

Swimmer Plot: Tell a Graphical Story of Your Time to Response Data Using PROC SGPLOT

Stacey D. Phillips, Inventiv Health Clinical, Princeton, NJ

ABSTRACT

ODS Statistical Graphics (SG) procedures are making complex and sophisticated graphics easier to create with every new release of SAS. Through the use of color, line-types, symbols and annotations we can tell a complicated graphical story of our data in one glance. This paper will demonstrate how to create a swimmer plot using PROC SGPLOT that shows multiple pieces of tumor response information for individual subjects in an oncology study. Specifically, the swimmer plot will show total time to tumor response, whether the response was complete or partial, when the response started/ended, censoring information and the current disease stage of the subject. The paper will demonstrate the step-by-step process of creating a swimmer plot from basic PROC SGPLOT statements through more complex annotations using SAS version 9.3. There will also be some discussion on generating similar plots using SAS version 9.2 as well as Graph Template Language (GTL).

INTRODUCTION

Investigators in oncology studies are frequently interested in the effects of a study drug on patients' tumor size and composition. In particular they are often concerned with whether or not a subject had a response to the study drug and the timing of that response in relation to the administration of the study drug. Often this "time to response" analysis is done in an aggregate manner using methods such as Kaplan-Meier analysis though sometimes the investigators prefer to dig deeper and look at an individual subject's pattern of response. A swimmer plot is a graphical way of showing multiple pieces of a subject's response "story" in one glance. Through the use of ODS Statistical Graphics procedures this paper will demonstrate step-by-step how to generate a swimmer plot that can be used to tell the story of tumor response for subjects in a clinical trial.

This paper is appropriate for intermediate SAS programmers and assumes some knowledge of ODS Graphics though a brief overview of ODS Graphics is included for those new to the concepts. The paper will start with a brief description and example of a swimmer plot and then will proceed to the step-by-step process of generating the plot using the SGPLOT procedure. The examples in the paper use SAS version 9.3 though there will be some discussion at the end regarding alternate ways to generate the plots using an older version of SAS and Graph Template Language.

BRIEF OVERVIEW OF ODS GRAPHICS

With the advent of the SAS ODS Graphics in SAS version 9.2 it is possible to create highly complex and customized graphs more easily than ever before. There are three main ways that graphs are produced in ODS Graphics:

AUTOMATICALLY GENERATE GRAPHS FROM SAS PROCEDURES

Certain procedures allow graphs to be automatically generated directly from statistical procedures. For example, a basic Kaplan-Meier plot is automatically generated from the LIFETEST procedure by adding simple ODS statements and a PLOTS= statement. More than 70 Base SAS and other SAS applications' analytic procedures can produce ODS Graphics output. These graphs can be edited in ODS Graphics Editor and require no graph programming knowledge.

CREATE GRAPHS USING STATISTICAL GRAPHICS PROCEDURES

The second way to create complex and customized graphics using very few lines of code is to use the Statistical Graphics (SG) procedures available starting in version 9.2 and improving significantly in version 9.3. The three main SG procedures are listed below:

1. SGPLOT procedure – generates many types of single-cell graphs and is the focus of this paper
2. SGPANEL procedure – generates graphs that are classified into multiple cells
3. SGSCATTER procedure – generates scatter plot panels and matrices

Examples using PROC SGPLOT are the focus of this paper as it is easy to use and powerful in its options for customizability. The key to creating sophisticated graphs with PROC SGPLOT is the overlaying of multiple plots on

top of each other. While not all plot types generated from PROC SGPLOT can be overlaid together many of them can be. A brief summary of which plot types can be combined is below:

1. Basic plots (eg. scatter, series, step, needle, hbarparm)
2. Fit and confidence plots (eg. ellipse, regression, loess)
3. Distribution plots (eg. histogram, density, vbox, hbox)
4. Categorization plots (eg. vbar, hbar, vline, hline)

