

Amazing Graph Series: Butterfly Graph Using SGPLOT and GTL

Tracy Sherman, Efficacy Consulting Group, Inc., Kelowna, BC.;
Aakar Shah, Acadia Pharmaceuticals Inc., San Diego, CA

ABSTRACT

Have you been tasked to create a butterfly graph? If so, you might be asking yourself, what is a butterfly graph and which SAS® procedure should I use to create one? Which procedure will be the most straight forward to learn but also has the flexibility for custom modifications? The two most common methods used to create these graphs are using the SGPLOT procedure and the GTL (graph template language). The documentation for these procedures to enhance the visual appearance of these graphs is lengthy and cumbersome. This paper will help you learn the syntax required and narrow the time it takes to produce high quality and amazing butterfly graphs which can be shared with upper management or in a conference presentation. In addition, the paper also compares the SAS 9.4 SGPLOT procedure and GTL so that you can choose which method fits well with your programming requirements.

INTRODUCTION

A butterfly graph is similar to a comparative bar chart or histogram that resembles the shape of a butterfly and is a recommended analysis when there is a need to display two values side by side with a grouping variable. In our example, the butterfly graph will be used to display the percentage of patients with specific adverse events by two grouping variables, treatment and CTCAE toxicity grade.

The main purpose of this paper is to give programmers a description of the syntax and an understanding of the similarities and differences between the SAS 9.4 SGPLOT procedure and GTL to produce high quality butterfly graphs. And along the way, it shines light on various options, tips and tricks associated with these procedures. This paper is one-stop shop to find what you need to provide best graphical value in your organization.

You might be wondering what are the differences between SGPLOT and GTL? Briefly, SGPLOT is one of the “SG” (Statistical Graphics) procedures that base SAS provides for creating stand-alone plots essential for exploring data and for constructing specialized displays for various analyses. GTL on the other hand is an extension of the TEMPLATE procedure, combines layouts and plots in flexible ways, supports statistical computations and plot types, supports ODS styles for a variety of usages, is a power tool for user-creation of complex analytical graphs and is the same tool used to create the automatic graphics that is produced by all SAS statistical procedures (Rodriguez and Kuhfeld, 2016). Essentially, GTL is the foundation upon which the SG procedures are built.

To produce the butterfly graph, we will use both SGPLOT and GTL to contrast these two methods. Due to the nature of SGPLOT, a single-celled graph will be shown with the percentage of patients arranged on each side of the axis and the preferred terms displayed on the left-hand side. As GTL has more flexibility, we will create a multi-cell graph with three overlays to create an efficient treatment comparison of the percentage of patients with specific preferred terms displayed down the middle of the graph by CTCAE toxicity grade.

SOURCE DATA AND FULL SAS CODE

Source data used throughout the paper consists of data from a dummy analysis data set of adverse event (ADAE). Post-processing of ADAE to add percentage of patients for each preferred term and toxicity grade is required before submitting the GTL and SGPLOT SAS code (see APPENDIX I DUMMY SOURCE DATA SET and POST-PROCESSING, APPENDIX II SGPLOT SAS CODE and APPENDIX III GTL SAS CODE).

MANAGING YOUR GRAPHICS WITH ODS

The Output Delivery System (ODS) manages all output created by procedures and enables you to display the output in a variety of formats, such as HTML, PDF, and RTF. The SAS ODS Graphics procedures such as PROC SGPLOT and GTL use ODS Graphics for creation of their graphs. You can use the ODS GRAPHICS statement options to control many aspects of your graphics including the size of the image, the type and name of the image created. The basic syntax for the ODS GRAPHICS statement is as follows:

```
ods graphics < off | on> </ options>;
```

For further details on how to use ODS to output your graphs, please see Shah and Sherman, 2021.

SINGLE-CELL BUTTERFLY GRAPH USING THE SGPLOT PROCEDURE

Using the SGPLOT procedure with two HBAR statements (one for each treatment) and the GROUP= option, you can create a single-celled butterfly graph (see Figure 1). Here we are displaying treatment-emergent adverse events by treatment and CTCAE toxicity grade. Ordering the preferred terms by overall treatment percentage of patients, places the highest percentage at the top of the graph which creates the butterfly effect. We have used only most frequent TEAEs occurring in $\geq 10\%$ of total patients.

