PharmaSUG 2016 - Paper PO10

Tips and Tricks for Bar Charts using SAS/Graph® Template Language

Randall Nordfors, Seattle Genetics, Inc., Bothell, WA
Boxun Zhang, Seattle Genetics, Inc., Bothell, WA

ABSTRACT

Bar charts are frequently used in clinical reporting to visually display quantitative information and identify patterns for a variety of purposes. SAS/Graph® Template Language (GTL) provides powerful functionality to create sophisticated analytical graphics independent of the need for statistical procedures.

Programmers may face situations where a conceptual idea for a graph display style is specified by a biostatistician or medical writer, even though there may not be a natural programming technique, or available creative precedent, for generating it using SAS/Graph®.

In this paper, using selected tips and tricks, we focus on ways to help you solve bar chart programming challenges efficiently. Our examples included in this paper are:

- How to create mirror bar chart for side by side comparison around a single zero point on the x-axis.
- How to display multiple distinct bar pairs within a single axis graphing infrastructure. The trick is that they only appear to be independent, while in actuality the data are from a single source.

INTRODUCTION

The graphics produced by the SAS/Graph® GTL are generated by template definitions that control the graph format and appearance and specify the variable roles to represent in the graph display. The GTL provides the possibility to create customized graphs as requested. The graphs can then be rendered by associating the templates with a data source.

PART I: MIRROR BAR CHART

Side by side comparisons allow the viewer to easily find differences among data points across groups, in an intuitive way. It’s easy to observe the difference by displaying two groups along a shared vertical axis. With the identical scale, the histograms are centralized in the middle and laid out side by side for each group. This kind of bar chart can be thought of as the reflection of a mirror. So it is often referred to as a mirror bar chart. The visual comparison between the left and right side is obvious to see difference and identify patterns.

In this paper, the adverse events incidence rate in two different cohorts of a study (Part A and Part B) is used to illustrate the tips for the mirror bar chart. As shown in Figure 1, two groups are drawn together with the single vertical axis. Part A is on the left side and Part B is on the right side. Both bar charts share the same plot area, axes and legends. The data come from one dataset and the bar charts are generated by two LAYOUT OVERLAY statements in GTL. Distinct colors indicate the different severity of the adverse events. From top to bottom, the adverse events are in the descending order by the overall incidence in the total population.

