PharmaSUG 2025 - Paper DV-188

Beyond Basic SG Procedures: Enhancing Visualizations in SAS Graphics

Fengying Miao, Incyte Corporation
Vicky Yuan, Incyte Corporation

ABSTRACT

SAS statistical graphics (SG) procedures provide powerful tools for generating high-quality graphs. However, users often encounter limitations when customizing plots beyond the standard options. This paper introduces **SG annotations**, **SG discrete attribute maps, AXIS tables and SGPANEL**, advanced techniques that extends the capabilities of SG procedures by allowing users to precisely control graphical elements, add custom labels, lines, shapes, and other enhancements. Through practical examples, we demonstrate how these techniques can be applied to various plot types, demonstrating techniques to refine presentation graphics and tailor statistical displays to specific analytical needs. By leveraging **SG annotations**, **SG discrete attribute maps, AXIS tables and SGPANEL**, users can create a graphic with more information, visually compelling and highly customization.

The SAS code used in this paper is using SAS@ 9.4 SAS Enterprise environment.

INTRODUCTION

In the realm of data analysis, effective visualization plays a crucial role in conveying complex information clearly and intuitively. The SAS SG procedure is a powerful tool that enables analysts to create a wide array of visualizations, from basic plots to intricate graphical representations. However, the true potential of these visualizations can be significantly enhanced using SG annotations**,** SG discrete attribute maps, AXIS tables and SGPANEL.

SG ANNOTATIONS

The SG annotations offer a versatile way to add custom elements to plots, such as text labels, lines and markers. This capability allows for the inclusion of supplementary information directly on the graph. Here are key steps for utilizing SG annotations.

1) Create a SG annotation dataset.
    - The variable names within the SG annotation dataset must utilize reserved keywords as defined by SAS.
    - Each observation in the dataset corresponds to a command that specifies how to draw a particular annotation.
2) Include annotation dataset in SG procedures.
    - Annotations can be employed in various procedures, such as SGPLOT, SGPANEL, and SGSCATTER.

      proc sgplot data=xxx sganno=<SG annotation dataset name>;

ANNOTATION DATASET VARIABLES

In each observation, the SG procedures specifically search for variables with predefined names. Here are essential variables that have designated names.

- Function: This variable defines the action performed by the observation with the annotation dataset. See all available functions at Table 1.

FUNCTION	**Function Descriptions**
ARROW	Draws an arrow from (x1, y1) to (x2, y2)
IMAGE	Draws an image at (x1, y1) location from an image file.
LINE	Draws a line from (x1, y1) to (x2, y2)

FUNCTION	Function Descriptions
OVAL	Draws an oval or circle centered at (x1, y1).
POLYGON	Specifies the beginning point (x1, y1) of a polygon.
POLYLINE	Specifies the beginning point (x1, y1) of a polyline.
POLYCONT	Continues adding points (x1, y1) to a polygon or polyline.
RECTANGLE	Draws a rectangle or square annotation centered at (x1, y1).
TEXT	Places text in the graph output starting at (x1, y1).
TEXTCONT	Continues a text string from the end of the previous string.

Table 1. SG Annotation Functions

- X1, Y1, X2, Y2: These variables specify the coordinates for drawing the annotation. Specially, X1 and Y1 denote the starting point, while X2 and Y2 indicate the endpoint. Certain functions, such as TEXT, require only a single set of coordinates.
- Label: This variable contains the text that will be displayed as part of annotation.
- Attribute Variables: These variables define the characteristics of the graphical element, including attributes such as TEXTCOLOR, TEXTSIZE, LINECOLOR and others.

EXAMPLE 1: CREATING TEXT LABEL AND LINE IN PROC SGPLOT

This example demonstrates how to incorporate survival data into a Kaplan-Meier pot, featuring a water mark with the text 'Preliminary'.