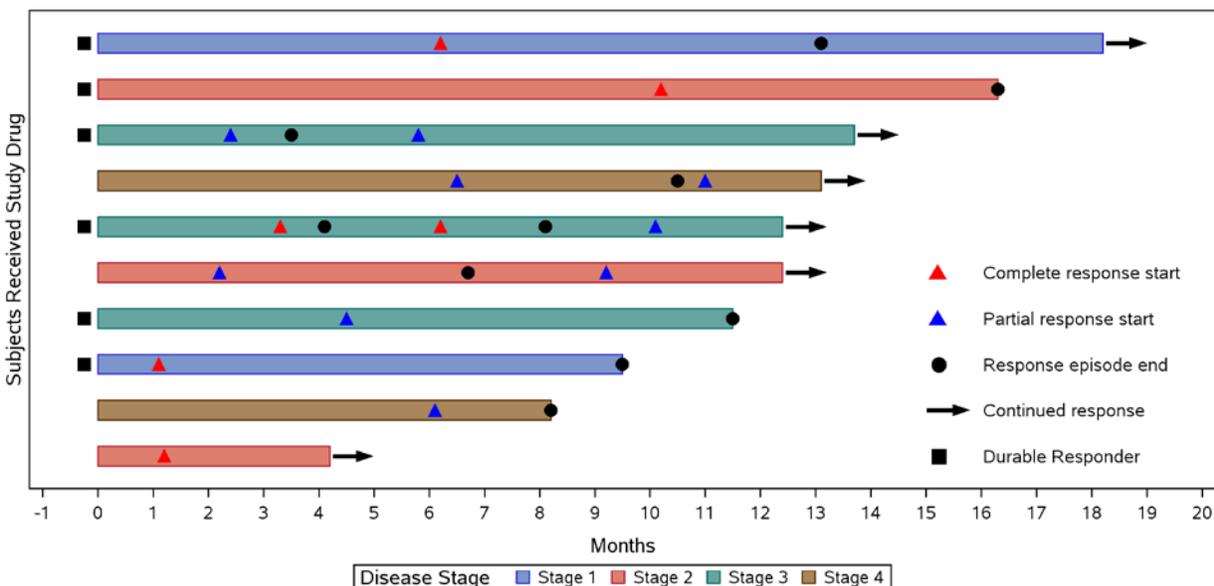
The first two plot types (basic and fit/confidence) can be easily combined with each other or used as a stand-alone graph. Distribution and categorization plots can be combined in limited ways within their plots types. The focus of this paper is on the use of SCATTER and HBARPARM statements to generate the example swimmer plot. Further, SAS version 9.3 affords the opportunity to use the SG Annotation to further customize the plot by adding a detailed legend to the plot area.

CREATE GRAPHS USING GRAPH TEMPLATE LANGUAGE (GTL)

The third way to create graphs using the ODS system is through the use of Graph Template Language or GTL. This is an advanced topic that uses a combination of the TEMPLATE and SGRENDER procedures to create highly complex and customizable graphs. A complete discussion of GTL is out of the scope of this paper though it is mentioned later with regards to programming swimmer plots in SAS version 9.2. There is a fairly steep learning curve for GTL but fortunately some of the SG Procedures assist in generating the underlying GTL code through the use of the TEMPLOUT= statement on the PROC SGxxx statement.

EXAMPLE AND DESCRIPTION OF SWIMMER PLOT

A swimmer plot is a graphical tool involving horizontal bars that can be used to show multiple pieces of information about a given data set in one glance. In this example a swimmer plot is used to tell “a story” about the effects of a study treatment on tumor response for individual subjects in an oncology study. The example plot below shows the following six pieces of information: length of treatment duration, when each response episode(s) started (time to response), when the response episode(s) ended or were ongoing, whether the response was complete or partial, whether or not the subject was considered a durable responder and the disease stage at baseline. Through the use of a swimmer plot we are able to look at our data on an individual subject level rather than an aggregate level that is often done for “time to response” analysis using Kaplan-Meier methods.



Each bar represents one subject in the study.
A Durable Responder is a subject who has confirmed response for at least 183 days (6 months).

Figure 1. Time to Response (Customized Plot with Annotations)

STEP-BY-STEP PROGRESSION OF SWIMMER PLOT WITH CODE AND PLOT EXAMPLES

Creating a swimmer plot with multiple pieces of information related to tumor response is actually quite simple using PROC SGPLOT and a combination of HBARPARM and SCATTER statements. In the below sections there are details about the input data set followed by a description of how to break the construction of the plot into three phases: basic, customized and customized with annotation and includes explanations of the statements used in each step.

EXPLANATION OF INPUT DATA

The data set used for the swimmer plot is below. It shows 10 subjects with the following variables:

- CRSTART1/CRSTART2: complete response start time in months for first and second response episodes
- PRSTART1/PRSTART2: partial response start time in months for first and second response episodes
- END1/END2: end time of response episode(s) (either complete or partial) in months
- LINE_END: end of treatment time for subject in months
- SYMBOL: “a” for arrow and “c” for circle (arrow indicates ongoing response, circle indicates response episode end)
- DSTAGE: disease stage at baseline
- ADRFL: durable responder flag

subjid	crstart1	crstart2	prstart1	prstart2	end1	end2	line_end	symbol	dstage	adrfl
1	3.3	6.2	10.1	.	4.1	8.1	12.4	a	3	Y
2	.	.	6.1	.	8.2	.	8.2	c	4	
3	1.2	4.2	a	2	
4	6.2	.	.	.	13.1	.	18.2	a	1	Y
5	.	.	4.5	.	11.5	.	11.5	c	3	Y
6	.	.	2.2	9.2	6.7	.	12.4	a	2	
7	10.2	.	.	.	16.3	.	16.3	c	2	Y
8	.	.	2.4	5.8	3.5	.	13.7	a	3	Y
9	1.1	.	.	.	9.5	.	9.5	c	1	Y
10	.	.	6.5	11	10.5	.	13.1	a	4	