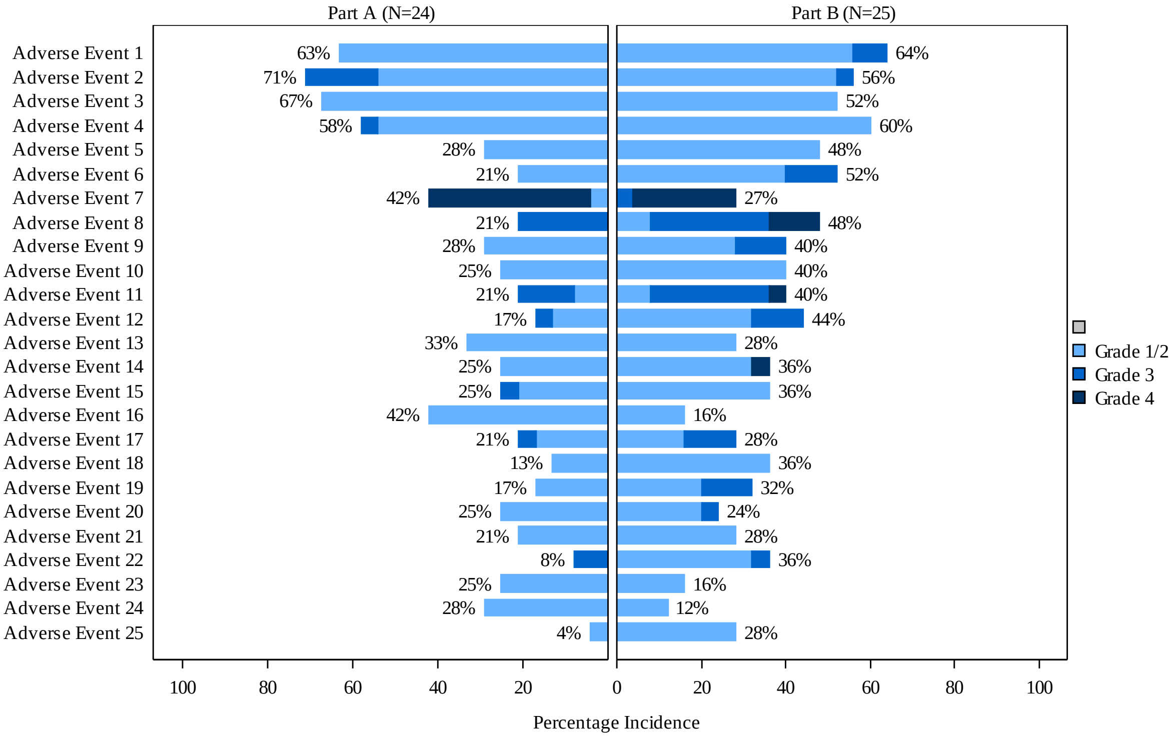


Figure 1. Mirror Bar Chart: Adverse Event Occurring in ≥ 25% of Patients by Study Part

Figure 1 is the output from the SAS code below. To make it easy to explain the selected tips, the source codes are annotated by number.

```
proc format;
  value multi                                                          ❶
       0='0'
    0.20='20'
    0.40='40'
    0.60='60'
    0.80='80'
    1.00='100'
   -0.20='20'
   -0.40='40'
   -0.60='60'
   -0.80='80'
   -1.00='100';

  picture pctpic (round)                                               ❷
    low-<0=009% (mult=100)
    0=' '
    0<-high=009% (mult=100);
run;

proc template;
  define style style_bh;
  parent=rtftnr10;
    style graphdata1 from graphdata1 /color=CX66B2FF linestyle=1; **Light Blue**;
    style graphdata2 from graphdata2 /color=CX0066CC linestyle=1; **Medium Blue**;
    style graphdata3 from graphdata3 /color=CX003366 linestyle=1; **Navy**;
    style graphdata4 from graphdata4 /color=Cx3CB371 linestyle=1;
  end;
run;
```

```
data _final;                                                                 ❸
  set group_a(rename=(aedecod=aedecod_1 grp=grp_1))
      group_b(rename=(aedecod=aedecod_2 grp=grp_2));
  format percent pctpic.;
run;

proc template;
  define statgraph sgdesign;                                                 ❹
  dynamic _aeterm_1 _aeterm_2 _count_1 _count_2 _grp_1 _grp_2;
  begingraph / designwidth=850 designheight=520 border=false;
    entryfootnote "                            Percentage Incidence";
    layout lattice / rows=1 rowdatarange=union columns=2 columndatarange=union;
    column2headers;
      entry "Part A (N=&total_1)";
      entry "Part B (N=&total_2)";
    endcolumn2headers;
    layout overlay / cycleattrs=true                                         ❺
                     xaxisopts=(reverse=false display=(TICKS TICKVALUES LINE)
                     griddisplay=off tickvalueattrs=GraphValueText
                     linearopts=(viewmin=-1.05 viewmax=0 tickvaluelist=(-.2 -.4 -.6
-.8 -1) tickvalueFormat=multi.))                                             ❻
                     yaxisopts=(display=(TICKS TICKVALUES LINE));
      barchart x=_aeterm_1 y=_count_1 / group=_grp_1 name='bar(h)'
                                        display=(FILL)
                                        barlabel=true
                                        orient=horizontal
                                        barwidth=0.7
                                        clusterwidth=0.3;
    endlayout;
    layout overlay / cycleattrs=true                                         ❼
                     xaxisopts=(reverse=false display=(TICKS TICKVALUES LINE)
                     griddisplay=off tickvalueattrs=GraphValueText
                     linearopts=(viewmin=0 viewmax=1.05 tickvaluelist=(0 .2 .4 .6
.8 1) tickvalueFormat=multi.))
                     yaxisopts=(display=(TICKS TICKVALUES LINE));
      barchart x=_aeterm_2 y=_count_2 / group=_grp_2 name='bar(h)'
                                        display=(FILL)
                                        barlabel=true
                                        orient=horizontal
                                        barwidth=0.7
                                        clusterwidth=0.3;
    endlayout;
    rowaxes;
      rowaxis / griddisplay=off display=(label tickvalues)
                label=" "  labelattrs=(weight=bold);
    endrowaxes;
    sidebar / align=right;
      discretelegend 'bar(h)' / opaque=false border=false order=rowmajor across=1
sortorder=ascendingformatted;
    endsidebar;
    endlayout;
    endgraph;
  end;
run;

ods _all_ close;
ods proclabel=' ';
ods graphics on/reset imagefmt=emf;
ods rtf file="&outputname"  path="&outdir." style= style_bh nogfootnote nogtitle
bodytitle toc_data headery=720 footery=720;

proc sgrender data=_final template=sgdesign;
```

