PharmaSUG 2021 - Paper AP-174

Overlay Graphs: A Powerful Tool to Visualize and Simplify Complex Data

Manasa Gangula, Cytel, Inc.

ABSTRACT

SAS® provides an extensive set of graphs for different needs and overlaying these graphs can be powerful in visualizing complex data. By overlaying graphs, we can display the relationship among multiple parameters in a single snapshot so the reviewer will get an explicit idea about the data without any further processing. In this paper, we will be looking at different ways to create an overlay of a vertical bar chart with scatter plot and bubble plot using sample data. We will be using Graph template language (GTL) and SAS/GRAPH procedures to achieve the result.

INTRODUCTION

Pictorial or graphical representation of data is always interesting and easy to understand. As clinical SAS® programmers, we present complex clinical data in the form of TLFs (Tables, Listings and Figures). Figures play an important role in visualizing data related to critical endpoints of a therapy. This paper explains how to visualize sample pain data for two treatment groups using GTL (Graph Template Language) and SAS® procedures like PROC SGPANEL, PROC SGRENDER and PROC SGPLOT. We will compare two different overlays and pick the best graph that helps comprehend the data:

1. Bubble plot over Vertical Bar Chart
2. Scatter plot over Vertical Bar Chart

PERFORMING DATA ANALYSIS

The first step in the process is to obtain all the analysis variables required to present in the graphs. Using summary and frequency procedures we can obtain the required statistics.

The following code gives us the required statistics:

PROC SUMMARY is used to get the mean and standard error for aval value and number of subjects by treatment and visit. Any summary procedure can be used here.

```
proc summary data=work.pain nway;
   class trt avisitn;
var aval;
output out=bar mean=mean n=n stderr=std;
run;
```

For bubble plot, we need the number of subjects for unique analysis value at a given visit. A simple PROC FREQ step can be used to get the counts:

```
proc freq data=pain noprint;
    tables trt*avalc*avisitn/out=count nopercent;
run;
```

CREATING A TEMPLATE

Ideally, you can find standard templates and reporting macros in an organizational set up and you can use the same and modify based on the requirement. If a standard template is not available, you will have

to create the template. The template below is used to create the overlay graphs in RTF format using PROC SGPLOT and PROC SGPANEL:

```
proc template;
  define style mypatt;
  parent = Styles.Rtf;
  replace fonts
     "Fonts used in the default style" /
     'TitleFont2' = ("Courier New", 8pt)
     'TitleFont' = ("Courier New", 8pt)
     'StrongFont' = ("Courier New", 8pt)
     'EmphasisFont' = ("Courier New", 8pt)
     'FixedEmphasisFont' = ("Courier New", 8pt)
     'FixedStrongFont' = ("Courier New", 8pt)
     'FixedHeadingFont' = ("Courier New", 8pt)
     'BatchFixedFont' = ("Courier New", 8pt)
     'FixedFont' = ("Courier New", 8pt)
     'headingEmphasisFont' = ("Courier New", 8pt)
     'headingFont' = ("Courier New", 8pt)
     'docFont' = ("Courier New", 8pt);


  style GraphData1 from GraphData1 /
        color=steel
        contrastcolor=blue;

  style GraphData2 from GraphData2 /
          color=salmon
        contrastcolor=red;
 end;
run;
```

This template is used to define the display attributes. Display Attributes are the attribute settings that you can specify for the lines, data markers, text, or area fills in a plot. For grouped data, defaults for these features are set by the linestyle, color, contrast Color, fillpattern, and markersymbol attributes of the GraphData1–GraphDataN style elements. This template used with ODS statements delivers the required output in RTF.

OVERLAY GRAPH

BUBBLE PLOT OVER VERTICAL BAR CHART

The next step is to create the overlay graph. In this scenario, we want to show mean pain score data along with number of subjects who have reported a particular pain score at each visit and compare it between treatment groups. Figure 1 shows vertical bar chart for mean pain score data at each visit for both groups. Figure 2 shows bubble plot with subject pain score data at each visit.