Display 1. Input data for swimmer plot

BASIC PLOT

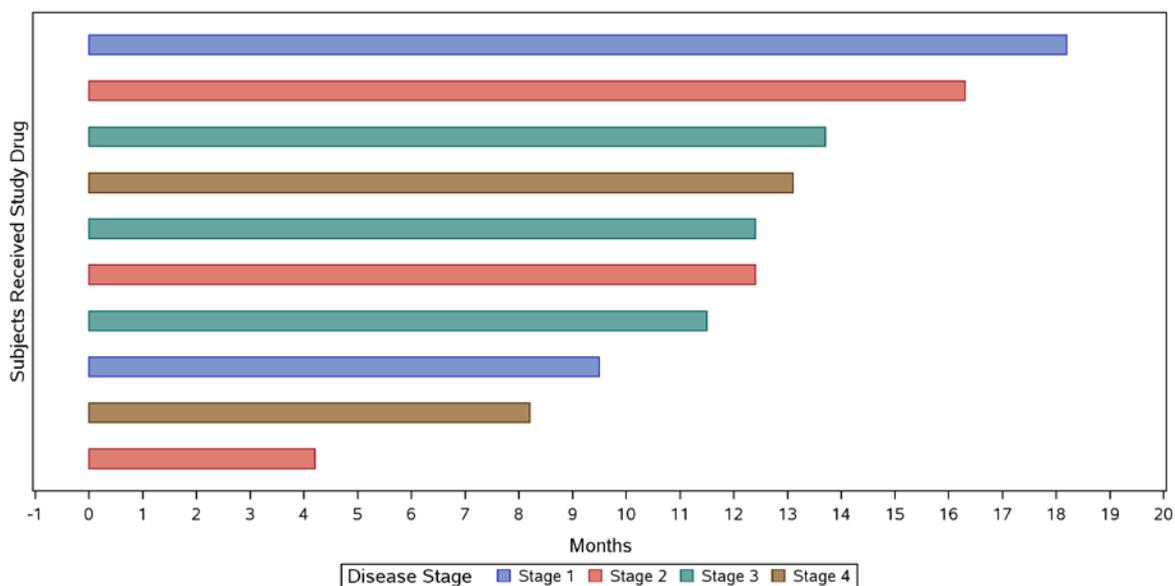


Figure 2. Time to Response (Basic Plot)

The first and most basic version of this plot demonstrates three things about the time to response data:

1. Total treatment time (represented by bar length).
2. Number of subjects who received study drug (represented by number of bars).
3. Disease stage at baseline (represented by the color of the bars).

The basic plot is simple to generate as it relies on only three PROC SGPLOT statements: HBARPARAM, YXAXIS and XAXIS. A short description of each statement and corresponding arguments are below.

```
proc sgplot data=ttr;  
  hbarparm category=subjn response=line_end/group=dstage barwidth=.5 ;  
  yaxis type=discrete display=(novalues noticks) label="Subjects Received Drug";  
  xaxis type=linear label="Months" values=(-1 to 20 by 1);  
run;
```

Statement 1: HBARPARAM

Creates a horizontal bar chart based on a pre-summarized response value for each unique value of the category variable. One difference between HBARPARAM and HBAR is that the response variable is required for HBARPARAM. The response variable should contain pre-summarized computed values such as a sum or a mean or numeric value that doesn't need further summarization. Additional arguments on the HBARPARAM in the swimmer plot example:

- **CATEGORY** (required): specifies the variable that categorizes the data. In this case the category variable is the subject number (SUBJN). This value is always treated as discrete even if it's a numeric value and should always be unique.
- **RESPONSE** (required): specifies the numeric response variable that is always pre-summarized (eg. sum, mean, etc.) In the swimmer plot example this variable is LINE_END that is the numeric value marking the end of the bar.
- **GROUP** (optional): specifies the variable that is used to group the data. This example utilizes the DSTAGE variable which distinguishes the different disease stages by showing different colors within each bar as it corresponds to each subject.
- **BARWIDTH** (optional): specifies the width of the bars on the plot using a numeric value ranging from 0 (narrow) to 1 (wide). Usually the programmer needs to experiment with this variable a bit because it depends on the number of bars in your graph and individual preference for spacing between the bars.

Statement 2: YAXIS

The YAXIS statement is used in the swimmer plot code to specify the details of range, axis type, axis label and other details of the y-axis:

- **TYPE** (optional): specifies one of four axis types (discrete, linear, log, time). The example considers the subject numbers (SUBJN) to be discrete to be consistent with the HBARPARAM category variable.
- **DISPLAY** (optional): specifies which features are displayed on the axis. You can specify ALL or NONE or individual pieces such as labels, lines, ticks and values. For the swimmer plot we don't want to see the values or the tickmarks so turn them off with NOVALUE and NOTICK options.
- **LABEL** (optional): specifies the y-axis label enclosed in double quotes