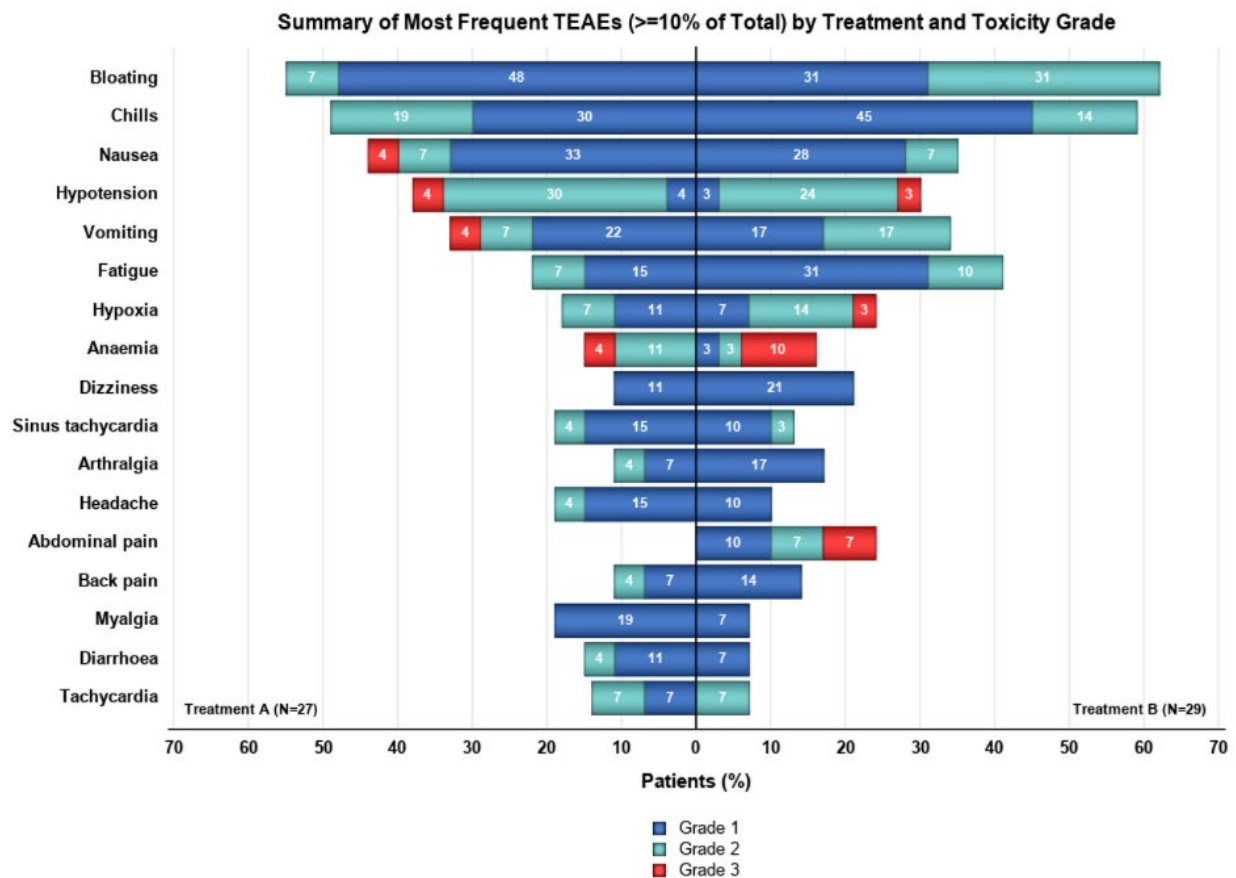


Figure 1. SGPLOT Butterfly Single-cell Graph of Most Frequent TEAEs by Treatment and Toxicity Grade

It is good to point out here that the values to the left of the centered axis are negative numbers, and conversely the values to the right are positive. A PROC FORMAT is used to display the negative numbers with positive tick values. In Figure 1, the left (negative) side displays horizontal bars for Treatment A with

each toxicity grade shaded in different colors (blue, green, and red), and the right (positive) side displays horizontal bars for Treatment B.

The following code was used to create the user-defined format for the left (negative) side of the butterfly graph:

```
proc format;
  picture positive
  low - < 0 = "0000"
  0 < - high = "0000";
run;
```

This DATA step creates the discrete attribute map data set MYATTRMAP. The values for the variable ID in the attribute map data set are 'myid'. In this example we have only used attributes value and fillcolor.

```
data myattrmap;
  input ID $ 1-4 value $ 6-12 fillcolor $;
  datalines;
    myid Grade 1 CX0B4CB5
    myid Grade 2 bibg
    myid Grade 3 red
    myid Grade 4 purple
;
run;
```

To display title and footnotes, use the TITLE and FOOTNOTE statements. The title position is defaulted to the center of the graph. You can define the data extraction date macro variable, &dexdt, into your initialization program for convenience:

```
/* Standard macro variables for Data Extraction Date used in footnoting */
proc sql noprint;
  select compress(put(datepart(crdate), date9.)) into: dexdt trimmed
  from sashelp.vtable
  where libname = 'RAW' and memname = 'DM2';
quit;

title "Summary of Most Frequent TEAEs (>=10% of Total) by Treatment and
Toxicity Grade";
footnote j = right "Data Extraction Date: &dexdt.";
```

- 1 The SGPLOT statement includes the NOBORDER, NOWALL to reduce the clutter of the graph and includes a SG attribute map option, DATTRMAP.
- 2 The FORMAT statement associates the variable _1 with the format POSITIVE as defined in the code to display the negative numbers with positive tick values.
- 3 Two HBAR statements with variables _1 and _2 (percentage of patients for AEDECOD by each treatment) as the RESPONSE variable and toxicity grade (grdcat) as the GROUP = option. The GROUPDISPLAY = stack option enables the toxicity grade to be stacked onto each horizontal bar. ATTRID = myid specifies the value of the ID variable in the attribute map data set. The second HBAR statement does not require the NAME option.
- 4 The segment label options, SEGLABEL, SEGLABELFITPOLICY, and SEGLABELATTRS were used to define the attributes such as fit, size, color, and weight of the labels on each horizontal bar.
- 5 XAXIS statement adds options for removing the tick marks (DISPLAY = (noticks)), including the vertical solid GRID lines, axis label (LABEL) and value attributes.
- 6 YAXIS statement adds options for removing line, labels and tick marks (DISPLAY = (noline nolabel noticks) and value attributes such as bolding (VALUEATTRS = (weight = bold)).