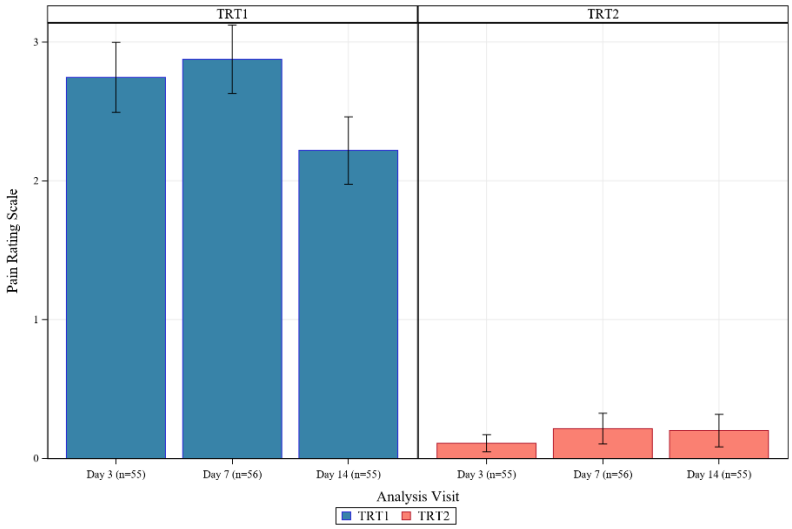

Figure 1. Vertical Bar Chart showing pain score over time for treatments TRT1 and TRT2. The limit lines represent mean+/-stderr.

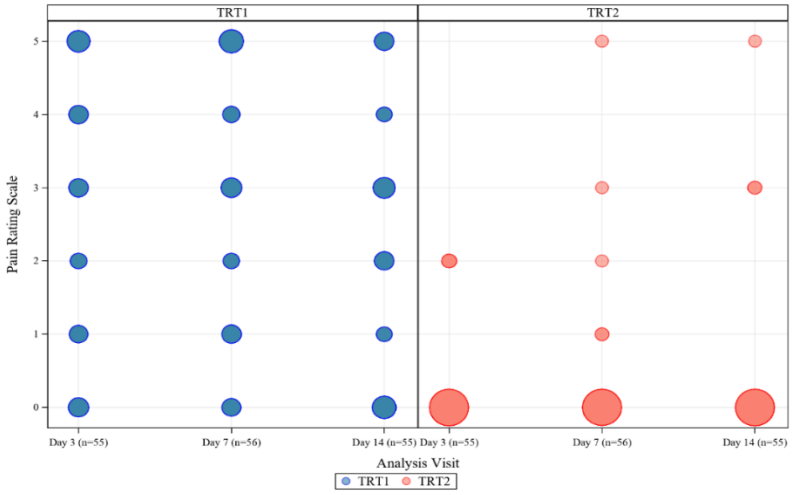

Figure 2. Bubble chart of Patient pain score over time by treatment.

PROC SGPANEL codes used for the above two graphs are as follows. PANELBY statement specifies classification variable, in this case it is treatment group. For both figures, data is grouped by treatment and GROUPDISPLAY is cluster, meaning both the treatment groups will be seen side by side. LIMITLOWER is calculated as mean-stderr. LIMITUPPER is calculated as mean+/-stderr. The position of legend is defined in KEYLEGEND statement. X-axis and Y-axis label attributes are defined in ROWAXIS and COLAXIS statements. The SIZE option in BUBBLE statement is directly proportional to the number of subjects by visit.

Code for Figure 1:

```
proc sgpanel data=bar noautolegend;
   panelby trt/novarname;
   VBARPARM CATEGORY=avisitn RESPONSE=mean /group=trt groupdisplay=CLUSTER
   LIMITLOWER=low LIMITUPPER=up;
   colaxis label="Analysis Visit" grid valueattrs=(size=8pt) integer;
   rowaxis label= "Pain Rating Scale" grid valueattrs=(size=8pt) integer;
```

```
  keylegend / position=bottom title='';
run;
```

Code for Figure 2:

```
proc sgpanel data=bar noautolegend;
   panelby trt/novarname;
   bubble x=avisitn y=aval size=count/ group=trt transparency=0.4;
   colaxis label="Analysis Visit" grid valueattrs=(size=8pt) integer;
   rowaxis label= Pain Rating Scale" grid valueattrs=(size=8pt) integer;
 keylegend / position=bottom title='';
run;
```

Figures 1 and 2 give us a good idea about mean pain score data and subject pain score data by treatment and visit. But what if we overlay both these graphs?

