – Survival plot with STRATUM labels

Color-coding the at-risk values and the STRATUM levels in the same way as the survival curves can make this graph much easier to consume. Figure 14 shows the annotation data set, and Figure 15 shows the final graph.

The data in the SurvivalPlotData is sorted by STRATUM and TIME. STRATUM is used as the GROUP variable for the survival curves (Series). So, GRAPHDATA style elements from the active style are used in sequence for each of the levels of the STRATUM variable. This are GRAPHDATA1, GRAPHDATA2 and GRAPHDATA3. So, to correctly match the colors of the annotated at-risk values to the survival curves, we have to specify the TEXTCOLOR value correctly for each row of the table.

The code below shows how we have built the correct text string for the TEXTCOLOR column by concatenating the strings and the index of the group value.

```

data Anno_At_Risk_Label_Color;
  length function $10 x1space $12 label $18;
  set SurvivalPlotData;
  by stratum;
  retain id 0 function x1space y1space textcolor textsize
    textweight anchor width widthunit y1;

  if first.stratum then do;
    id+1;
    textcolor="GraphData" || put(id, 1.0) || ":contrastcolor";
    function='text'; x1space='datavalue'; y1space='graphpercent';
    widthunit='percent'; textsize=7; textweight='normal';
    anchor='center'; width=20; y1=(id-1)*4+3;
  end;

  if tatrisk ne . then do;
    x1=tatrisk; label=put(atrisk, 3.0); output;
  end;

  if last.stratum then do;
    textcolor="GraphData" || put(id, 1.0) || ":contrastcolor";
    function='text'; x1space='wallpercent'; y1space='graphpercent';
    textsize=7; textweight='normal'; anchor='right';
    width=30; widthunit='percent';
    x1=-2; y1=(id-1)*4+3; label=stratum; output;
  end;
run;

```

function	x1space	y1space	x1	y1	width	widthunit	textcolor	textsize	textweight	anchor	label
text	datavalue	graphpercent	0	3	20	percent	GraphData1:contrastcolor	7	normal	center	38
text	datavalue	graphpercent	500	3	20	percent	GraphData1:contrastcolor	7	normal	center	16
text	datavalue	graphpercent	1000	3	20	percent	GraphData1:contrastcolor	7	normal	center	11