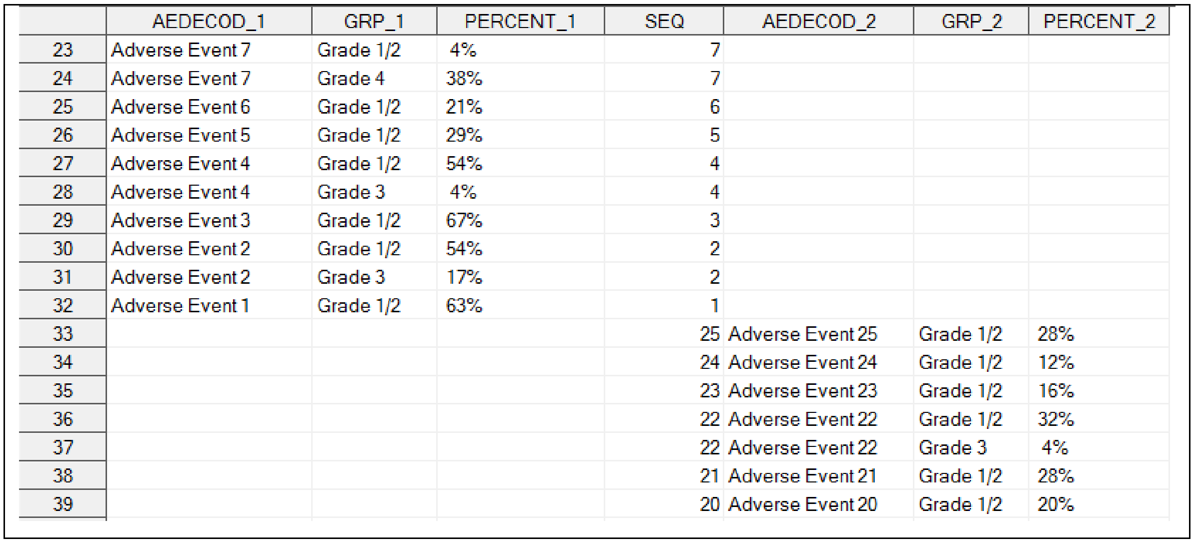
```
    dynamic _aeterm_1="aedecod_1" _aeterm_2="aedecod_2" _count_1="percent_1"
  _count_2="percent_2" _grp_1="grp_1" _grp_2="grp_2";
  run;

  ods rtf close;
  ods listing;
```

TIPS AND TRICKS

❶ In order to draw two charts side by side, Part A, on the left side in the graph, is assigned as negative values. By doing this, Group A can take the left quadrant. If we use positive values for two groups, all the histograms will be on the right side of the y-axis. As we want to present the graph as a mirror, we don't want to show negative values. The negative values need to be assigned to a new format, so they can be shown as positive values.

❷ Format PCTPIC is used to convert the decimal number to percentage, regardless negative or positive. Zero is assigned as a blank because it is not used in the figure. The percentage is used for the annotation to the incidence rate.

