PharmaSUG China 2024 - Paper PO- 10020

Graph Programming Tips for Swimmer Plot

Xin Jiang, Qilu Pharmaceutical Co., Ltd
Wenyi Yu, Qilu Pharmaceutical Co., Ltd

ABSTRACT

Frequently, graphs of the data are required by analysts prior to beginning an analysis. The results of a complex analysis might require data analysis using multiple procedures and some custom data management. Although many of these tasks can be accomplished using the SG (Statistical Graphics) Procedures, those procedures do not provide many of the advanced layout capabilities of GTL (Graph Template Language). In clinical trials industry, the “swimmer plot” displays the response of the tumor to the study drug over time. This paper will share some tips about how to custom graphs using the **TEMPLATE** procedures.

INTRODUCTION

A “Swimmer Plot” is a graphical way of displaying several aspects of a subject’s tumor response such as total time to tumor response. Indicators for the status of each subject are also included, along with details on their treatment group, best overall response, and other information required to be displayed. The **AXISTABLE** statement when combined with the **LAYOUT LATTICE** statement places text values along the Y axis at specific values on the axis, resulting in easier layout and better typography. Due to the limitation of symbol types, statement **SYMBOLCHAR/SYMBOLIMAGE** offers more flexibility to present subject’s status as well as other indicators.

EXAMPLE TABLES

The data used in all the examples in this paper consists of 12 columns: subjid, trtp, trtpn, bor, rsstresc, rsdy, rsdy1, eotstt, ontrtdy, eotdy, dthfl and dthdy. Character columns are shown by red while numeric columns are indicated by blue.

Table 1. Tumor Response Data for Swimmer Plot

subjid	trtp	trtpn	bor	rsstresc	rsdy	rsdy1	eotstt	ontrtdy	eotdy	dthfl	dthdy
001	1mg	1	SD	SD	27	27	EOT		28		
002	5mg	2	SD	SD	26	26	EOT		29	Y	34
003	10mg	3	SD	SD	26	26	ONGOING	27			
004	10mg	3	PD	PD	29	29	EOT		31	Y	36
005	10mg	3	SD	SD	28	28	EOT			Y	
005	10mg	3	SD	PD	40	12	EOT		39	Y	44
006	20mg	4	SD	SD	29	29	ONGOING	30			
007	20mg	4	SD	SD	26	26	EOT			Y	
007	20mg	4	SD	PD	54	28	EOT		54	Y	59
008	20mg	4	PR	PR	7	7	EOT			Y	
008	20mg	4	PR	SD	26	19	EOT			Y	
008	20mg	4	PR	PD	40	14	EOT		44	Y	49
009	40mg	5	CR	CR	40	40	EOT				
009	40mg	5	CR	SD	68	28	EOT				
009	40mg	5	CR	PD	96	28	EOT		98		
010	40mg	5	CR	CR	29	29	ONGOING				
010	40mg	5	CR	SD	36	7	ONGOING	37			

CUSTOMIZE SYMBOLS BY SYMBOLCHAR/SYMBOLIMAGE

The symbol markers supported are limited to the ones listed in the table below. There is only one direction saved for the arrow type symbol - “down”; nevertheless, because swimmer plots are always horizontally laid out, the righthead arrow would be more appropriate for indicating that one is still in treatment. As a result, the existing symbol types cannot match the requirements of drawing.

Table 2. Supported Marker Symbols

Symbol Name	Plot Symbol	Symbol Name	Plot Symbol	Symbol Name	Plot Symbol	Symbol Name	Plot Symbol
ArrowDown		StarFilled		HomeDown		TriangleLeftFilled	
Asterisk		Tack		HomeDownFilled		TriangleRight	
Circle		Tilde		IBeam		TriangleRightFilled	
CircleFilled		Triangle		LessThan		Union	
Diamond		TriangleFilled		Plus		X	
DiamondFilled		TriangleDown		Square		Y	
GreaterThan		TriangleDownFilled		SquareFilled		Z	
Hash		TriangleLeft		Star			

In this situation, using the **SYMBOLCHAR** and **SYMBOLIMAGE** statements can help to custom symbol markers. The **SYMBOLCHAR** and **SYMBOLIMAGE** statement define markers from Unicode characters and images separately. A complete listing of the Unicode hexadecimal constants can be found at the following URL: http://www.unicode.org/charts/charindex.html. The image must be stored in the local file system and in one of the supported formats GIF, JPG, or PNG. When plotting, the symbols can be accessed by their specified names using **SYMBOL**= suboption of the **MARKERATTRS**= option. For instance, the symbol setup code for Figure 1 is as follows:

```
symbolchar name=arrowdiy char='21E8'x / scale=2.2 voffset=0.3;

symbolimage name=alert image="your address\alert-16.png";
```

The **VOFFSET**=/ **HOFFSET**= option in the **SYMBOLCHAR** statement can be used to align the position and axis of the marker. The markers are clipped to the original marker size after rotation, scaling, and offsets are applied. To eliminate the clipping, the **SCALE**= option in the **SYMBOLCHAR** statement as well as the **SIZE**= suboptions of the **MARKERATTRS**= option in the plot statement can be employed to adjust the size and scaling.