Figure 14 – Annotation data set with stratum labels colored by stratum

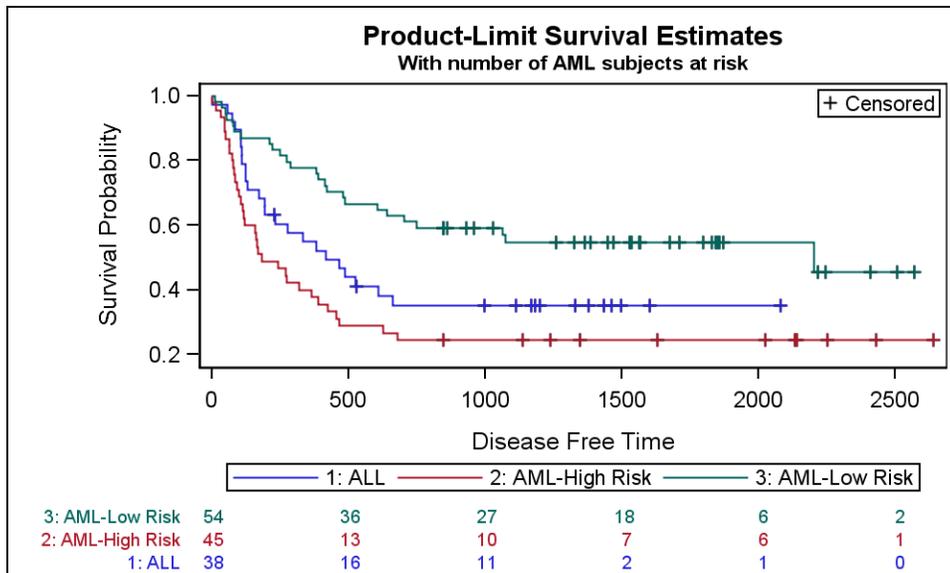
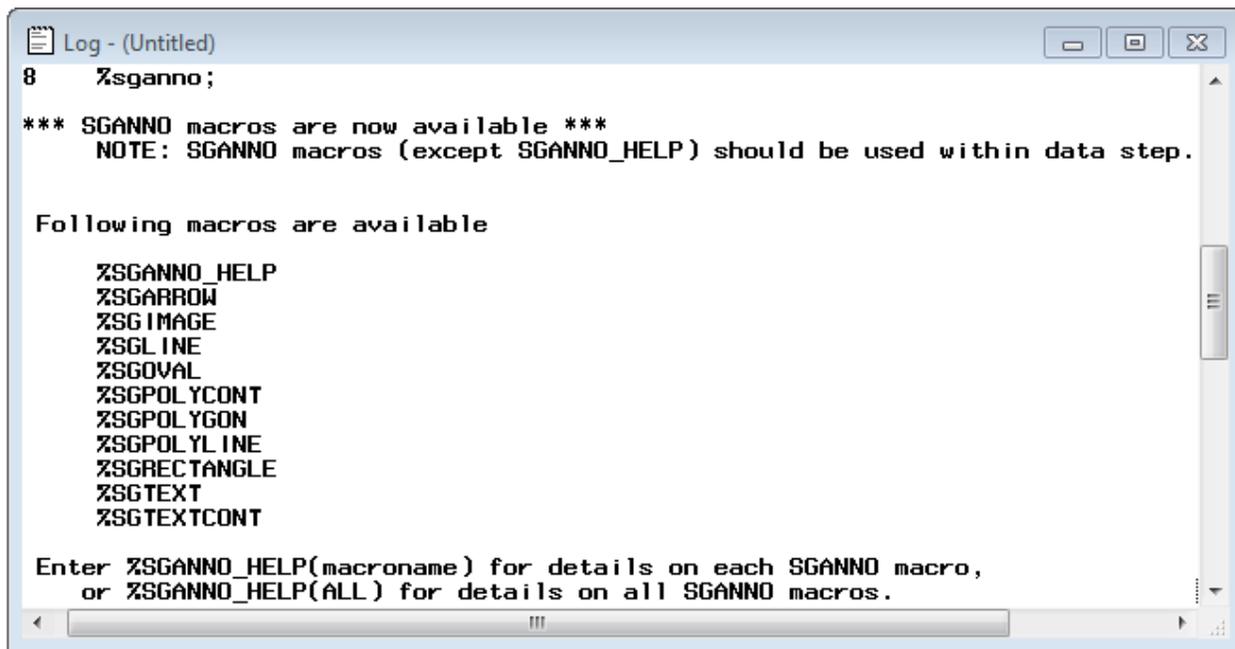


Figure 15 – Final Survival Plot with risk values colored by stratum

%SGANNO – ANNOTATION MACROS

SG annotation macros are included with **SAS 9.4**. These are a set of macros that make it easier to write the observation needed for the annotations to the annotation data set. To use these macros, you have to first initialize the set of macros by submitting the "%sganno;" statement in the program window. When you submit this code, a log message is displayed informing you the names of the macros that are available, and how to get help on each macro. Each macro is also documented in the SAS user documentation.



```
Log - (Untitled)
8 %sganno;

*** SGANNO macros are now available ***
NOTE: SGANNO macros (except SGANNO_HELP) should be used within data step.

Following macros are available

%SGANNO_HELP
%SGARROW
%SGIMAGE
%SGLINE
%SGOVAL
%SGPOLYCONT
%SGPOLYGON
%SGPOLYLINE
%SGRECTANGLE
%SGTEXT
%SGTEXTCONT

Enter %SGANNO_HELP(macroname) for details on each SGANNO macro,
or %SGANNO_HELP(ALL) for details on all SGANNO macros.
```

Figure 16 – Initializing SG annotation macros in SAS 9.4

Now, we can use the %SGTEXT macro to create the overlay water mark like we had done in Figure 2. Here is the code for this case:

```
/*--Initialize the SG annotation macros--*/
%sganno;

/*--Create annotation data set using SG Anno Macros--*/
data Watermark_94;
  %sgtext(label='Preliminary', textsize=60, textweight='bold', rotate=30,
    width=100, transparency=0.6);
run;

/*--Distribution Plot with Watermark--*/
title 'Distribution of Cholesterol';
proc sgplot data=sashelp.heart sganno=Watermark_94;
  histogram cholesterol;
run;
```

The result of this program is included in Figure 17. Note in the macro call above, we have only provided the information that is necessary, accepting the default settings for all the other options.