Figure 3 shows overlay graph of bubble plot over vertical bar chart which gives the reviewer a clear picture of the efficiency of the treatment. The treatment with mean pain score and number of subjects with pain by visit are represented in the graph below.

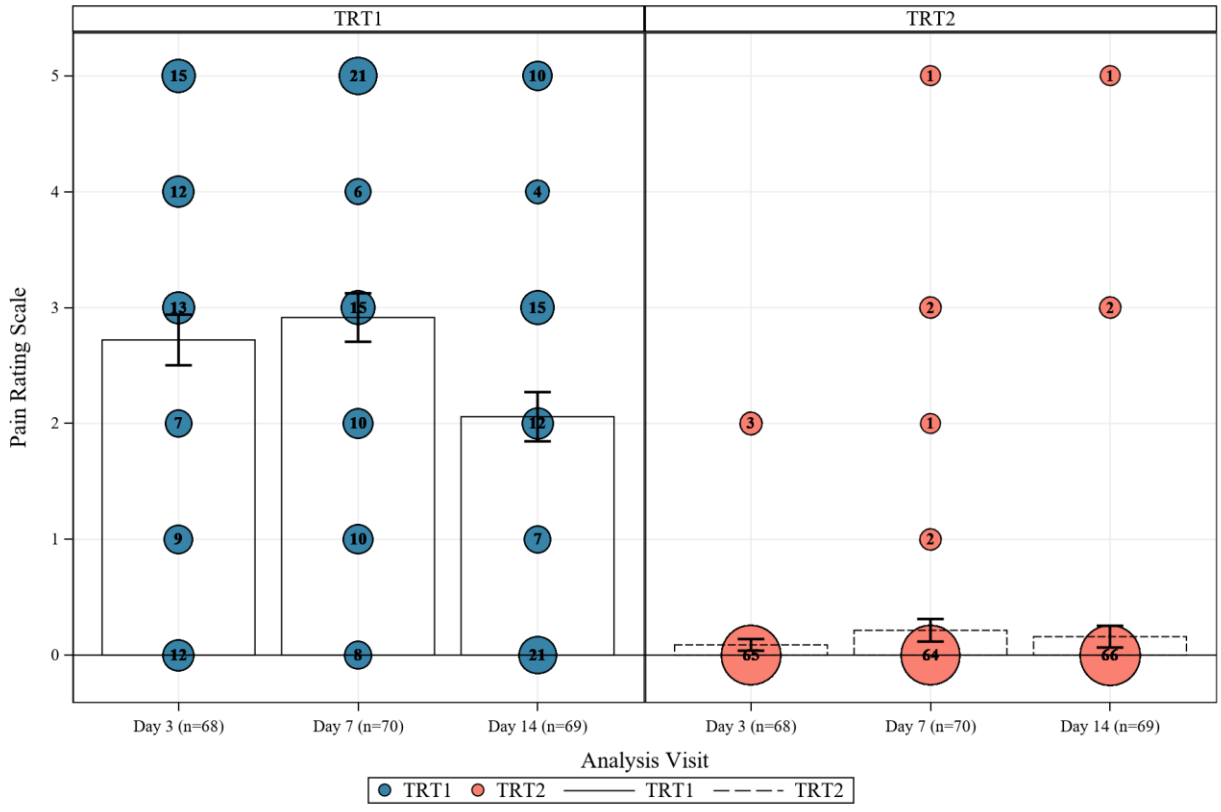

Figure 3. Overlay graph of subject pain score data over mean pain score +/-stderr at different visits by treatment.