Step 1: Create a dataset TSURV (Refer to Dataset 1) that includes the required survival data using Proc Lifetest and Proc Transpose (See Program 1).

```
ods listing close;
ods graphics on;
ods output survivalplot=splot;
proc lifetest data=adeff method=km conftype=loglog
              plots=survival(atrisk=0 to 42 by 6)
              outsurv=surv timelist=12, 24, 36 reduceout;
     time avalm*cnsr(1);
     strata trt01pn/test=logrank;
run;
ods graphics off;
ods listing;

data surv;
  set surv;
  length ospct $50;
  ospct=strip(put(round(survival*100, 1.0), best.))||'%';
run;

proc sort; by timelist trt01pn; run;

proc transpose data=surv out=tsurv(drop=_name_) prefix=c;
  by timelist;
  var ospct;
  id trt01pn;
run;

data tsurv(drop=timelist);
  set tsurv end=last;
  row=_n_;
  length variable $200;
  variable=strip(put(timelist, survfmt));
  output;
  if last then do;
     call missing(variable);
```

```
      row=0.1;
     c1='active';
     c2='Best Available Therapy';
      output;
      row=0.2;
      c1="(n="||strip(put(&ntrt1., best.))||")";
      c2="(n="||strip(put(&ntrt2., best.))||")";
      output;
   end;
run;
```

Program 1. Code for creating survival dataset

variable	c1	c2
	Active	Best Available Therapy
	(n=142)	(n=71)
% 1-year Survival	96%	94%
% 2-year Survival	86%	80%
% 3-year Survival	79%	58%

Dataset 2. Survival Data named TSURV

Step 2: Utilize the dataset from Step 1 to generate an SG annotation dataset called ANNO (Refer to Dataset 2, only a portion of dataset is displayed here).

```
data anno(keep=label textcolor linecolor drawspace function x1 y1 x2 y2 width textsize
                rotate textweight transparency anchor);
   set tsurv end=last;
   length label $200 textcolor linecolor $10;
   width=100;
    ...
   if row=1 then do;
      label=variable; textsize=8; textcolor='black'; linecolor=' '; drawspace='datavalue';
      function='text'; x1=4; y1=0.27; x2=.; y2=.; output;

      label=c1; textsize=8; textcolor='black'; drawspace='datavalue'; function='text';
      x1=11; y1=0.27; x2=.; y2=.; output;

      label=c2; textsize=8; textcolor='black'; drawspace='datavalue'; function='text';
      x1=17; y1=0.27; x2=.; y2=.; output;

      label=''; textsize=.; textcolor=''; linecolor='black'; drawspace='datavalue';
      function='line'; x1=1; y1=0.23; x2=19; y2=0.23; output;
   end;
   ...
   if last then do;
      anchor='left'; label='Preliminary'; textcolor='gray'; drawspace='datavalue'; x1=0;
      y1=0; x2=.; y2=.;function='text'; rotate=30; textsize=80; textweight='bold';
      transparency=0.8; output;
   end;
run;
```

Program 2. Code for creating annotation dataset

label	textcolor	linecolor	width	textsize	drawspace	function	x1	y1	x2	y2	anchor	rotate	textweight	transparency
% 1-year Survival	black		100	8	datavalue	text	4	0.27						
96%	black		100	8	datavalue	text	11	0.27						
94%	black		100	8	datavalue	text	17	0.27						
		black	100		datavalue	line	1	0.23	19	0.23				
Preliminary	gray		100	80	datavalue	text	0	0			left	30	bold	0.8

Dataset 2. Annotation dataset called ANNO

Step 3: Employ the SGANNO option in PROC SGPLOT to produce the final Kaplan-Meier figure (Refer to Figure 1).

```
proc sgplot data=splot noborder sganno=anno;
    styleattrs datacolors=(gray black) datacontrastcolors=(black)
               datalinepatterns=(solid shortdash);
    format stratumnum ztrtfmt.;
    step x=time y=survival / group=stratumnum name='s';
    scatter x=time y=censored / markerattrs=(symbol=plus) name='c';
    scatter x=time y=censored / markerattrs=(symbol=plus) group=stratumnum;
    keylegend 'c' / location=inside position=bottomright noborder
                    valueattrs=(weight=bold size=9) ;
    keylegend 's' / linelength=20 location=inside noborder across=1
                    position=topright valueattrs=(weight=bold size=9 color=black);
    yaxis values=(0 to 1 by 0.1) label='Survival Probability'
          valueattrs=(weight=bold size=9) ;
    xaxis values=(0 to 42 by 6) valueshint min=0 max=42 label='Time (Months)'
         valueattrs=(weight=bold size=9) offsetmin=0.015;
run;
```