Statement 3: XAXIS

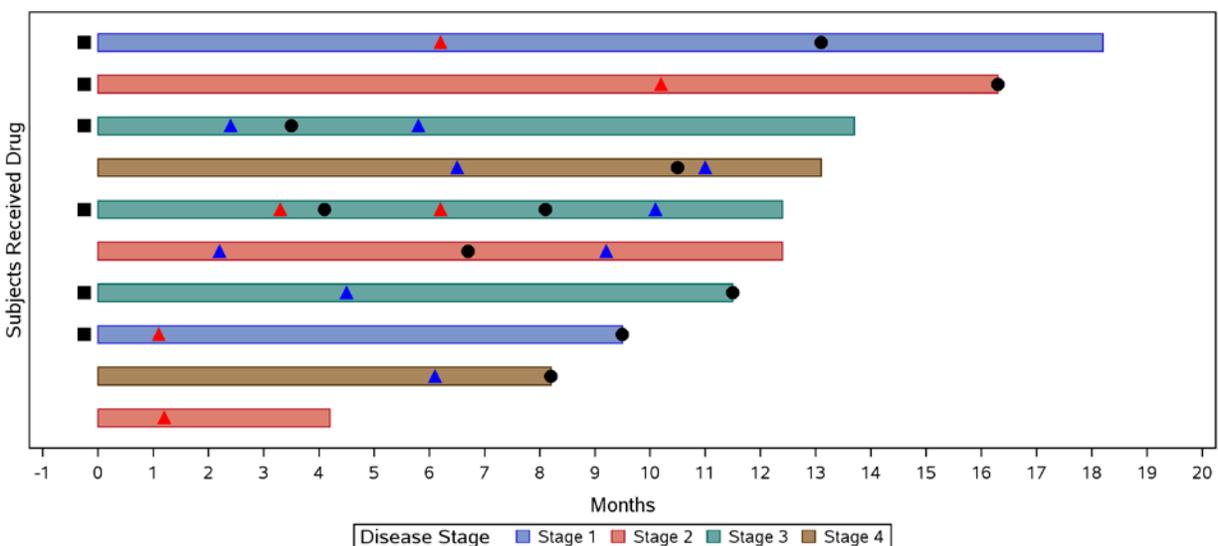
Similar to the YAXIS statement the swimmer plot code requires an XAXIS statement as well. The arguments for the XAXIS statement are nearly the same as the YAXIS but include one additional argument:

- **VALUES**: specifies the values of the tick marks on the axis. There are many ways to specify values for the axis and in this example we've chosen *value-1 TO value-2 BY increment-value* which specifies the range from -1 to 20 by increments of 1.

CUSTOMIZED PLOT

The basic plot is a good place to start but is clearly lacking the main point of interest in the swimmer plot: time to response information. This information is shown by marking on each bar where the response episode(s) started and ended and what type of response the subject achieved (complete or partial). These customizations will add another three pieces of information to the swimmer plot:

1. When the response started or ended (represented by the position of the triangles and black dots along the bar)
2. Whether the response was partial or complete (represented by the color of the triangles)
3. Whether or not the subject was a durable responder (represented by a black square before each bar)



Each bar represents one subject in the study.
 The red and blue triangles represent the start of the each response episode for complete response and partial response respectively.
 A black circle represents the end of a response episode.
 A black square indicates the subject is a durable responder (subject who has confirmed response for at least 183 days (6 months)).

Figure 3. Time to Response (Customized Plot)

PROC SGPLOT makes it simple to add this extra detail through the use of a series of overlaid SCATTER statements that put the symbols directly onto the bars.

```
proc sgplot data=final1 ;
  hbarparm category=subjn response=line_end/group=dstage barwidth=.5 ;
  yaxis type=discrete display=(novalues noticks) label="Subjects Received tudy Drug";
  xaxis type=linear label="Months" values=(-1 to 20 by 1);
  scatter X=durresp Y=subjn /markerattrs=(symbol=squarefilled size=9 color=black);
  scatter X=crstart1 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=red);
  scatter X=crstart2 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=red);
  scatter X=prstart1 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=blue);
  scatter X=prstart2 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=blue);
  scatter X=end1 Y=subjn /markerattrs=(symbol=circlefilled size=9 color=black);
  scatter X=end2 Y=subjn /markerattrs=(symbol=circlefilled size=9 color=black);
run;
```

Statement 4: SCATTER

- X (required): specifies the variable containing the value for the location of the scatter plot markers on the x-axis
- Y (required): specifies the variable containing the value for the location of the scatter plot markers on the y-axis
- MARKERATTRS (optional): specifies the appearance of the markers on the plot. There are several ways to specify this style element and in this example the SYMBOL, SIZE and COLOR arguments are used explicitly

CUSTOMIZED PLOT WITH ANNOTATIONS

While the customized plot is definitely an improvement over the basic plot, it's pretty inconvenient to have the symbol definitions in the footnotes. Usually the preference would be to have this information in a legend of some sort but this would not be possible using just the basic PROC SGPLOT statements. The default legend that appears is from the GROUP=DSTAGE option on the HBARPARM statement showing the disease stage by color for grouped data. To create a customized legend for this plot we need to use SG Annotation that was added to ODS Statistical Graphics for SAS version 9.3. Annotation can appear complicated at first but it's fairly similar to the Annotate Facility from SAS/Graph and even without prior annotation experience it is not too difficult to learn.