The code used for creating overlay graph in Figure 3:

To show the number of subjects at each visit inside the bubble DATALABEL and DATALABELPOS options are used in the bubble statement. To show the bubbles for pain score 0 on x-axis clearly OFFSETMIN option is used. The procedure draws the plots in your graph in the same order that you specify the plot statements.

```
proc sgpanel data=bar noautolegend;
  panelby _tc/novarname;

  BUBBLE x=avisitn y=aval size=count/ group=trt datalabel=count
   datalabelpos=center datalabelattrs=(weight=bold) transparency=0.0;
  VBARPARM CATEGORY=avisitn RESPONSE=mean /group=trt  groupdisplay=CLUSTER
    limitattrs=(pattern=solid thickness=2pt) LIMITLOWER=low LIMITUPPER=up
    outline TRANSPARENCY=0.0;

   colaxis label="Analysis Visit" grid valueattrs=(size=8pt) integer;
   rowaxis label= "Pain Rating Scale" grid valueattrs=(size=8pt) integer
   offsetmin=0.07;
  keylegend / position=bottom title='';
 run;
```

SCATTER PLOT OVER VERTICAL BAR CHART

The above data can be displayed using a different overlay graph, scatter over vertical bar. The most common SAS® graphical procedure SGPLOT is used to get this overlay. In earlier versions of SAS® scatter and bubble plots were incompatible with VBAR (vertical bar chart). In SAS9.4M3 and later releases, this issue has been tackled by HBARBASIC and VBARBASIC statements. Also, to achieve the required attributes in RTF file format we must use a template with SGPLOT. This overlay is one more way of visualizing the pain score data for the two treatment groups. The code below Is used to get Figure 4 using SGPLOT:

```
proc sgplot data=_bar1 noautolegend;

  vbarparm CATEGORY=avisitn RESPONSE=mean /group=trt groupdisplay=CLUSTER
  limitlower=low limitupper=up;

   styleattrs datacontrastcolors=(black) datasymbols=(triangledown
    circlefilled);
   scatter x=avisitn y=aval/group=trt groupdisplay=cluster jitter
  transparency=0.1 clusterwidth=0.4;
  yaxis min=0 label="Pain Rating Scale" valueattrs=(size=8pt);
  xaxis max=6 label="Analysis Visit" valueattrs=(size=8pt);
  keylegend / location=outside position=bottom title='';
 run;
```

The style attributes statement applies to scatter plot and the graph template mentioned earlier in Page 2 has been used for VBARPARM style attributes.