Program 3. Code for employing annotation dataset in PROC SGPLOT

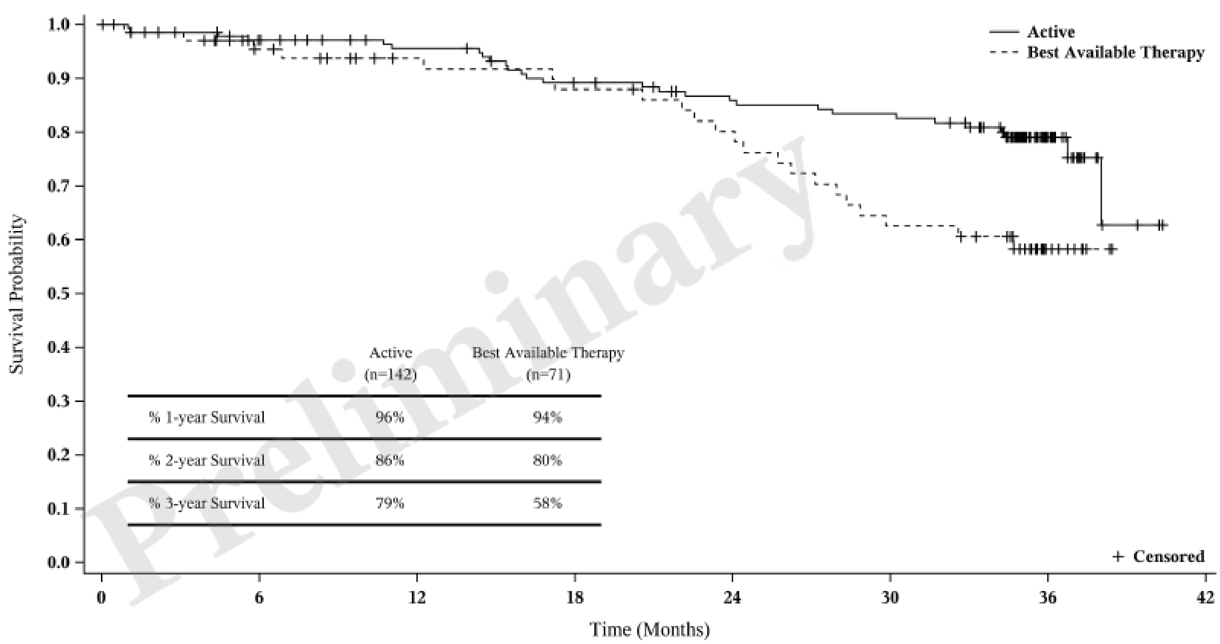

Figure 1. Overall Survival – KM by Treatment Group

EXAMPLE 2: CREATING TEXT LABEL IN PROC SGPANEL

This example shows how to include text annotations in SGPANEL. The data area is not available for SGPLANEL or SGSCATTER, so we use DRAWSPACE='graphpercent' in this context. We aim to display the number of participants for each treatment group, along with a text label for the reference line and x-axis label. The participant counts are stored in two macro variables &NTRT1 and &NTRT2. Following approach from Example 1, we create an annotation dataset ANNO2 using codes as specified in Program 3.

```
data anno2;
  length label $200 textcolor $10;
  anchor='left';
  width=100;
  textfont='Courier';
  drawspace='graphpercent';
  function='text';
  textcolor='black';
  label='Active'; textsize=9; x1=18; y1=68; output;
  label="N=&ntrt1."; textsize=9; x1=18; y1=65; output;
  label='Placebo'; textsize=9; x1=62; y1=68; output;
  label="N=&ntrt2."; textsize=9; x1=62; y1=65; output;
  label="35% Reduction"; textsize=8; x1=65; y1=28; output;
run;
```

Program 4. Code for creating annotation dataset for use in SGPANEL

In this step, we utilize the annotation dataset in PROC SGPANEL to generate Figure 2. The PANELBY option allows us to specify the number of columns or rows in the panel using COLUMNS or ROWS. The REFLINE statement is employed to create either a horizontal or vertical reference line.

```
proc sgpanel data=aseff_fin SGANNO=ANNO;
  panelby trt01pn/columns=2 novarname noborder noheaderborder spacing=4 noheader;
  vbar order/response=pc barwidth=0.3 fillattrs=(color=lightgray) transparency=0.4;
  colaxis display=(noticks novalues noline) labelpos=center;
  rowaxis values=(-80 to 70 by 10) label="Percent Change From Baseline"
          labelattrs=(Weight=Bold) valueattrs=(Weight=Bold);
  refline -35/lineattrs=(pattern=shortdash);
run;
```