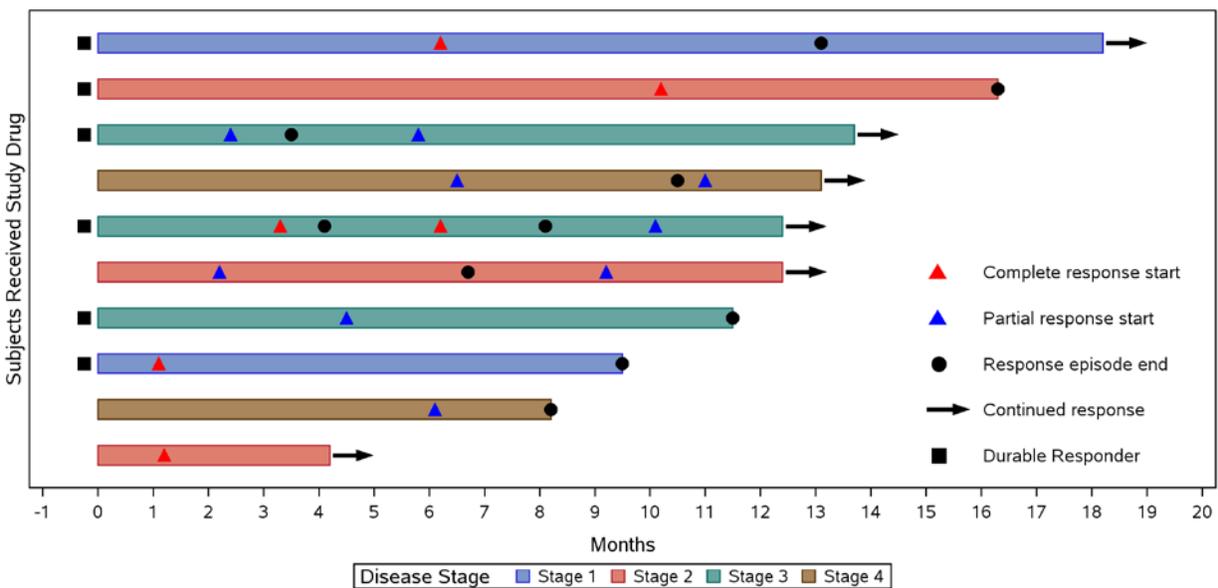


Figure 4. Time to Response (Customized Plot with Annotations)

The final customized plot with annotations has almost exactly the same code as the previous plot with the addition of the SGANNO=ANNO text added to the PROC SGPLOT statement. To use the SGANNO=ANNO option a data set named ANNO needs to be created to produce the custom legend.

```
proc sgplot data=final1 sganno=anno ;
  hbarparm category=subjn response=line_end/group=dstage attrid=A barwidth=.5 ;
  scatter X=durresp Y=subjn /markerattrs=(symbol=squarefilled size=9 color=black);
  scatter X=crstart1 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=red);
  scatter X=crstart2 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=red);
  scatter X=prstart1 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=blue);
  scatter X=prstart2 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=blue);
  scatter X=end1 Y=subjn /markerattrs=(symbol=circlefilled size=9 color=black);
  scatter X=end2 Y=subjn /markerattrs=(symbol=circlefilled size=9 color=black);
  yaxis type=discrete display=(novalues noticks) label="Subjects Received Drug";
  xaxis type=linear label="Months" values=(-1 to 20 by 1);
run;
```

SGANNO=ANNO

The final version of the swimmer plot needs to include annotation that creates both the arrows at the end of the bars indicating ongoing response and annotation for the legend within the plot that shows what each marker character symbolizes. Annotation can be somewhat intimidating at first but it's really just creating a SAS data set with certain variables and functions that tell ODS where to place text, lines, symbols, etc. Each observation within an annotate data set represents a command to draw an annotation element using reserved keywords. A complete description of all of the features within SG Annotation is outside the scope of this paper and can be found in the online SAS documentation. Below is some discussion regarding the annotation elements used in the swimmer plot.

```
/*CREATE ARROWS AT THE END OF THE BARS TO INDICATE CONTINUING RESPONSE*/
data anno1;
  length function $25;
  retain drawspace "datavalue" linecolor "black" linethickness 3;
  set finall(keep=subjid subjn line_end symbol where=(symbol='a'));
  function='ARROW';
  x1=line_end+.05;
  x2=line_end+.8;
  shape='FILLED';
  y1=subjn;
  y2=subjn;
run;
```

function	linecolor	drawspace	linethicknes	subjid	line_end	symbol	subjn	x1	x2	y1	y2	shape
ARROW	black	datavalue	3	4	18.2 a		1	18.25	19	1	1	1 FILLED
ARROW	black	datavalue	3	8	13.7 a		3	13.75	14.5	3	3	3 FILLED
ARROW	black	datavalue	3	10	13.1 a		4	13.15	13.9	4	4	4 FILLED
ARROW	black	datavalue	3	1	12.4 a		5	12.45	13.2	5	5	5 FILLED
ARROW	black	datavalue	3	6	12.4 a		6	12.45	13.2	6	6	6 FILLED
ARROW	black	datavalue	3	3	4.2 a		10	4.25	5	10	10	10 FILLED

Display 2. Annotate data set #1 (ANNO1) to add arrows to end of bars

One of the key components of creating an annotate data set is setting the “drawing space.” For PROC SGPLOT there are several options to choose from and in the swimmer plot example we chose the DATAVALUE option which indicates the positions and scaling of the annotations and will be based on the data values within the graph (rather than an alternate scale based on percentage of plot area or pixels, etc.). Since all observations in the data set will use the same DRAWSPACE, LINECOLOR and LINETHICKNESS values these variables are assigned in a RETAIN statement.