UNICODE text, superscripts and subscripts can also be specified by using the ODS escape chars and the special {sup 'text'} and {sub 'text'} functions.

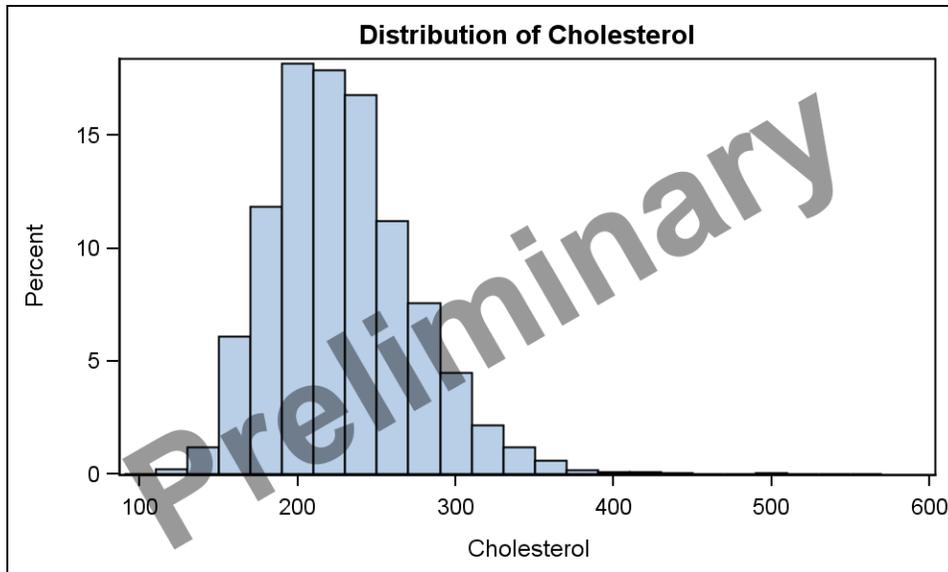


Figure 17 – Using %SGAnno macros

TEXTCONT is a useful feature to create rich text. This can be used immediately after a TEXT function, and the provided text is appended to the previous text with new attributes such as TEXTSIZE, TEXTWEIGHT and TEXTCOLOR. Other attributes from the TEXT function are still valid, and the entire text string is properly wrapped within the WIDTH, and then rotated as one block, as shown in Figure 18.

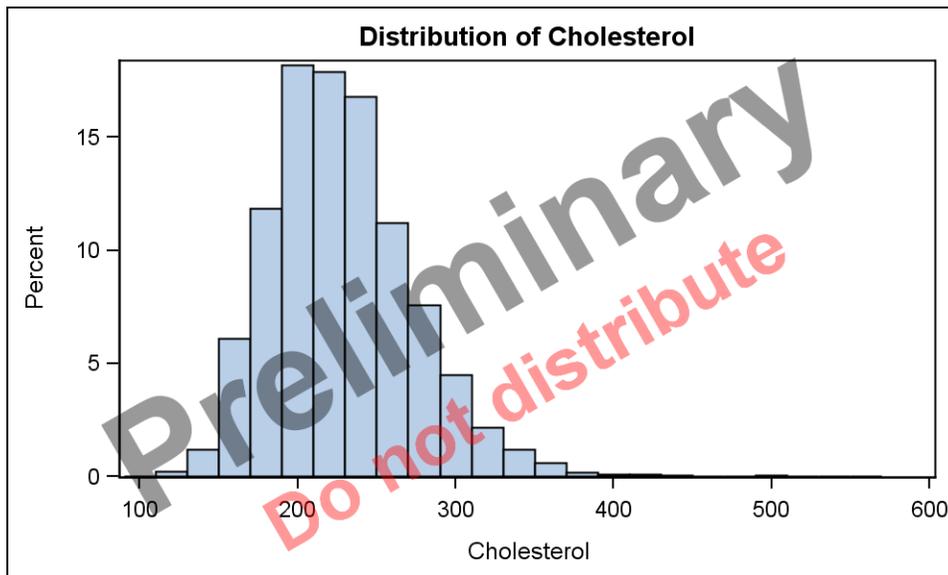


Figure 18 – Using text continuation with rich text attributes, with wrapping and rotation.