❸ Both Group A and B come from one dataset, but the data has to be in a special structure. This example concatenates two SAS datasets into one dataset and uses separate variables for each group. The output dataset is shown below:



	AEDECOD_1	GRP_1	PERCENT_1	SEQ	AEDECOD_2	GRP_2	PERCENT_2
23	Adverse Event 7	Grade 1/2	4%	7			
24	Adverse Event 7	Grade 4	38%	7			
25	Adverse Event 6	Grade 1/2	21%	6			
26	Adverse Event 5	Grade 1/2	29%	5			
27	Adverse Event 4	Grade 1/2	54%	4			
28	Adverse Event 4	Grade 3	4%	4			
29	Adverse Event 3	Grade 1/2	67%	3			
30	Adverse Event 2	Grade 1/2	54%	2			
31	Adverse Event 2	Grade 3	17%	2			
32	Adverse Event 1	Grade 1/2	63%	1			
33				25	Adverse Event 25	Grade 1/2	28%
34				24	Adverse Event 24	Grade 1/2	12%
35				23	Adverse Event 23	Grade 1/2	16%
36				22	Adverse Event 22	Grade 1/2	32%
37				22	Adverse Event 22	Grade 3	4%
38				21	Adverse Event 21	Grade 1/2	28%
39				20	Adverse Event 20	Grade 1/2	20%

❹ As driven by the dataset structure, six variables (AEDECOD_1, GRP_1, PERCENT_1, AEDECOD_2, GRP_2, PERCENT_2) are defined in the PROC TEMPLATE. The graph will then be rendered by associating the templates with the data source. Please note, SEQ variable is used for sorting the adverse events, but it is not needed in the GTL.

❺ The LAYOUT OVERLAY in this example specifies BARCHART with histograms showing the distribution of observations of the data columns named _AETERM_1 and _COUNT_1 for Group A.

❻ The TICKVALUELIST= option is used in the LINEAROPTS= option to display all of the values on the x axis. The TICKVALUEFORMAT= option is used to assign the new format, as explained in ❶.

❼ The LAYOUT OVERLAY can overlay the results from the other plot statement, as indicated ❺ above. Both plots share the same plot area, axes and legends. BARCHART from this part generates the plot for Group B.

PART II: MULTIPLE BAR SETS ON A SINGLE BAR CHART

For ease of visual comparison, it can be useful to display multiple bar sets within a single bar chart structure, when they share a common axis. This is further accomplished by treating what would normally be the opposite axis, in 2-dimensional space, as a location for labeling/indicating which group a particular set of bars represents. In this way you can conveniently compare groups on a single chart. The example presented here is a chart for three groups, each with a pair of bars.

Normally, a chart visually presents one set of data, but it's not necessary to deviate from that in the programmatic context when using the following technique. Arranging for three sets/groups of data to be simultaneously contained within the one dataset processed by Proc Template covers one aspect of the technique (tip and trick).

The second aspect is to visually isolate each group without its proximity to adjoining groups being so close that there is no distinction among them. The solution is to insert dummy records between groups and assign it a very small negative frequency, -0.001, such that its depth gives it the appearance of a chart boarder line that helps frame the x-axis labels. Yet in actuality, they are simply very small negative frequency bars.

The third key element of this technique is to display the group labels (in this case analyte percentage ranges) such that they are each positioned below their appropriate group, in the location normally reserved for x-axis tick marks.