There are several types of elements that can be added to a plot using SG Annotation. It’s common to add shapes such as circles and rectangles though in this first case we’re interested in creating an arrow. Luckily there is an ARROW function within SG Annotation so that we don’t have to draw the arrows manually using annotation. The ARROW function has four required arguments: x1, x2, y1, y2. These variables specify the coordinates of the beginning of the arrow (x1, y1) and coordinates of the end of the arrow (x2, y2). The original swimmer plot data set is subset to keep only the subjects requiring an arrow (using the WHERE statement) and by using the X1 argument set the start of the arrow .05 past the end of the bar whose value is specified in the LINE_END variable for each Y1 value of SUBJN. Similarly, the end of the arrow is set with the X2 argument at .8 past the end of the bar for each Y2 value of SUBJN and uses the FILLED style arrowhead.

The SG annotation code to create the legend is similar to creating the arrows above. Due to space limitations the discussion will only be for the first row of the legend (the red triangle indicating complete response start time). The complete code for the customized swimmer plot with annotation can be found at the end of the paper after the conclusion.

```
/*CREATE FIRST ROW OF THE LEGEND TO SHOW RED TRIANGLE INDICATING COMPLETE RESPONSE*/
data anno2;
  length label $100 function fillcolor linecolor $25;
  retain drawspace 'datavalue' linethickness 3 display 'ALL'
  widthunit 'data' heightunit 'data' anchor 'left';

  /*Draw red triangle indicating complete response start time*/
  function='POLYGON'; x1=15.20; y1=4.10; linecolor='red'; fillcolor='red'; output;
  function='POLYCONT'; x1=15.30; y1=3.90; linecolor='red'; output;
  function='POLYCONT'; x1=15.10; y1=3.90; linecolor='red'; output;
  function='POLYCONT'; x1=15.20; y1=4.10; linecolor='red'; output;

  /*Create text labels for the legend*/
  function='TEXT'; x1=16; y1=6; label="Complete response start"; width=7; shape='';
  fillcolor=''; output;
run;
```

label	function	fillcolor	linecolor	drawspace	linethickness	display	widthunit	heightunit	anchor	x1	y1	height	width	x2	y2	shape
	POLYGON	red	red	datavalue	3	ALL	data	data	left	15.2	4.1
	POLYCONT	red	red	datavalue	3	ALL	data	data	left	15.3	3.9
	POLYCONT	red	red	datavalue	3	ALL	data	data	left	15.1	3.9
	POLYCONT	red	red	datavalue	3	ALL	data	data	left	15.2	4.1
Complete response start	TEXT		black	datavalue	3	ALL	data	data	left	16	6	0.25	7	15.8	9	

Display 3. Annotate data set #2 (ANNO2) to create first row of customized legend for “Complete Response Start”

Similar to the annotation of the arrows at the end of the bars, the values for the functions that will remain constant are set in a RETAIN statement which saves having to retype these functions and values for each observation. Because there is no TRIANGLE function within SG Annotation the triangles have to be drawn using the POLYGON and POLYCONT functions. The X1 and Y1 parameters are the coordinates of the points that the lines of the polygon will be drawn between and LINECOLOR/FILLCOLOR='red' specifies the color of the lines and fill of the triangles.

The text labels for the legend are created using the TEXT function. The X1 and Y1 coordinates specify the text location which combines with the ANCHOR function in the retain statement to position the text in the plot area. The WIDTH function controls the width of the annotation itself.

Ultimately these two annotate data sets are set together and referenced in the PROC SGPLOT code below:

```
/*Set both annotate data sets together to specify in PROC SGPLOT*/
data anno;
  set anno1 anno2;
run;

proc sgplot data=final1 sganno=anno ;
  hbarparm category=subjn response=line_end/group=dstage attrid=A barwidth=.5 ;
  scatter X=durresp Y=subjn /markerattrs=(symbol=squarefilled size=9 color=black);
  scatter X=crstart1 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=red);
  scatter X=crstart2 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=red);
  scatter X=prstart1 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=blue);
  scatter X=prstart2 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=blue);
  scatter X=end1 Y=subjn /markerattrs=(symbol=circlefilled size=9 color=black);
  scatter X=end2 Y=subjn /markerattrs=(symbol=circlefilled size=9 color=black);
  yaxis type=discrete display=(novalues noticks) label="Subjects Received Drug";
  xaxis type=linear label="Months" values=(-1 to 20 by 1);
run;
```

BRIEF DISCUSSION REGARDING THE USE OF SAS 9.2 AND 9.3

The examples in this paper use SAS version 9.3 but there are ways to produce similar and more basic versions of the swimmer plot using SAS version 9.2. Below are some of the key differences between the two versions.

DIFFERENCE IN PROCEDURES DEPENDING ON VERSION OF SAS

There are other PROC SGPLOT statements that can be used to create graph similar in style to a swimmer plot.

Version 9.2 (VECTOR Statement)

This statement creates a vector plot that draws arrows from a point of origin to each data point. It can be used in conjunction with the SCATTER plot statement and while the effect is more of a “line” plot than a “bar” plot it could easily be used to tell the same story. Since SG Annotation is not available in 9.2 the customized legend would need to be replaced with footnotes explaining the various marker symbols. The HBARPARG statement is not available in SAS version 9.2.