- 7 The KEYLEGEND statement enables you to specify how you want the legend displayed. Here we have specified that the legend should have three vertical rows (DOWN=3), toxicity grades to be displayed as Grade 1-Grade 3, with no border (NOBORDER) and no legend title (TITLE = ' ').
- 8 Use the INSET statement to add a text box inside the axes of the plot. We added 'Treatment A (N = xx)' and 'Treatment B (N = xx)' to the bottom left (POSITION = BOTTOMLEFT) with text attributes (TEXTATTRS) of color = black, size = 7, weight = bold, and STYLE = NORMAL.

```
proc sgplot data=dsin noborder nowall dattrmap=myattrmap; 1
  format _1 positive.; 2
  hbar aedecod / response = _1 group = grdcat groupdisplay = stack 3
               attrid = myid dataskin = pressed seglabel
               seglabelfitpolicy = thin
               seglabelattrs = (size = 7 color = white weight = bold)
               name = "c1";
  hbar aedecod / response = _2 group = grdcat groupdisplay = stack 3
               attrid = myid dataskin = pressed seglabel 4
               seglabelfitpolicy = thin
               seglabelattrs = (size = 7 color = white weight = bold);
  xaxis grid gridattrs = (pattern = solid) display = (noticks) 5
        label = "Patients (%)" labelattrs = (weight = bold)
        values = (-70 to 70 by 10) valueattrs = (weight = bold);
  yaxis display = (noline nolabel noticks) discreteorder = data 6
        valueattrs = (weight = bold);
  keylegend "c1" / down = 3 noborder title = ''; 7
  inset "Treatment A (N = &pop1)" / position = bottomleft 8
        textattrs = (color = black size = 7 weight = bold style = normal);
  inset "Treatment B (N = &pop2)" / position = bottomright
        textattrs = (color = black size = 7 weight = bold style = normal);

run;
```

USE SGPLOT TO GENERATE GTL CODE

While the SG procedures such as SGPLOT are very powerful, they do not contain all the options available in GTL. You can, however, use the SGPLOT procedure to generate GTL code, which you can customize as required. To output the GTL code to a SAS program, use the TMPLOUT = FILENAME option on the procedure statement. From here, you can modify the template with your requirements and add a PROC SGRENDER to produce the graph (Schwartz, 2009).

```
proc sgplot data = datain tmpout = "<insert pathname>\GTL_butterfly.sas";
  <insert code from above>

run;
```

The resulting GTL PROC TEMPLATE code saved to the SAS program, GTL_butterfly.sas. When you review the SAS code, it illustrates the point that GTL is the foundation upon which the SG procedures are built.

The next section will outline the GTL code for a multi-cell butterfly graph. Due to requirements of placing the preferred terms down the middle of the graph, we did not use and modify the GTL code from the single-celled SGPLOT to produce this next graph.

MULTI-CELL BUTTERFLY GRAPH USING GRAPH TEMPLATE LANGUAGE

PROC TEMPLATE and SGRENDER procedures are used to create the overall layout of the multi-cell butterfly graph (see Figure 2) with adverse events displayed in the middle and percentage of patients for each toxicity grade by treatment in the left- and right-side plots.