Figure 1 displayed below is the inside part of swimmer plot drawn from sample data. The tumor response of each subject is illustated by stacked **barchart** over time with distinct bar colors designated by **DISCRETEATTRMAP** to signify various response types.

There are three different types of markers on the plot: the green rightward arrow produced by **SYMBOLCHAR** means that treatment is still ongoing at the day of last response; the "X" flag,

which is supported by the original symbol type, denotes the day when the individual ends treatment; and the purple alert formed by **SYMBOLIMAGE** indicates the day of death.

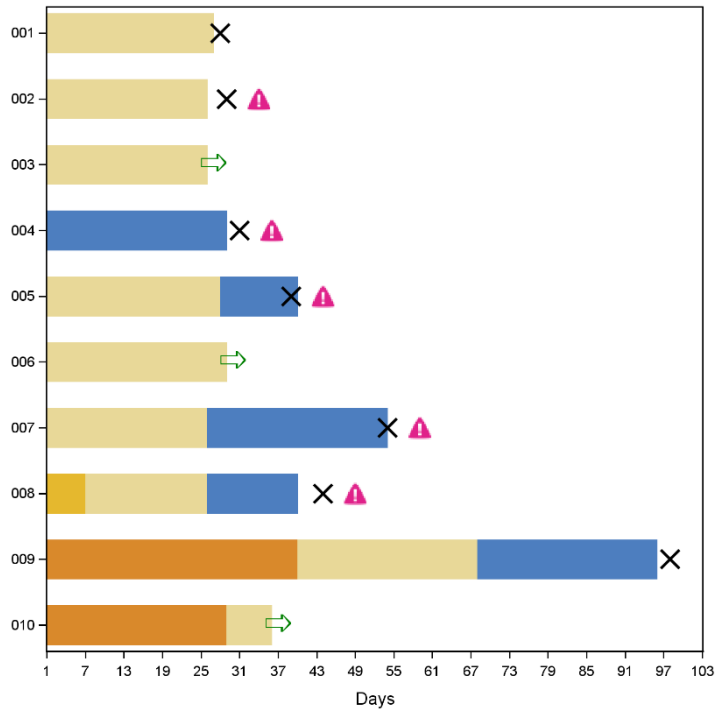

Figure 1. Inside Part of Swimmer Plot

COMPATIBILITY PROBLEM WITH OUTPUT FORMAT

Markers formed by the **SYMBOLCHAR** statement are unaffected by the **WEIGHT**= suboption of the **MARKERATTRS**= option. This issue can be resolved by using the **WEIGHT**= suboption of the **TEXTATTRS** = option, but it not works with PDF format, which has been proved and should be noted. Hence, keep in mind to check whether the final output format supports makers in bold before exporting the graph.

DISPLAY INFORAMTION BY AXISTABLE

Although the inside block has mapped the main swimmer lane and indicators, further information that is necessary for display still needs to be provided. One method is to paste all the data to the y axis tick values, making sure the titles correspond to each column. Nevertheless, the space between each column, which can be controlled by blanks, should be modified several times. If the value is altered, the entire procedure may require to be repeated. Therefore, using the **AXISTABLE** statement in conjunction with the **LAYOUT** statement can provide a more reasonable way to deal with it.

The **AXISTABLE** statement guarantees that all column headers align on the same line and matches the values of each column to axes individually even though they are repeated in data only if they remain consistent. Here is also a small trick: the values of Y axis can be hidden and move to the second column but still match the original axis making the analysis sequence more logical.

As can be seen in the following code, using **LAYOUT LATTICE** statement, the whole canvas is divided into 4 columns with corresponding percentages, which are demonstrated in Figure 2 by blocks of different colors. The **LAYOUT OVERLAY** statement then specifies detailed settings for each column including title, font and size.

```
LAYOUT LATTICE/ rows=1 columns=4 columnweights=(.13 .1 .06 .71)  rowweights=preferred
                rowdatarange=unionall columndatarange=unionall;
    Layout Overlay / yaxisopts=(display=none reverse=true) xaxisopts=(display=none)
                     walldisplay=none;
        AxisTable Value=TRTP Y=SUBJID/ title="Treatment" display=(values)
                titleattrs=(Family="Arial" SIZE=10 weight=bold)
                valueattrs=(SIZE=8 Family="Arial");
    ENDLAYOUT;
    Layout Overlay / yaxisopts=(display=none reverse=true) xaxisopts=(display=none)
                     walldisplay=none;
        AxisTable Value=SUBJID Y=SUBJID/ title="Subject" display=(values)
                titleattrs=(Family="Arial" SIZE=10 weight=bold)
                valueattrs=(SIZE=8 Family="Arial");
    ENDLAYOUT;
    Layout Overlay / yaxisopts=(display=none reverse=true) xaxisopts=(display=none)
                     walldisplay=none;
        AxisTable Value=BOR Y=SUBJID/ title="BOR" display=(values)
                titleattrs=(Family="Arial" SIZE=10 weight=bold)
                valueattrs=(Family="Arial" SIZE=8);
    ENDLAYOUT;

    . . .

ENDLAYOUT;
```