Version 9.3 (VECTOR or HBARPARG Statement)

Similar to version 9.2 a VECTOR statement can also be used in version 9.3. I chose to use HBARPARG because the statistician wanted the “bar” effect rather than the line effect. Also, since not all of the bars on the swimmer plot would require an arrow at the end it made sense to annotate them explicitly where needed.

Version 9.3 (HIGHLOW Statement)

Another interesting option in 9.3 that I didn't have a chance to explore thoroughly is the use of the HIGHLOW statement to generate a swimmer plot. In his excellent blog “Graphically Speaking”, Sanjay Matange includes examples of how to create an Adverse Event timeline using a HIGHLOW plot. The example swimmer plot from this paper is very similar aesthetically to an AE timeline and I have no doubt that the HIGHLOW statement could also be used to generate a graph similar to the example swimmer plot.

ANNOTATION ONLY AVAILABLE IN VERSION 9.3

One very important and key difference between versions 9.2 and 9.3 is the availability of SG Annotation. This was only introduced in version 9.3 so any plot being generated in 9.2 will have reduced ability to customize things such as legends and other text within the plot area. The KEYLEGEND statement is available in 9.2 but has limited usage for many requested customizations.

ASSIGNING GROUPED DATA ATTRIBUTES CORRECTLY

An ongoing challenge when programming figures in SAS has been making sure the correct groups are assigned to the same attributes consistently and easily across plots. The default in SAS is to assign attributes such as colors, line patterns, etc. to each group in the order that they appear in the data. This is often not what the programmer wants and while it seems like assigning color and other attributes should be an easy thing to do it isn't in 9.2. Further, it can be challenging to assign the colors consistently across multiple plots if one of the plots is missing a particular group. Both 9.2 and 9.3 have a solution to this issue though the 9.3 attribute map is much simpler.

Version 9.2 – INDEX= option with Graph Template Language

Graph Template Language (GTL) is an advanced topic for programming graphs and figures and is out of the scope of this paper. It is worth mentioning though that a solution to applying color attributes consistently across graphs does exist for version 9.2 but rather than using the simpler SG procedures the programmer would need to utilize the more complicated GTL method with corresponding INDEX= option.

Version 9.3 – Attribute Maps

Attribute maps are a concept that was added to version 9.3 which greatly simplifies the assignment of style attributes such as colors and patterns within a graph. The use of attribute maps negates the need to use the more complicated GTL and INDEX= option when customizing a graph's attributes. In many ways an attribute map is similar in concept to an annotate data set but is simpler and more straightforward. The addition of attribute maps to version 9.3 has been a significant and much-needed improvement to graphing capabilities in SAS.

CONCLUSION

SAS has provided the user community with powerful tools to generate graphs with the introduction of ODS Graphics. Using ODS Statistical Graphics procedures this paper has shown that telling a complex story about our data is possible using just a few SAS statements. Through the technique of overlaying plots on top of each other a graph that looks complicated is easily created even for users with little SAS graphics experience. The addition of SG Annotation in version 9.3 has improved the power and capability even further allowing the user to further customize their graphs.

COMPLETE CODE FOR CUSTOMIZED SWIMMER PLOT WITH ANNOTATION

INPUT DATA: FINAL1

subjid	crstart1	crstart2	prstart1	prstart2	end1	end2	line_end	symbol	dstage	adrf1
1	3.3	6.2	10.1	.	4.1	8.1	12.4	a		3 Y
2	.	.	6.1	.	8.2	.	8.2	c		4
3	1.2	4.2	a		2
4	6.2	.	.	.	13.1	.	18.2	a		1 Y
5	.	.	4.5	.	11.5	.	11.5	c		3 Y
6	.	.	2.2	9.2	6.7	.	12.4	a		2
7	10.2	.	.	.	16.3	.	16.3	c		2 Y
8	.	.	2.4	5.8	3.5	.	13.7	a		3 Y
9	1.1	.	.	.	9.5	.	9.5	c		1 Y
10	.	.	6.5	11	10.5	.	13.1	a		4