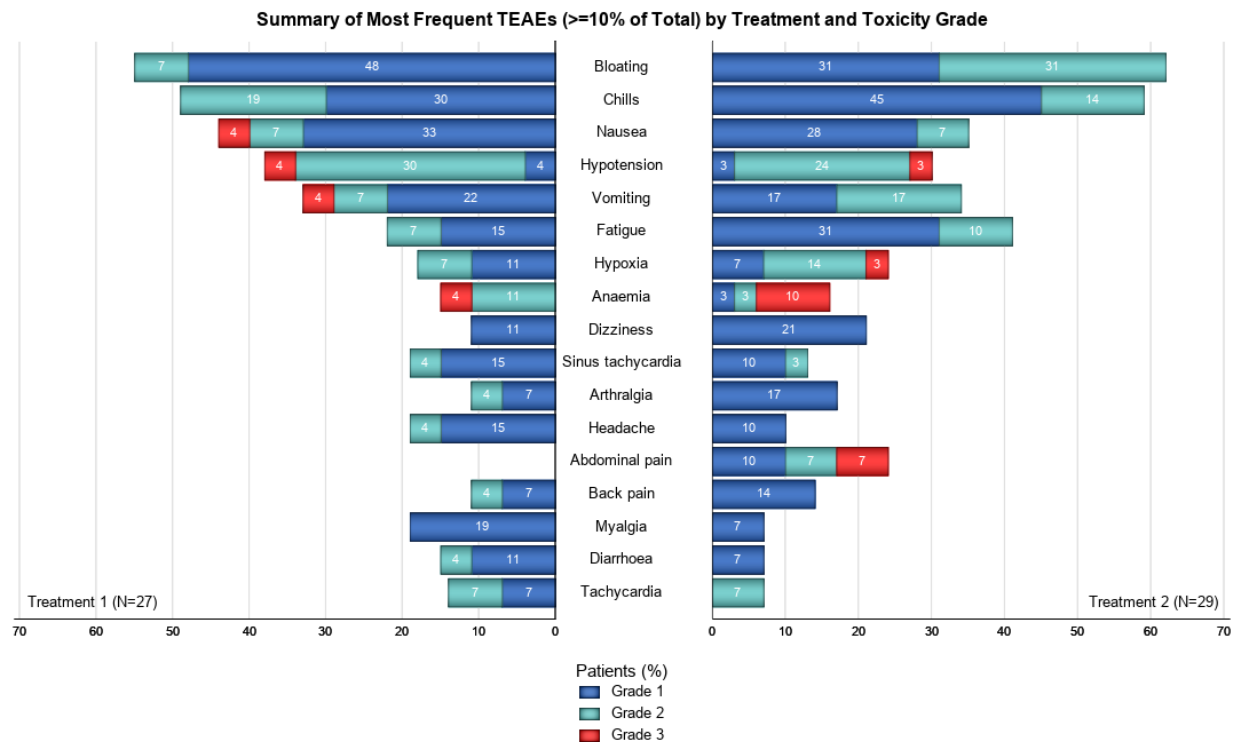


Figure 2. GTL Butterfly Multi-cell Graph of Most Frequent TEAEs by Treatment and Toxicity Grade

GTL graphics are generated by template definitions (PROC TEMPLATE) that control the graph format, appearance and specify the variable roles and attributes to represent in the graph display. The graphs can then be rendered by associating the templates with a data source.

- ❶ DEFINE STATGRAPH statement creates the graph template BUTTERFLY_TOXGR which is specified in the PROC SGRENDER statement after the PROC TEMPLATE code.
- ❷ The BEGINGRAPH statement defines the outermost container for a single GTL-layout-block and one or more GTL-global-statements.
- ❸ To display title and footnotes, use the ENTRYTITLE and ENTRYFOOTNOTE statements. The title position is defaulted to the center of the graph and the HALIGN=right option right-aligns the footnote. PROC SQL code above pulls the creation date of any data set you specify, using the SAS variable, CRDATE, from the data set in SASHELP.VTABLE. Here we used the creation date of the raw data set, DM2, to obtain our data extraction date for the footnote. When the source data has been updated, this will be reflected in the graph output.
- ❹ Using a discrete attribute map, the DISCRETEATTRMAP statement, assigns a different fill color (e.g., FILLATTRS = (color = CX0B4CB5)) to each toxicity grade.
- ❺ The discrete attribute variable statement DISCRETEATTRVAR creates a named association between the attribute map and an input data column such as toxicity grade (VAR = GRDCAT).
- ❻ One LAYOUT LATTICE statement defines the y-axis data range of the columns (ROWDATARANGE = union), number of columns (COLUMNS = 3), and the width of the individual columns (COLUMNWEIGHTS = (0.42 0.16 0.42)). We recommend keeping the first and last column to the same weight to preserve the scale.

Three LAYOUT OVERLAY statements pairing with END OVERLAY create blocks (left side, middle, and right side) which define the plot type and attributes (x- and y-axis, reversing the axes, lines, ticks) of each overlay. Within each of the LAYOUTs, a BARCHART statement is used to produce the horizontal bars.

- 7 The first BARCHART statement creates the left-hand side plot, with CATEGORY = AEDECOD for the x-axis variable, and RESPONSE = _1 for the y-axis.

In the options, GROUP = MYID_GRD creates a separate bar segment for the grouping variable, toxicity grade. MYID_GRD is the attribute variable name for toxicity grade and this variable name is specified in the discrete attribute map statement.

NAME = 'bar1' assigns a name to the BARCHART statement for reference in the DISCRETELEGEND statement.

Additional options, ORIENT = HORIZONTAL specifies the orientation of the y-axis and the bars, DATASKIN = PRESSED enhances the visual appearance of the bars, SEGMENTLABEL = TRUE specifies whether that a label will be displayed inside each bar segment, and SEGMENTLABELATTRS = (COLOR = WHITE) specifies the segment color.

Finally, the ENTRY statement specifies a line to text in the plot area. Here we display the text 'Treatment A (N = xx)' in the bottom left of the plot with the option AUTOALIGN = (BOTTOMLEFT).

- 8 For the middle 'plot' of the preferred terms that run vertically, the second LAYOUT OVERLAY block is used to NOT display the middle x-axis (XAXISOPTS = (DISPLAY = NONE) and to NOT display the walls and plot outline (WALLDISPLAY = NONE). On the y-axis, specify that tick values are to be displayed in a centered alignment (YAXISOPTS = (DISPLAY = (TICKVALUES) TICKVALUEHALIGN = CENTER)).

The second BARCHART statement creates the middle plot, with CATEGORY = AEDECOD for the x-axis variable, and RESPONSE=_1 for the y-axis. As explained above, the x-axis is suppressed.

- 9 The syntax for the right-hand plot is like the first LAYOUT OVERLAY block with the following exceptions:
 - remove display the y-axis values and reverse the values (YAXISOPTS = (DISPLAY = NONE REVERSE = TRUE)) as these are displayed using the middle plot.
 - the BARCHART statement, uses RESPONSE = _2 and the NAME = 'bar2' is not necessary for referencing in the legend because the first BARCHART statement took care of those details already.
 - the ENTRY statement specifies the text 'Treatment B (N = xx)' in the bottom right of the plot with the option AUTOALIGN = (BOTTOMRIGHT).
- 10 A SIDEBAR statement specifies the beginning of a side bar block and is used to produce the label of the x-axis (TITLE = "Patient (%)") and the legend for toxicity grades at the bottom of the graph (ALIGN=BOTTOM).