Here, we have added the %SGTEXTCONT macro statement to add a second line of text to the watermark. The second line uses different text attributes. The entire concatenated string is treated as one text object for text wrapping and rotation. With SAS 9.3, you can add a 2nd observation with the TEXTCONT function to get the same results.

```
data Watermark_94_RT;
  %sgtext(label='Preliminary', textsize=60, textweight='bold', rotate=30,
    width=90, transparency=0.6, justify='center');
  %sgtextcont(label='Do not distribute', textsize=30,
    textweight='bold', textcolor='red');
run;
```

LAYERING – FRONT OR BACK.

Most of the time, annotations are placed on top of the graphics. But, sometimes, you may want to place something behind the graphics, and then draw the graph on top of the annotation. This is controlled by the LAYER="BACK" | "FRONT" option available with all the annotation statements and in the SGANNO macros. LAYER="FRONT" is the default for all functions, hence we do not need to specify it most of the time.

One reason to place annotations in the back is for using image backgrounds. Note, when you place an annotation in the back, the graph itself will draw on top of the back annotation. The large, filled region of the "wall" will cover most of the background. In such cases, it may be necessary to turn off the wall fill color and borders. Also, you may need to adjust the colors of the axis values and grid lines.

IMAGE ANNOTATION FOR GRAPH BACKGROUNDS

In this section, we will place a background image behind the graph, using the IMAGE function. First, let us create the graph showing the average low and high temperatures for some fictitious city. The data set is shown on the right. A format is used for coloring the bars by group.

We have used a HIGHLOW plot in the program shown below to create a graph of this data as shown below in Figure 19. Note, we have suppressed the display of walls, borders and axis lines.

```
proc sgplot data=weather nowall noborder;
  format mean temp.;
  styleattrs datacolors=('lightblue' 'yellow'
    'orange' 'red');
  highlow x=month low=low high=high /
    type=bar group=mean dataskin=sheen
    lineattrs=graphoutlines
    fillattrs=(transparency=0.2);
  xaxis display=(nolabel noline noticks);
  yaxis display=(nolabel noline noticks) grid;
run;
```

Month	High	Low	Precip	Mean
Jan	49	30	4.46	39.5
Feb	53	32	3.53	42.5
Mar	61	40	4.46	50.5
Apr	71	48	2.98	59.5
May	78	57	4.03	67.5
Jun	84	65	4.06	74.5
Jul	88	69	4.35	78.5
Aug	86	68	4.30	77.0
Sep	80	62	4.27	71.0