Display 4. Input data for swimmer plot

```
/*Sort by descending length of bars*/
proc sort data=final1;
  by descending line_end;
run;

proc format;
  value stage 1="Stage 1"
             2="Stage 2"
             3="Stage 3"
             4="Stage 4";
```

```

run;

data finall;
  set finall;
  /*Create new subject number variable based on the descending order*/
  subjn=_n_;

  /*Flag the Durable Responder subjects and set DURRESP as the location for symbol*/
  if adrfl='Y' then durrep=-.25;

  label dstage="Disease Stage";
  format dstage stage.;
run;

proc sort data=finall;
  by subjn;
run;

***CREATE ANNOTATE DATA SETS***;

/*Arrows at the end of the bars for continuing response*/

data anno1;
  length function linecolor $25;
  retain drawspace "datavalue" linecolor "black" linethickness 3;
  set finall(keep=subjid subjn line_end symbol where=(symbol='a'));
  function='ARROW';
  x1=line_end+.05;
  x2=line_end+.8;
  y1=subjn;
  y2=subjn;
  shape='FILLED';

run;

/*Customized legend with symbols and text*/
data anno2;
  length label $100 function fillcolor linecolor $25;
  retain drawspace 'datavalue' linethickness 3 display 'ALL'
  widthunit 'data' heightunit 'data' anchor 'left';

  /*Draw red triangle indicating complete response start time*/
  function='POLYGON'; x1=15.20; y1=4.10; linecolor='red'; fillcolor='red'; output;
  function='POLYCONT'; x1=15.30; y1=3.90; linecolor='red'; output;
  function='POLYCONT'; x1=15.10; y1=3.90; linecolor='red'; output;
  function='POLYCONT'; x1=15.20; y1=4.10; linecolor='red'; output;

  /*Draw blue triangle indicating partial response start time*/
  function='POLYGON'; x1=15.20; y1=3.1; linecolor='blue'; fillcolor='blue'; output;
  function='POLYCONT'; x1=15.30; y1=2.9; linecolor='blue'; output;
  function='POLYCONT'; x1=15.10; y1=2.9; linecolor='blue'; output;
  function='POLYCONT'; x1=15.20; y1=3.1; linecolor='blue'; output;

  /*Draw a circle indicating response end time using the OVAL function*/
  function='OVAL'; x1=15.125; y1=8; linecolor='black'; fillcolor='black'; height=.25;
  width=.2; output;

  /*Draw an arrow indicating continuing response*/
  function='ARROW'; x1=15; x2=15.8; y1=9; y2=9; linecolor='black'; shape='FILLED';
  output;

  /*Draw a square indicating durable responder using the RECTANGLE function*/
  function='RECTANGLE'; x1=15.125; y1=10; linecolor='black'; fillcolor='black';
  height=.25; width=.2; output;

  /*Create text labels for the legend*/

```

```

function='TEXT'; x1=16; y1=6; label="Complete response start"; width=7; shape='';
    fillcolor=''; output;
function='TEXT'; x1=16; y1=7; label="Partial response start"; width=7; shape='';
    fillcolor=''; output;
function='TEXT'; x1=16; y1=8; label="Response episode end"; width=7; shape='';
    fillcolor=''; output;
function='TEXT'; x1=16; y1=9; label="Continued response"; width=7; shape='';
    fillcolor=''; output;
function='TEXT'; x1=16; y1=10; label="Durable Responder"; width=7; shape='';
    fillcolor=''; output;
run;

/*Set both annotate data sets together to specify in PROC SGPLOT*/
data anno;
    set annol anno2;
run;

/*Attribute map for assigning colors for disease stage in GROUP= option*/
data attrmap;
    length fillcolor $25;
    ID='A'; value=1; fillcolor='cx7C95CA'; linecolor='cx7C95CA'; output;
    ID='A'; value=2; fillcolor='cxDE7E6F'; linecolor='cxDE7E6F'; output;
    ID='A'; value=3; fillcolor='cx66A5A0'; linecolor='cx66A5A0'; output;
    ID='A'; value=4; fillcolor='cxA9865B'; linecolor='cxA9865B'; output;
run;

proc sgplot data=final1 sganno=anno ;
    hbarparm category=subjn response=line_end/group=dstage attrid=A barwidth=.5 ;
    scatter X=durresp Y=subjn /markerattrs=(symbol=squarefilled size=9 color=black);
    scatter X=crstart1 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=red);
    scatter X=crstart2 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=red);
    scatter X=prstart1 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=blue);
    scatter X=prstart2 Y=subjn /markerattrs=(symbol=trianglefilled size=9 color=blue);
    scatter X=end1 Y=subjn /markerattrs=(symbol=circlefilled size=9 color=black);
    scatter X=end2 Y=subjn /markerattrs=(symbol=circlefilled size=9 color=black);
    yaxis type=discrete display=(novalues noticks) label="Subjects Received Drug";
    xaxis type=linear label="Months" values=(-1 to 20 by 1);
run;

```

REFERENCES

- Matange, Sanjay (2012) *"Clinical Graphs Using SG Procedures Course Notes" Proceedings of the Pharmsug 2012 Conference*. Cary, NC: SAS Institute.
- Matange, Sanjay. *"AE Timeline by Name"*. Cary, NC: SAS Institute.
<http://blogs.sas.com/content/graphicallyspeaking/2013/01/30/ae-timeline-by-name/>

RECOMMENDED READING

- Matange, Sanjay. *"Graphically Speaking: Data Visualization with a Focus on ODS Graphics"*. Cary, NC: SAS Institute. <http://blogs.sas.com/content/graphicallyspeaking/>
- SAS® 9.3 ODS Graphics Procedures Guide
- SAS® 9.3 Graph Template Language User's Guide
- SAS® 9.3 Graph Template Language Reference
- SAS/STAT® 9.3 User's Guide

CONTACT INFORMATION

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

Name: Stacey D. Phillips
Enterprise: Inventiv Health Clinical
Address: 3071 Cimarron Trl.
City, State ZIP: Madison, WI 53719
Work Phone: (608) 298-7008
Fax: (608) 298-7008
E-mail: stacey.phillips@inventivhealth.com
Web: www.inventivhealthclinical.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.