In the DISCRETELEGEND statement, 'bar1' is used to reference the values in first plot, ACROSS = 1 specifies the number of entries that are placed horizontally before the next row begins, HALIGN = CENTER positions the legend, and BORDER = FALSE removes the legend border. ITEMSIZE = (FILLHEIGHT = 10px FILLASPECTRATIO = GOLDEN) determines the height and the aspect ratio of the legend boxes. The 'golden' aspect ratio is defaulted to 1.618 (width = 1.618 * height) for both solid color and pattern fill swatches. This essentially means the filled boxes beside the toxicity grades are an appropriate size to show the color associated with each category.

```
proc template;
  define statgraph butterfly_toxgr; ❶
  begingraph; ❷
  entrytitle " Summary of Most Frequent TEAEs (>=10% of Total) by Treatment
and Toxicity Grade " ❸
  entryfootnote halign = right "Data Extraction Date: &dexdt.";
  discreteattrmap name = " _ATTRMAP_ "; ❹
    value "Grade 1" / fillattrs = (color = CX0B4CB5);
    value "Grade 2" / fillattrs = (color = bibg);
```

```

    value "Grade 3" / fillattrs = (color = red);
    value "Grade 4" / fillattrs = (color = purple);
enddiscreteattrmap;
discreteattrvar attrvar = MYID_GRD var = grdcatt attrmap = "__ATTRMAP__";5
layout lattice / rowdatarange = union6
    columnweights = (0.42 0.16 0.42)
    columns = 3;

*** Left side ***;
layout overlay / walldisplay = none7
    xaxisopts = (tickvalueattrs = (size = 7)
    display = (line tickvalues) griddisplay = on reverse = true
    linearopts = (viewmin = 0 viewmax = 70 tickvaluesequence = (start = 0
    end = 70 increment = 10)))
    yaxisopts = (display = none reverse = true);
    barchart category = aedecoded response = _1 / group = MYID_GRD
        name = 'bar1' orient = horizontal
        dataskin = pressed segmentlabel = true
        segmentlabelattrs = (color = white);
    entry "Treatment 1 (N = &pop1)" / autoalign = (bottomleft);
endlayout;

*** Middle ***;
layout overlay / xaxisopts = (display = none) walldisplay = none8
    yaxisopts = (display = (tickvalues) tickvaluehalign = center);
    barchart category = aedecoded response = _2 / orient = horizontal
        datatransparency = 1;
endlayout;

*** Right side ***;
layout overlay / walldisplay = none9
    xaxisopts = (tickvalueattrs = (size = 7)
    display = (line tickvalues) griddisplay = on
    linearopts = (viewmin = 0 viewmax = 70 tickvaluesequence = (start = 0
    end = 70 increment = 10)))
    yaxisopts = (display = none reverse = true);
    barchart category = aedecoded response = _2 /group = MYID_GRD
        orient = horizontal dataskin = pressed segmentlabel = true
        segmentlabelattrs = (color = white);
    entry "Treatment 2 (N = &pop2)" /autoalign = (bottomright);
endlayout;

*** Bottom Sidebar ***;
sidebar / align = bottom;10
    discretelegend 'bar1' / across = 1 halign = center title = "Patients
    (%)" itemsize = (fillheight = 10px fillaspectratio = golden) border =
    false;
endsidebar;
endlayout;
endgraph;
end;

```

The PROC SGRENDER associates the input data set (perc_tox3) with the template name (butterfly_toxgr) which is defined in the GTL DEFINE STATGRAPH statement to then create the graph.

```
proc sgrender data = perc_tox3 template = butterfly_toxgr;
run;
```

HIGH-LEVEL COMPARISON OF BUTTERFLY GRAPH SAS STATEMENTS: SGPLOT VS GTL

Table 1 shows a high-level comparison of SGPLOT and GTL butterfly plot SAS code that performs the same actions. For example, TITLE and FOOTNOTE statements are used in SGPLOT whereas ENTRYTITLE and ENTRYFOOTNOTE are used in GTL. For attribute maps, SGPLOT requires an additional DATA step that is referenced in the SGPLOT statement while in GTL two statements, DISCRETEATTRMAP and DISCRETEATTRVAR are required. Interestingly, SGPLOT takes three statements to create positive numbers for reversing the x-axis whereas GTL only uses an option in the LAYOUT OVERLAY statement: XAXISOPTS (REVERSE=TRUE).

Another difference that can be seen in the butterfly plot code is that GTL has quite a few more lines of code than SGPLOT due to the addition of the LAYOUT LATTICE and multiple LAYOUT OVERLAY statements. Even with more lines of code, GTL has more flexibility when it comes to modifying the different overlays within the graph. For example, moving the preferred terms down the middle of the graph simply entailed adding an additional LAYOUT OVERLAY statement and options.

SGPLOT		GTL	
Statement	Description	Statement	Description
TITLE	Displays a title	ENTRYTITLE	Displays a title
FOOTNOTE	Displays a footnote	ENTRYFOOTNOTE	Displays a footnote
DATA SGPLOT	Create attribute data set Specifies the SG data attribute map created in the additional data step	DISCRETEATTRMAP DISCRETEATTRVAR	Maps user-defined values to graphical properties Associate user-defined attribute map with input data column
DATA PICTURE FORMAT	Create negative input numbers for the left plot Creates a template for printing numbers Associates the picture format with variables	LAYOUT OVERLAY	Option: XAXISOPTS (REVERSE=TRUE) display the numbers with positive tick values
HBAR	Bar chart that summaries values of a categorical variable	BARCHART	Bar chart that summaries values of a categorical variable
XAXIS	Specifies options for the x-axis	LAYOUT OVERLAY	Option: XAXISOPTS Specifies options for the x-axis
YAXIS	Specifies options for the y-axis	LAYOUT OVERLAY	Option: YAXISOPTS Specifies options for the x-axis
INSET	Adds a text box inside the axes of a plot	ENTRY	Displays a line of text in the plot area
KEYLEGEND	Creates a legend for the plot	DISCRETELEGEND	Creates a legend with entries that refer to plots and/or groups