Figure 19 – Temperature data

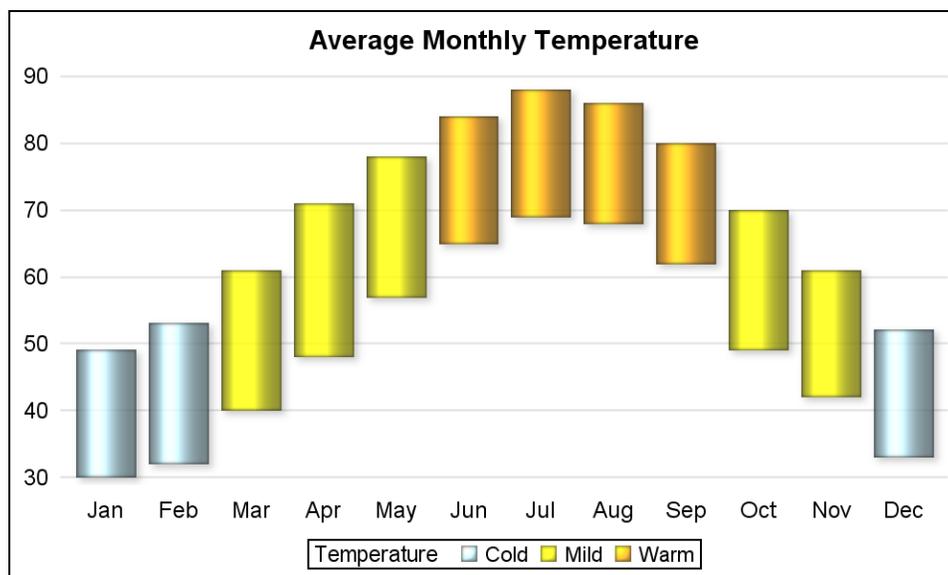


Figure 20 – Graph of average low and high monthly temperature

Now, let us use an image backdrop behind this graph. Before we go down this path, a **note of caution** about usage of images and other such "embellishments" in a graph. These are generally not recommended for analytical or clinical graphs. These are often considered as distractions by thought leaders in the analytical graphics community, and should be avoided. For "Info Graphics" or creation of "eye-catching" graphs, such techniques may be acceptable.

To place an image as a backdrop, we will create an annotation data set that uses the IMAGE function and provides the information needed to render the image in the graph. This includes, at a minimum, the fully specified path to the source image file to be used. This file must be accessible on the users file system. Here is the program and the data set.

```
data annoImage;
  length function $10;
  function='image'; height=100; width=100;layer='back';
  image='C:\Work\Images\Rainbow.png'; output;
run;
```

function	height	width	image	layer
image	100	100	C:\Work\Images\Rainbow.png	back

Figure 21 – Annotation data for image

Now, we provide the annotation data set to the SGPLOT procedure using the SGANNO option as follows to create the graph shown in Figure 22.

```
proc sgplot data=weather nowall noborder sganno=annoImage pad=0;
  format mean temp.;
  styleattrs datacolors=('lightblue' 'yellow' 'orange' 'red');
  highlow x=month low=low high=high / type=bar group=mean name='h'
    lineattrs=graphoutlines dataskin=matte fillattrs=(transparency=0);
  xaxis display=(nolabel noline noticks) valueattrs=(color=white);
  yaxis display=(nolabel noline noticks) valueattrs=(color=white)
    grid offsetmax=0.1;
  inset 'Average Monthly Temperature' / position=top
    textattrs=graphtitletext(color=white);
  keylegend 'h' / noopaque noborder valueattrs=(color=white);
run;
```

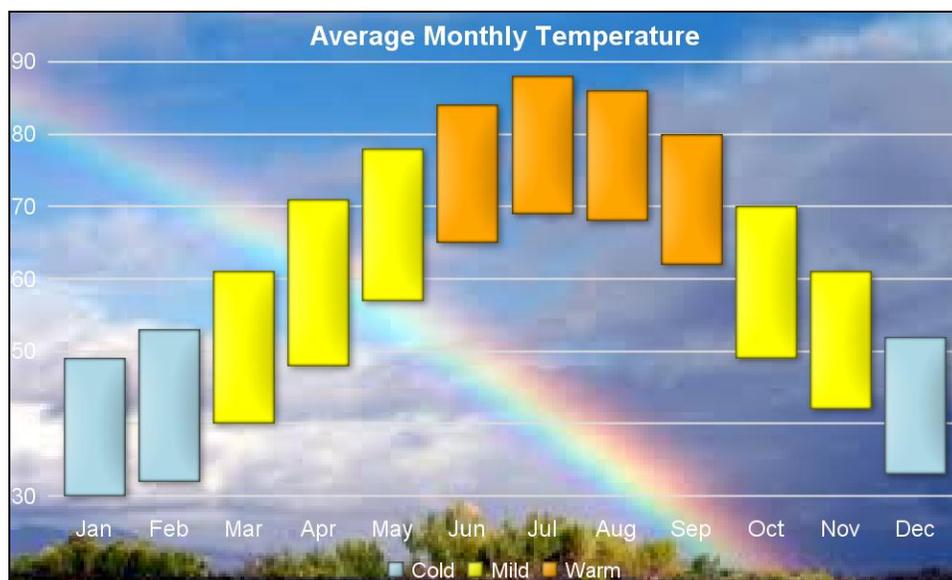


Figure 22 – Graph with image backdrop

Note, in the graph above, we have avoided usage of the TITLE statement. This is because the title is considered outside the GRAPH region, and the image will not fill in behind the title. To ensure the image fills the whole graph, we used an INSET instead, and set the text attributes to GRAPHTITLETEXT. Also, the axis values and gridlines are made white for aesthetic reasons.