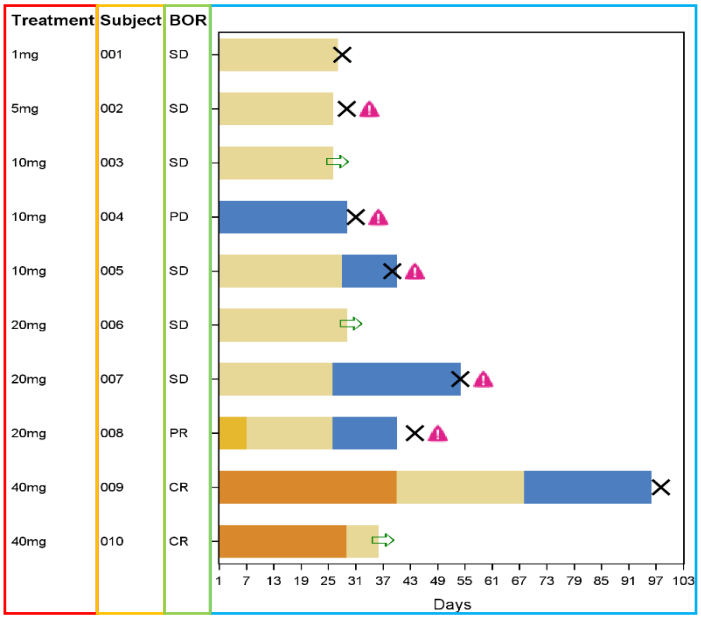

Figure 2. Swimmer Plot with Outside Layout

LEGEND LAYOUT

In the same way as the above idea of splitting the layout, the legend can also be separated into rows and columns. In this example, the legend block is divided into 2 rows. One is for the markers of response bar, which are directly generated by setting "discretelegendentrypolicy=attrmap" of **DISCRETEATTRMAP** and "type=marker" of **DISCRETELEGEND**, so that the order of the response level can be arranged without interference from the order in which the data appears.

The other is for the status of subject created by **LEGENDITEM**, which allows the size of each marker to be adjusted for various scales and appear at the same size in the legend. The final output of swimmer plot for the given data has been displayed in Figure 3.

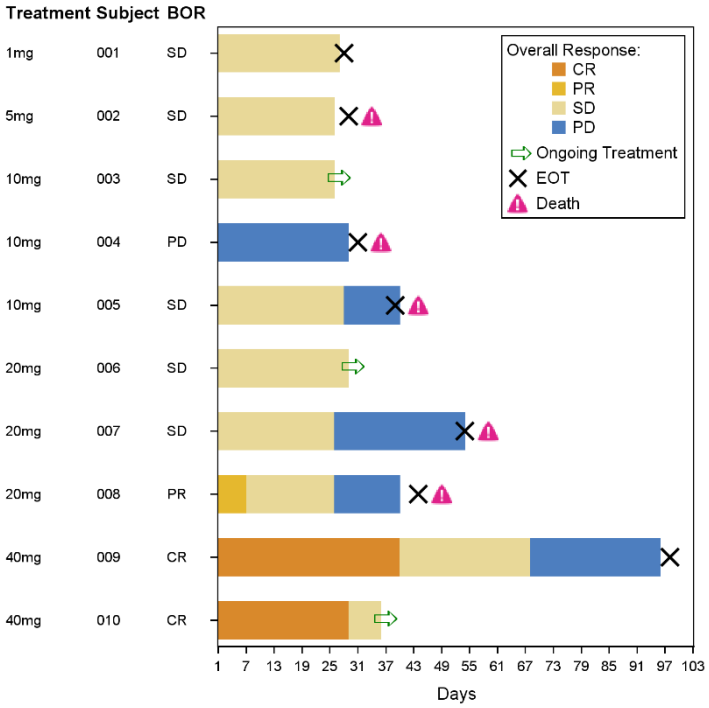

Figure 3. Swimmer Plot with Outside Layout

CONCLUSION

The GTL is a valuable tool for data visualization, offering users a flexible foundation to create highly customizable and complex graphs for data analysis and presentation by defining templates that control various aspects of the graph's appearance and behavior. This paper illustrates tricks and techniques specifically for clinical swimmer plot, using dummy example data, validated code and output graphs. Hopefully, these tips will be useful in practical applications and can be extended to various clinical graphs, such as waterfall and spider plots, among others.