Table 1. High-level Comparison of Butterfly Graph SAS Statements: SGPLOT vs GTL

CONCLUSION

SGPLOT and GTL are efficient tools to produce essential clinical trial graphs such as butterfly graphs. We demonstrated that SGPLOT and GTL can produce very similar graphical output except for GTL excelling in the ability to produce sophisticated graphics with the flexibility to place text such preferred terms on any axis (left, right or middle axes). We have shown that from any SGPLOT code, you can produce the GTL code that you can then modify accordingly. Once you get a general understanding of the basic SGPLOT and GTL statements, you will be able to easily modify any graph and produce amazing looking graphs on your own.

REFERENCES

Robert N. Rodriguez and Warren F. Kuhfeld, 2012. "An Overview of ODS Statistical Graphics in SAS 9.4." SAS Institute Inc., Cary, NC. Accessed November 29, 2022.

Available at https://support.sas.com/rnd/app/ODSGraphics/papers/overview_odsgraphics_94.pdf

Shah Aakar and Tracy Sherman (2022). [A Beginner's Guide to Create Series Plots Using SGPLOT Procedure: From Basic to Amazing](#). Proceedings of the 2022 Pharmaceutical Industry SAS Users Group Conference. Page 2.

Schwartz, Susan. 2009. "Clinical Trial Reporting Using SAS/GRAPH® SG Procedures" Proceedings of the SAS Global Forum 2009 Conference.

Available at: <https://support.sas.com/resources/papers/proceedings09/174-2009.pdf>

ACKNOWLEDGMENTS

The authors would like to thank Eric Song, Ganesh Gopal, and Syamala Schoemperlen for their ongoing support and encouragement in conference participation.

RECOMMENDED READING

- *SAS® 9.4 ODS Graphics: Procedure Guide, Sixth Edition (SAS Institute, June 8, 2022)*
- *SAS® 9.4 Graph Template Language: Reference, Fifth Edition (SAS Institute, June 8, 2022)*

CONTACT INFORMATION

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

Tracy Sherman
Ephicity Consulting Group, Inc.
tracy.sherman@ephicity.com
www.ephicity.com

Aakar Shah
Acadia Pharmaceuticals Inc.
Aakar.Shah@acadia-pharm.com
www.acadia.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.

APPENDIX I DUMMY SOURCE DATA SET AND POST-PROCESSING

Due to lack of space, a portion of the dummy ADAE data that can be used to produce the butterfly figures is shown in the data step below.