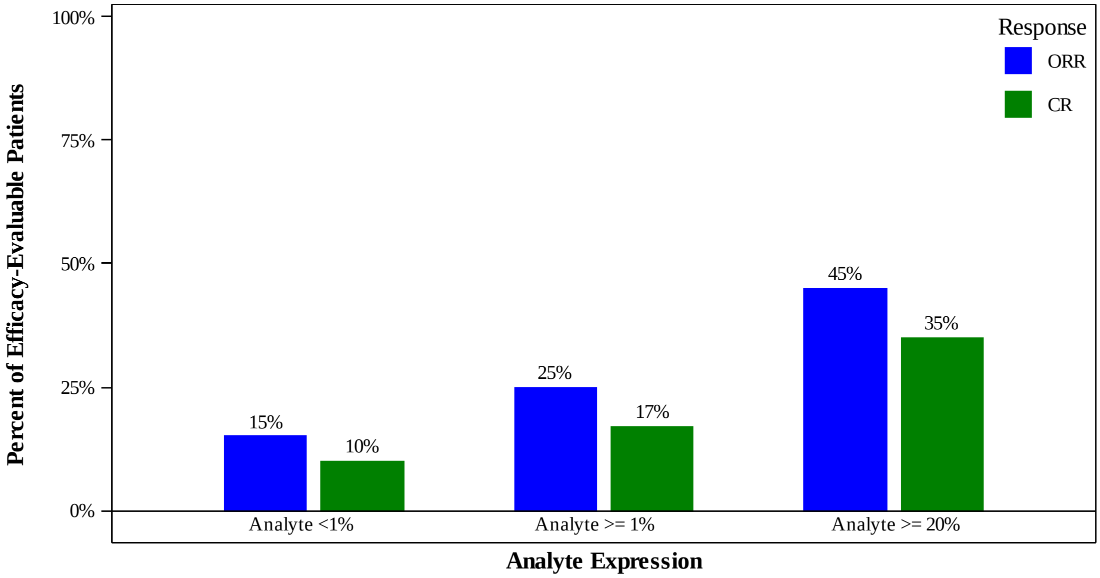
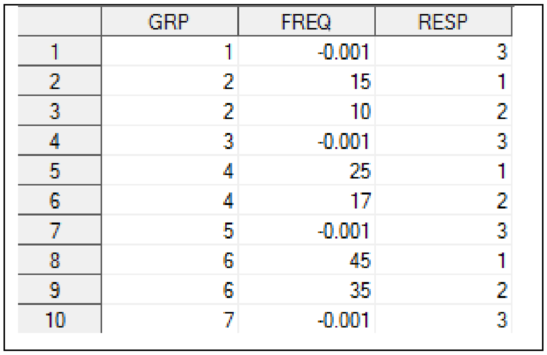


Figure 2. Side-by-Side Bar Charts: Objective and Complete Response Rates Grouped by % Analyte Cutoff

After processing the input data to calculate frequencies and setting up the structure for displaying three groups in one, the SAS Dataset looks like this:

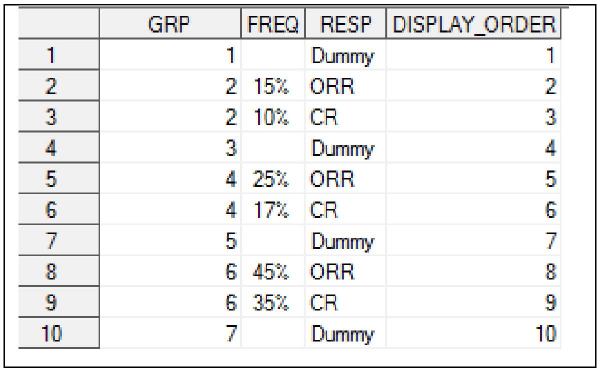


	GRP	FREQ	RESP
1	1	-0.001	3
2	2	15	1
3	2	10	2
4	3	-0.001	3
5	4	25	1
6	4	17	2
7	5	-0.001	3
8	6	45	1
9	6	35	2
10	7	-0.001	3

After applying these formats:

```
proc format;
  value resp
    1='ORR'
    2='CR'
    3='Dummy';

  picture pctpic (round)
    -0.001<-0=009% (mult=1)
    -0.001=' '
    0<-high=009% (mult=1);
run;
```

…the dataset, "work.final" looks like this, and is used as the input dataset for the PROC SGRENDER call:



	GRP	FREQ	RESP	DISPLAY_ORDER
1	1		Dummy	1
2	2	15%	ORR	2
3	2	10%	CR	3
4	3		Dummy	4
5	4	25%	ORR	5
6	4	17%	CR	6
7	5		Dummy	7
8	6	45%	ORR	8
9	6	35%	CR	9
10	7		Dummy	10