Often, when using image backdrop, it is still recommended to fade out the image so it is less distracting. That can be done by using a transparency factor to fade the image with the background. Here is the data set and the corresponding graph.

function	height	width	transparency	image	layer
image	100	100	0.4	C:\Work\Images\Rainbow.png	back

Figure 23 - Annotation data set with transparent image

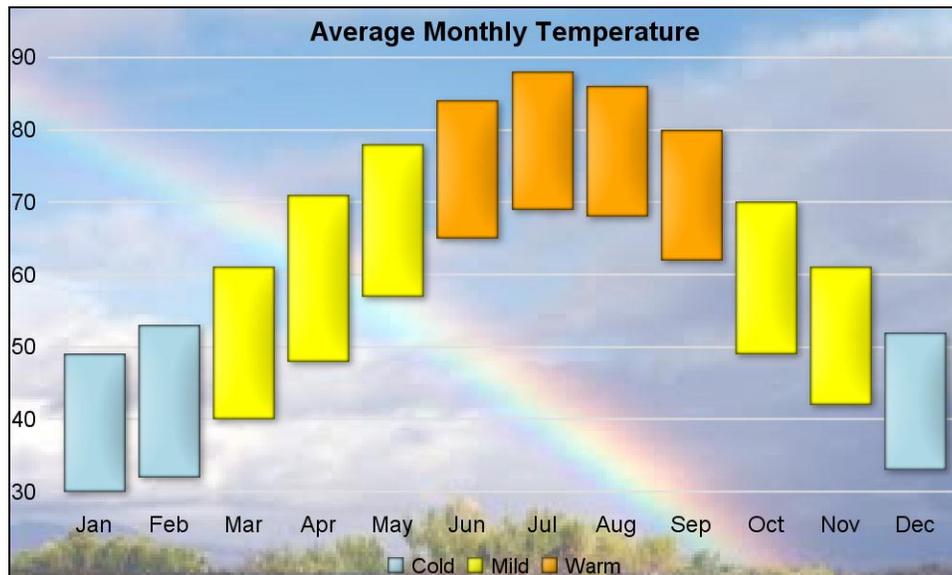


Figure 24 – Graph with faded image backdrop

OVALS, ARROWS AND OTHER ANNOTATIONS

In this last example of using annotation, we want to call out an observation that is outside the confidence ranges. In this example, I have slightly modified the weight of one of the observations so it falls outside the confidence limits.

Figure 25 shows the graph from this modified class data set along with the name of the student. We can see that Judy falls outside the confidence bands, and we want to highlight this observation in the graph.

Using the same techniques described earlier, we can create an annotation data set with the right functions to highlight this observation and display an explanatory note about this student.