USUBJID and AEDECOD variables are greater than eight characters long so adding an informat with a colon modifier in the INPUT statement (i.e. :\$200.) will prevent SAS from truncating these values to the first eight characters:

```
data adae;
infile datalines delimiter = ',';
input usubjid :$18. trta $ trtan aedecod :$200. aetoxgrn ;
datalines;
PHARM-ABC-100-0005, TRT A, 1, Bloating, 1
PHARM-ABC-100-0005, TRT A, 1, Chills, 1
PHARM-ABC-100-0005, TRT A, 1, Myalgia, 1
PHARM-ABC-100-0005, TRT A, 1, Sinus tachycardia, 2
PHARM-ABC-100-0005, TRT A, 1, Nausea, 3
PHARM-ABC-101-0005, TRT A, 1, Hypotension, 3
PHARM-ABC-101-0004, TRT A, 1, Bloating, 1
PHARM-ABC-101-0004, TRT A, 1, Chills, 2
PHARM-ABC-101-0004, TRT A, 1, Blood lactic acid increased, 1
PHARM-ABC-101-0004, TRT A, 1, Diarrhoea, 1
PHARM-ABC-101-0004, TRT A, 1, Hypoglycaemia, 1
PHARM-ABC-101-0004, TRT A, 1, Hypomagnesaemia, 1
PHARM-ABC-101-0004, TRT A, 1, Hypotension, 2
PHARM-ABC-101-0004, TRT A, 1, Hypoxia, 2
PHARM-ABC-101-0004, TRT A, 1, Nausea, 1
PHARM-ABC-101-0004, TRT A, 1, Pollakiuria, 1
PHARM-ABC-101-0004, TRT A, 1, Skin infection, 2
PHARM-ABC-101-0004, TRT A, 1, Thrombocytopenia, 2
PHARM-ABC-101-0005, TRT A, 1, Arthralgia, 1
PHARM-ABC-101-0005, TRT A, 1, Back pain, 1
PHARM-ABC-101-0005, TRT A, 1, Bloating, 1
PHARM-ABC-101-0005, TRT A, 1, Cough, 1
PHARM-ABC-101-0005, TRT A, 1, Dyspnoea exertional, 1
PHARM-ABC-101-0005, TRT A, 1, Fatigue, 1
PHARM-ABC-101-0005, TRT A, 1, Nausea, 1
PHARM-ABC-101-0005, TRT A, 1, Oropharyngeal pain, 1
PHARM-ABC-101-0005, TRT A, 1, Tachycardia, 1
PHARM-ABC-101-0006, TRT A, 1, Bloating, 2
PHARM-ABC-101-0006, TRT A, 1, Hypotension, 2
PHARM-ABC-101-0006, TRT B, 2, Hypoxia, 2
PHARM-ABC-101-0006, TRT B, 2, Nausea, 1
PHARM-ABC-101-0006, TRT B, 2, Pleural effusion, 2
PHARM-ABC-101-0006, TRT B, 2, Tachycardia, 2
PHARM-ABC-101-0006, TRT B, 2, Thrombocytopenia, 1
PHARM-ABC-102-0004, TRT B, 2, Abdominal pain upper, 3
PHARM-ABC-102-0004, TRT B, 2, Cough, 1
PHARM-ABC-102-0004, TRT B, 2, Hypoxia, 1
PHARM-ABC-102-0004, TRT B, 2, Nausea, 1
PHARM-ABC-102-0004, TRT B, 2, Pulmonary embolism, 2
PHARM-ABC-102-0004, TRT B, 2, Skin laceration, 1
PHARM-ABC-102-0005, TRT B, 2, Abdominal discomfort, 1
PHARM-ABC-102-0005, TRT B, 2, Adrenal insufficiency, 2
PHARM-ABC-102-0005, TRT B, 2, Arthralgia, 2
PHARM-ABC-102-0005, TRT B, 2, Bloating, 1
```

```

PHARM-ABC-102-0005,      TRT B,      2,      Chest discomfort, 1
PHARM-ABC-102-0005,      TRT B,      2,      Chills,          2
PHARM-ABC-102-0005,      TRT B,      2,      Decreased appetite, 1
PHARM-ABC-102-0005,      TRT B,      2,      Fatigue,         2
PHARM-ABC-102-0005,      TRT B,      2,      Hepatitis,       1
PHARM-ABC-102-0005,      TRT B,      2,      Hypoxia,         1
PHARM-ABC-102-0005,      TRT B,      2,      Myalgia,         1
PHARM-ABC-102-0005,      TRT B,      2,      Pain in extremity, 1
PHARM-ABC-102-0005,      TRT B,      2,      Skin mass,       1
run;

** Add Total Treatment group **;
data adae;
  set adae;
  output;
  trtan = 99;
  trta = 'Total';
  output;
run;

** POPx macro variables for denominators (Treatment A and B) **;
proc sql;
  create table pop as
  select trtan, count(distinct usubjid) as cnt
  from adae
  group by trtan;
quit;

data _null_;
  set pop;
  call symput ("pop"||strip(put(trtan,best.)),strip(put(cnt,best.)));
run;
%put pop1 = &pop1 pop2 = &pop2 pop99 = &pop99;

** Select the maximum toxicity grade of event by treatment, subject,
preferred term **;
proc sort data = adae;
  by trtan usubjid aeecod descending aetoxgrn;
  where aeecod^ = '';
run;

data ae_max;
  set adae;
  by trtan usubjid aeecod descending aetoxgrn;
  if first.aeecod;
run;

** Count of subjects by preferred term, toxicity, treatment **;
proc freq data = ae_max noprint;
  tables aeecod*aetoxgrn*trtan / out = _freq1 (drop = percent);
run;

** Count of subjects by preferred term, treatment **;
proc freq data = ae_max noprint;
  tables aeecod*trtan / out = _freq2 (drop = percent);
run;

```

```

** Transpose by preferred term **;
proc transpose data = _freq2 out = _freq3 (drop = _name_ _label_);
  by aedecod;
  id trtan;
  var count;
run;

options symbolgen mprint;

%macro perc;
%** Percent of Subjects by preferred term and toxicity grade **;
data perc_tox0;
  set _freq1;
  %do l = 1 %to 2;
    if count > . then do;
      if trtan = &l then perc = input(put((count / &&pop&l.)*100,5.),8.);
    end;
    else perc = 0;
    if count = . then count = 0;
  %end;
  if count > . then do;
    if trtan = 99 then perc = input(put((count / &pop99.)*100,5.),8.);
  end;
  else perc = 0;
  if count = . then count = 0;
run;

proc transpose data = perc_tox0 out = perc (drop = _name_);
  by aedecod aetoxgrn;
  id trtan;
  var perc;
run;

%** Overall percent of subjects by preferred term **;
data perc_no_tox;
  set _freq3;
  if _99 > . then do;
    perc99 = input(put((_99 / &pop99.)*100,5.),8.);
  end;
  else perc99 = 0;
  if _99 = . then _99 = 0;
  keep aedecod perc99;
run;

data perc_all;
  merge perc perc_no_tox;
  by aedecod;
  label
  aedecod = 'Preferred Term'
  _1 = 'Percent Subjects Trt A'
  _2 = 'Percent Subjects Trt B'
  perc99 = 'Percent Subjects Total'
  ;
run;
%mend perc;
%perc;

```

```

** Select TEAE >= 10 percent of total subjects **;
data input_tox;
  set perc_all;
  if perc99 >= 10;
run;

proc sort data = input_tox;
  by descending perc99;
run;

** Add GRDCAT for grouping variable **;
data dsin;
  set input_tox;
  grdcat = 'Grade '||strip(put(aetoxgrn,8.));
  aetoxgrn = put(aetoxgrn,best.);
  _1 = -_1; ** negative numbers for left-hand plot (Trt A) **;
run;

** Orders AEs in plot based on overall percentage **;
proc sort data = dsin;
  by descending perc99;
run;