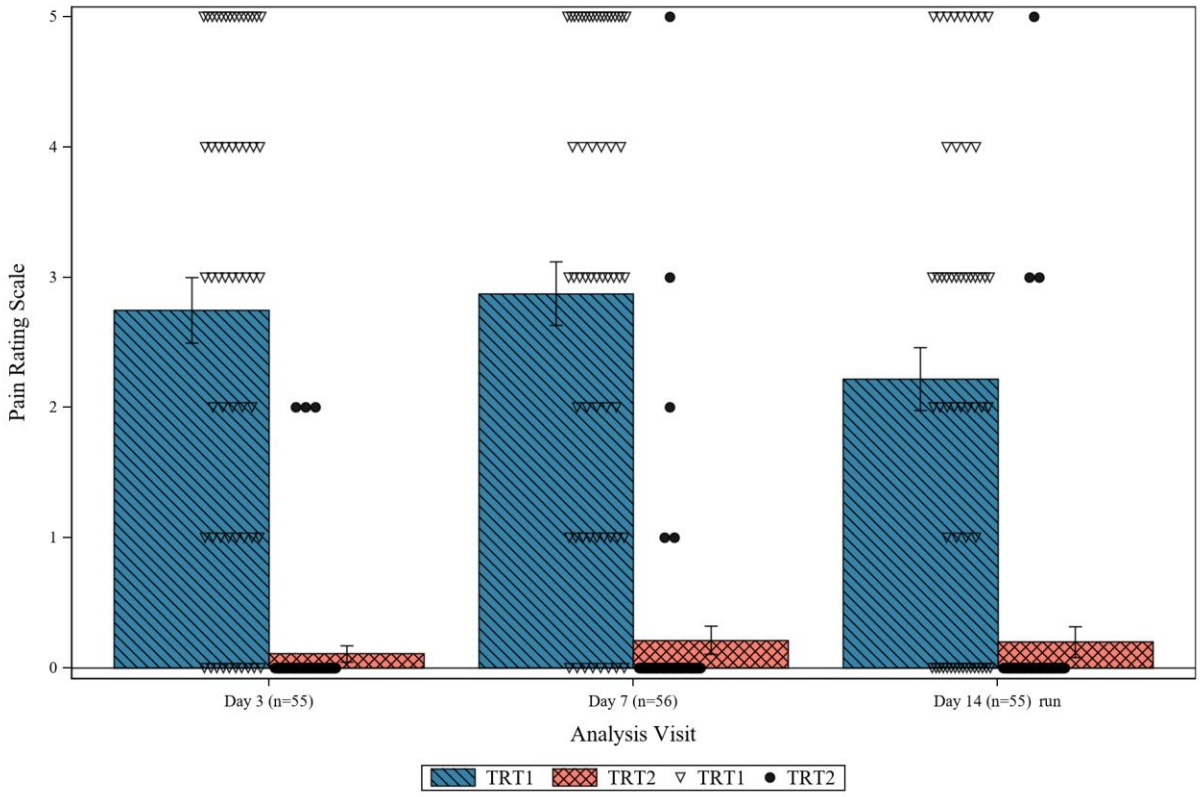

Figure 4. Overlay graph of subject pain score data over mean pain score +/-stderr by treatment by visit using PROC SGPLOT and PROC SGRENDER.

Procedure SGRENDER can also be used to get the same graph, but with a different template. The upper and lower limits for mean are calculated for vertical bar chart limit lines using options ERRORLOWER and ERRORUPPER in BARCHARTPARM statement in PROC TEMPLATE. JITTER specifies the data markers are offset when multiple observations have the same response value. DISCRETELEGEND creates a legend with entries that refer to plots, or group values, or both legend items. The code for Figure 4 using SGRENDER is shown below:

```
proc template;
   define statgraph scatter;
   begingraph;
         layout overlay /
               xaxisopts=(label="Analysis Visit")
               yaxisopts=(label="Pain Rating Scale");
         barchartparm category=avisitn response=mean / group=trt
            groupdisplay=cluster name='trt'
            errorlower=eval(mean-std)
            errorupper=eval(mean+std;
        scatterplot x=avisitn y=aval /group=_tc groupdisplay=cluster
        name='_tc2' datatransparency=0.30 jitter=auto;
        discretelegend 'trt1' 'trt2'/across=1 down=1 valign=bottom
         order=columnmajor halign=center;
     endlayout;
  endgraph; end;
run;

proc sgrender data=bar template=scatter;
run;
```

CONCLUSION

Presenting multiple analyses in a single graph gives a complete picture of the data to the reviewer. Creating overlay graphs might seem like a daunting task at first but once you start visualizing the trends in the data, you will only want to do more. This paper intends to inspire readers to explore different ways in order to get the best visualization for any kind of analysis.

REFERENCES

Matange, Sanjay. "*Cluster Groups*". *From Graphically Speaking*. March 30, 2012. https://blogs.sas.com/content/graphicallyspeaking/2012/03/30/cluster-groups/

Richann Watson. (2020). "Great Time to Learn GTL: A Step-by-Step Approach to Creating the Impossible." *Proceedings of PharmaSUG 2020.* https://www.lexjansen.com/pharmasug/2020/DV/PharmaSUG-2020-DV-009.pdf

ACKNOWLEDGMENTS

The author would like to thank Shen Hongfang for all the discussions and support while preparing the paper. Thank you Pavani Potuganti and Divyaja Padamati for reviewing this paper.

RECOMMENDED READING

- Matange, Sanjay. Clinical Graphs Using SAS®, SAS Institute (SAS Press, 2016).

CONTACT INFORMATION

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

Manasa Gangula
Cytel, Inc.
manasa526.sas@gmail.com