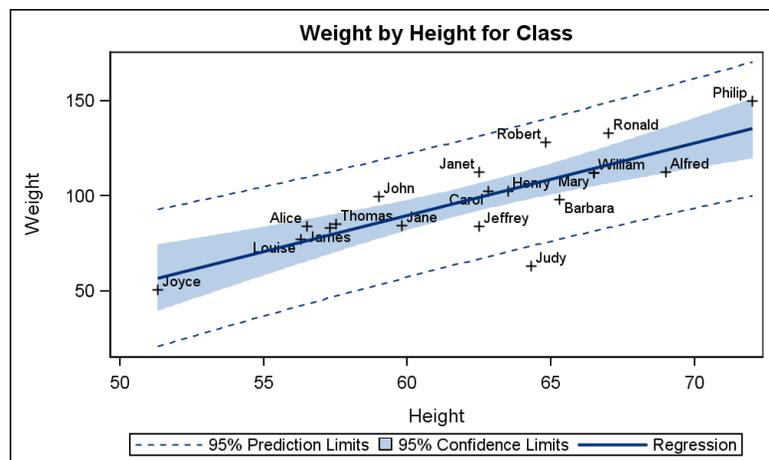


Figure 25 – Weight by height plot

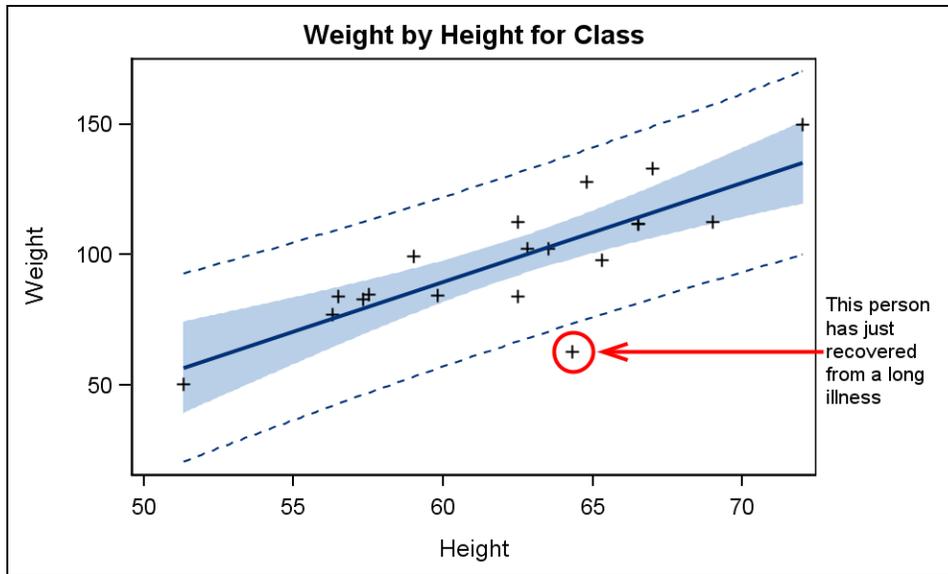


Figure 26 – Graph with explanatory annotation

The data set for this graph is displayed below. Since the data set has many columns, I have split it in two half.

Name	ht	Weight	ratio	function	drawspace	x1	y1	height	width	display	linecolor	linethickness
Judy	64.3	63	0.97978	oval	datavalue	64.3	63	10	6	outline	red	2
Judy	64.3	63	0.97978	arrow	datavalue	101.0	63	10	6	outline	red	2
Judy	64.3	63	0.97978	text	datavalue	101.0	63	10	20	outline	red	2

linecolor	linethickness	x2	y2	x2space	y2space	x1space	y1space	textsize	label	anchor
red	2
red	2	65.3	63	.	.	wallpercent	datavalue	.	.	.
red	2	wallpercent	datavalue	8	This person has just recovered from a long illness	left

Figure 27 – Annotation data set.

This annotation data set contains three annotation objects, an oval surrounding the marker, an arrow pointing to the marker from outside the graph, and a textual note regarding this observation. Note the use of the appropriate draw spaces for each of the X1, Y1, X2 and Y2 coordinates. The text is drawn so it wraps within a width of 25% with an anchor to the left.

For more examples of using annotations, see the paper "Now you can annotate your SG Procedure Graphs", SAS Global Forum 2011 by Dan Heath, SAS Institute Inc., Cary, NC.

CONCLUSION

SG annotation is a flexible and powerful facility to add important data to a graph or to embellish a graph for presentation. It may actually be possible to create an entire graph using SG annotation, but that is not its intended purpose. With SG procedures, the goal is to provide to you the plot statements and features you need to create a graph easily and in a manner that it is portable to other use cases. However, there are always some gaps that need custom treatment, and these can be filled using the annotation facility. SG annotation macros are also released with SAS 9.4, making the process easier.

REFERENCES

Heath, Dan. *Now you can annotate your SG Procedure Graphs*. Cary, NC: SAS Institute Inc. Proceedings of the SAS Global Forum 2011.

Matange, Sanjay and Dan Heath. 2011. *Statistical Graphics Procedures by Example: Effective Graphs Using SAS®*. Cary, NC: SAS Institute Inc.