```

APPENDIX II SGPLOT SAS CODE

```

ods escapechar = '^';

ods _all_ close;
options ls = max;

proc format;
  picture positive
    low -< 0 = "0000"
    0 <- high = "0000";
run;

*This DATA step creates the discrete attribute map data set MYATTRMAP.
The ID values for the attribute map are MYID.*;
data myattrmap;
input ID $ 1-4 value $ 6-12 fillcolor $;
datalines;
myid Grade 1 CX0B4CB5
myid Grade 2 bibg
myid Grade 3 red
myid Grade 4 purple
;
run;

ods listing gpath = "&output.";
ods graphics / reset imagename = "f_butterfly_sgplot_3_pharmasug"
  outputfmt = png width = 10in;

title "Summary of Most Frequent TEAEs (>=10% of Total) by Treatment and
  Toxicity Grade";
footnote j = right "Data Extraction Date: &dexdt.";

```

```

proc sgplot data = dsin noborder nowall dattrmap = myattrmap;
  format _1 positive.;
  hbar aedecod / response = _1 group = grdcats groupdisplay = stack
    attrid = myid dataskin = pressed seglabel
    seglabelfitpolicy = thin
    seglabelattrs = (size = 7 color = white weight = bold)
    name = "c1";
  hbar aedecod / response = _2 group = grdcats groupdisplay = stack
    attrid = myid dataskin = pressed seglabel
    seglabelfitpolicy = thin
    seglabelattrs = (size = 7 color = white weight = bold);
  xaxis grid gridattrs = (pattern = solid) valueattrs = (weight = bold)
    labelattrs = (weight = bold) display=(noticks)
    label = "Patients (%)" values = (-70 to 70 by 10);
  yaxis display = (noline nolabel noticks) discreteorder = data
    valueattrs = (weight = bold);
  keylegend "c1" / down = 3 noborder title = '';
  inset "Treatment A (N = &pop1)" / position = bottomleft
    textattrs = (color = black size = 7 weight = bold
    style = normal);
  inset "Treatment B (N = &pop2)" / position = bottomright
    textattrs = (color = black size = 7 weight = bold
    style = normal);

run;

ods html close;
ods listing;

```

APPENDIX III GTL SAS CODE

```

ods escapechar = '^';

ods _all_ close;
options ls = max;

proc template;
  define statgraph butterfly_toxgr;
    begingraph;
    entrytitle "Summary of Most Frequent TEAEs (>=10% of Total) by
      Treatment and Toxicity Grade";
    entryfootnote halign = right "Data Extraction Date: &dexdt.";

    discreteattrmap name = "__ATTRMAP__";
    value "Grade 1" / fillattrs = (color = CX0B4CB5);
    value "Grade 2" / fillattrs = (color = bibg);
    value "Grade 3" / fillattrs = (color = red);
    value "Grade 4" / fillattrs = (color = purple);
    enddiscreteattrmap;

    discreteattrvar attrvar = MYID_GRD var = grdcats
      attrmap = "__ATTRMAP__";

    layout lattice / rowdatarange = union columnweights = (0.45 0.12 0.43)
      columns = 3;
  end;

```

```

*** Left side ***;
  layout overlay / walldisplay = none
  xaxisopts = (tickvalueattrs = (size = 7)
  display = (line tickvalues) griddisplay = on
  reverse = true linearopts = (viewmin = 0 viewmax = 70
  tickvaluesequence = (start = 0 end = 70 increment = 10))
  yaxisopts = (display = none reverse = true);
  barchart category = aeDecod response = _1 /
    group = MYID_GRD name = 'bar1'
    orient = horizontal dataskin = pressed
    segmentlabel = true
    segmentlabelattrs = (color = white);
  entry "Treatment 1 (N = &pop1)" /autoalign = (bottomleft);
endlayout;

*** Middle ***;
  layout overlay / xaxisopts = (display = none) walldisplay = none
  yaxisopts = (display = (tickvalues)
  tickvaluehalign = center);
  barchart category = aeDecod response = _2 /
    orient = horizontal datatransparency = 1;
endlayout;

*** Right side ***;
  layout overlay / walldisplay = none
  xaxisopts = (tickvalueattrs = (size = 7)
  display = (line tickvalues) griddisplay = on
  linearopts = (viewmin = 0 viewmax = 70
  tickvaluesequence = (start = 0 end = 70 increment = 10))
  yaxisopts = (display = none reverse = true);
  barchart category = aeDecod response = _2 /
    group = MYID_GRD orient = horizontal
    dataskin = pressed segmentlabel = true
    segmentlabelattrs = (color = white);
  entry "Treatment 2 (N = &pop2)" /
    autoalign = (bottomright);
endlayout;

*** Bottom Sidebar ***;
  sidebar / align = bottom;
  discretelegend 'bar1' / across = 1 halign = center
  title = "Patients (%)"
  itemsize = (fillheight = 10px
  fillaspectratio = golden) border = false;
  endsidebar;
endlayout;
endgraph;
end;

run;

ods listing gpath = "&output." ;

ods graphics on / reset imagename = "f_butterfly_gtl_pharmasug"
  outputfmt = png width = 1100px height = 700px noborder;

```

```
** Associate the data with the template using the SGRENDER procedure  
   to create the graph**;
```

```
proc sgrender data = dsin template = butterfly_toxgr;  
run;
```

```
ods html close;  
ods listing;
```