Below are the Proc Template and Proc SGRENDER statements that are used to read the dataset and produce the bar chart:

```
proc template;
  define style sty_km;
  parent=rtftnr10;
  style graphfonts from graphfonts /
    'GraphDataFont'    =("times new roman", 10pt)
    'GraphDataValue'   =("times new roman",10pt)
    'GraphLabelFont'   =('times new roman',12pt)
    'GraphTitleFont'   =('Arial',11pt)
    'GraphFootnoteFont'=('times new roman',10pt);

     style graphdata1 from graphdata1 /color=red   contrastcolor=red linestyle=1
                                                   markersymbol='squarefilled';
      style graphdata2 from graphdata2 /color=blue  contrastcolor=blue linestyle=1
                                                    markersymbol='squarefilled';
      style graphdata3 from graphdata3 /color=green contrastcolor=green linestyle=1
                                                    markersymbol='squarefilled';
 end;
run;

proc template;
 define statgraph sgdesign;
 dynamic _DISPLAY_ORDER _FREQ _RESP ;                                       ❶
  begingraph/designwidth=750 designheight=400 border=false;
   layout lattice / rowdatarange=data columndatarange=data rowgutter=10
                    columngutter=10;

  legenditem  type=marker name="resp_ord1" / label="ORR" markerattrs=graphdata2
                                             (size=20);
  legenditem  type=marker name="resp_ord2" / label="CR"  markerattrs=graphdata3
                                             (size=20);
    layout overlay / xaxisopts=( display=(LINE LABEL ) label=("Analyte Expression")
                     labelattrs=(weight=bold))
                     yaxisopts=(label=('Percent of Efficacy-
                                Evaluable Patients') labelattrs=(weight=bold)
                                tickvalueattrs=GraphValueText
    linearopts=( viewmin=-5 viewmax=101.0 tickvaluesequence=( start=0 end=100.0   ❷
                 increment=25.0))) cycleattrs=true;
    barchart x=_DISPLAY_ORDER                                                      ❸
             y=_FREQ / group=_RESP
                       name='bar'  barlabel=true
                       display=(FILL) clusterwidth=1.0;


          discretelegend 'resp_ord1' 'resp_ord2'  /
                                               opaque=false
                                               border=false
                                               halign=right
                                               valign=top
                                               title='Response'
                                               displayclipped=true
                                               across=1
                                               order=rowmajor
                                               location=inside;

         entry halign=center "Analyte <1%                                         ❹
Analyte >= 1%                                      Analyte >= 20%" / valign=bottom;

        endlayout;
     endlayout;
    endgraph;
   end;
  run;

  proc sgrender data=WORK.final template=sgdesign;                               ❺
   dynamic _DISPLAY_ORDER="DISPLAY_ORDER" _FREQ="FREQ" _RESP="RESP";
  run;
```

TIPS AND TRICKS

❶ The variable `_DISPLAY_ORDER` is used in place of x-axis tick marks, only to establish the order in which the dataset rows will be displayed on the chart horizontally. The order values are not displayed. What is displayed is discussed in ❹.

❷ A value of -5 for viewmin creates space, and the illusion that the group labels are normally-positioned x-axis labels, but in fact they are technically inside the chart.

❸ Another statement, demonstrating that `DISPLAY_ORDER` is used as the bar chart x-axis index instead of data points.

❹ A single “entry” statement containing one long character string that contains all three group labels. It wraps on this 8.5x11 inch page because the number of blank spaces between groups in critical to where the groups are positioned on the chart.

❺ The PROC SGRENDER procedure showing the choice of input dataset (shown earlier in this paper, post-formatted), and the dynamic element: Display order (across the x-axis), frequency (percentage of patients in a particular group and response category, and response (legend).

CONCLUSION

In this paper we described the use of tips and tricks that are solutions for producing creative bar charts. Their application was instrumental in accomplishing design challenges we faced in business. This exploration into the use of SAS/Graph® GLT need not be limited to the examples and techniques provided here. Further customizations, or altogether different methods while still using GTL, are possible in the production of additional bar chart designs.

REFERENCES

Sanjay Matange, 2011. “Tips and Tricks for Clinical Graphs using ODS Graphics” *Proceedings of the SAS Global Forum 2011 Conference*. Cary, NC: SAS Institute Inc. Available at https://support.sas.com/resources/papers/proceedings11/281-2011.pdf

ACKNOWLEDGMENTS

We would like to thank Rajeev Karanam and Shefalica Chand for encouraging us to write this paper.

CONTACT INFORMATION

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

Randall Nordfors
Seattle Genetics, Inc.
21823 30th Drive Southeast
Bothell, WA 98021
E-mail: rnordfors@seagen.com

Boxun Zhang
Seattle Genetics, Inc.
21823 30th Drive Southeast
Bothell, WA 98021
E-mail: bzhang@seagen.com