Matange, Sanjay. *Graphically Speaking*. Available at <http://blogs.sas.com/content/graphicallyspeaking>.

RECOMMENDED READING

- *SAS ODS Graphics Procedures: Reference*
- *Knowledge Base / Focus Areas on Graphics* available at <http://support.sas.com/rnd/datavisualization/index.htm>

CONTACT INFORMATION

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

Sanjay Matange
SAS Institute, Inc.
SAS Campus Drive
Cary, NC 27513
(919) 531-6753
(919) 677-8000
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.

APPENDIX – SURVIVAL PLOT DATA FROM LIFETEST SAMPLE 49_2_1

```
proc format;
  value risk 1='ALL' 2='AML-Low Risk' 3='AML-High Risk';

/*--Data for proc LIFETEST sample 49_2_1--*/
data BMT;
  input Group T Status @@;
  format Group risk.;
  label T='Disease Free Time';
  datalines;
1 2081 0 1 1602 0 1 1496 0 1 1462 0 1 1433 0
1 1377 0 1 1330 0 1 996 0 1 226 0 1 1199 0
1 1111 0 1 530 0 1 1182 0 1 1167 0 1 418 1
1 383 1 1 276 1 1 104 1 1 609 1 1 172 1
1 487 1 1 662 1 1 194 1 1 230 1 1 526 1
1 122 1 1 129 1 1 74 1 1 122 1 1 86 1
1 466 1 1 192 1 1 109 1 1 55 1 1 1 1
```

```

1 107 1 1 110 1 1 332 1 2 2569 0 2 2506 0
2 2409 0 2 2218 0 2 1857 0 2 1829 0 2 1562 0
2 1470 0 2 1363 0 2 1030 0 2 860 0 2 1258 0
2 2246 0 2 1870 0 2 1799 0 2 1709 0 2 1674 0
2 1568 0 2 1527 0 2 1324 0 2 957 0 2 932 0
2 847 0 2 848 0 2 1850 0 2 1843 0 2 1535 0
2 1447 0 2 1384 0 2 414 1 2 2204 1 2 1063 1
2 481 1 2 105 1 2 641 1 2 390 1 2 288 1
2 421 1 2 79 1 2 748 1 2 486 1 2 48 1
2 272 1 2 1074 1 2 381 1 2 10 1 2 53 1
2 80 1 2 35 1 2 248 1 2 704 1 2 211 1
2 219 1 2 606 1 3 2640 0 3 2430 0 3 2252 0
3 2140 0 3 2133 0 3 1238 0 3 1631 0 3 2024 0
3 1345 0 3 1136 0 3 845 0 3 422 1 3 162 1
3 84 1 3 100 1 3 2 1 3 47 1 3 242 1
3 456 1 3 268 1 3 318 1 3 32 1 3 467 1
3 47 1 3 390 1 3 183 1 3 105 1 3 115 1
3 164 1 3 93 1 3 120 1 3 80 1 3 677 1
3 64 1 3 168 1 3 74 1 3 16 1 3 157 1
3 625 1 3 48 1 3 273 1 3 63 1 3 76 1
3 113 1 3 363 1

```

```
;
```

```

/*--Get survival plot data from procedure--*/
ods output Survivalplot=SurvivalPlotData;
proc lifetest data=BMT plots=survival(atrisk=0 to 2500 by 500);
  time T * Status(0);
  strata Group / test=logrank adjust=sidak;
run;

```

A portion of the SurvivalPlotData is shown below.

Time	Survival	AtRisk	Event	Censored	tAtRisk	Stratum	StratumNum
0	1.00000	38	0	.	.	1: ALL	1
0	.	38	.	.	0	1: ALL	1
1	0.97368	38	1	.	.	1: ALL	1
55	0.94737	37	1	.	.	1: ALL	1
74	0.92105	36	1	.	.	1: ALL	1
86	0.89474	35	1	.	.	1: ALL	1
104	0.86842	34	1	.	.	1: ALL	1
107	0.84211	33	1	.	.	1: ALL	1
109	0.81579	32	1	.	.	1: ALL	1
110	0.78947	31	1	.	.	1: ALL	1