Program 5. Code of employing annotation dataset in PROC SGPANEL

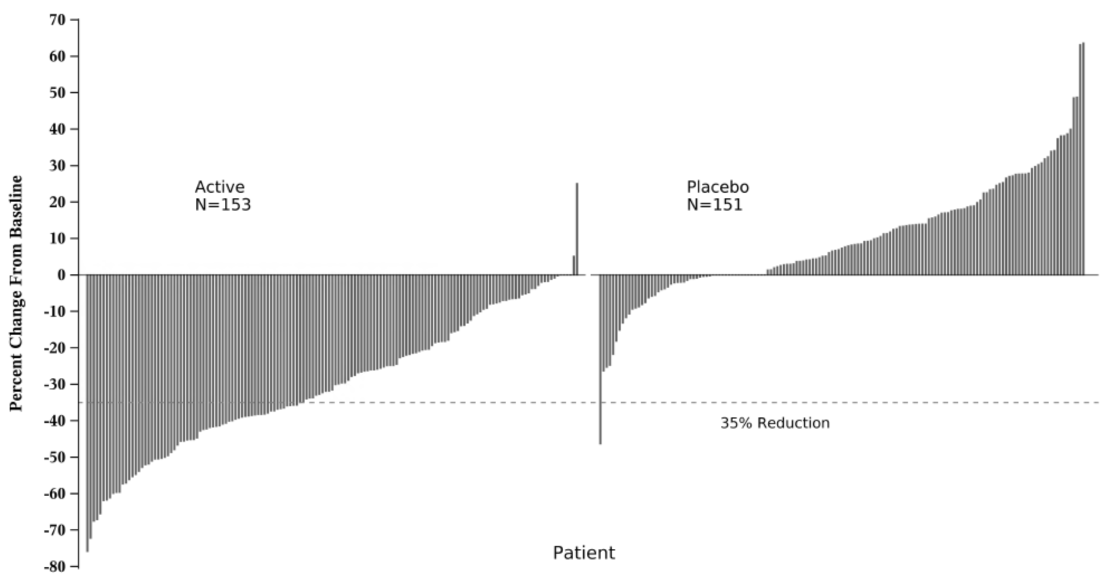

Figure 2. Percentage Change from Baseline at Week x

SG DISCRETE ATTRIBUTE MAPS

The SG discrete attribute maps enable the customization of plot aesthetics based on categorical variables. By mapping attributes such as color, size or line type to discrete values, analysts can create visually distinct representations of different groups within the data. This technique not only enhances the visual appeal of the plots but also aids in the differentiations of categories, making it easier for viewers to discern relationships and comparisons among groups. Here are the main steps for discrete attribute mapping.

1) Create a Discrete Attribute Map dataset.

   - Begin by developing a discrete attribute map dataset that utilizes reserved keywords for its variable names.
   - Each observation within the dataset specifies the attributes corresponding to a particular group value.

2) Incorporate the dataset into the SG procedure.

   - This discrete attribute map dataset can be included in various SG procedures, such as SGPLOT, SGPANEL, and SGSCATTER. For example, you can use the following syntax:

     proc sgplot data=xxx DATTRMAP=<discrete attribute map dataset name>;

DISCRETE ATTRIBUTE MAP DATASET VARIABLES

The following key variables are fundamental to the structure of a discrete attribute map dataset, each serving a specific purpose in defining graph properties and group attributes.

- **ID (Attribute Map Identifier):**

  This is a required variable that holds the name of the attribute map, which will be referenced using the ATTRID option in the plot statement. A discrete attribute map dataset can include multiple attribute maps, allowing for more than one ID.

- **VALUE (Group Value):**

  This is also a required variable that contains the group values to which specific graph properties will be assigned. The values in this variable must be valid according to the data group specified in the GROUP option of the plot statement.

- **Attribute Variables:**

  These variables define the attributes that you wish to assign to the observations in the discrete dataset. Example of attribute variables include LINECOLOR, MARKERCOLOR, LINEPATTERN, TEXTCOLOR, TEXTWEIGHT, FILLCOLOR, and FILLTRANSPARENCY.

EXAMPLE 3: USING DISCRETE ATTRIBUTE MAP TO ASSIGN ATTRIBUTES

By default, the graphical attributes for each group of values are derived from the style elements GraphData1-GraphDataN in the order of the data. The discrete attribute map allows for the assignment of identical graphical properties to specific values, independent of their data order.

In this example, we will demonstrate how to assign blue to treatment group A and red to treatment group B. To begin, we create a discrete attribute map dataset named MYATTR2(Refer to Dataset 3).

id	value	linecolor	fillcolor
mycolor	TREATMENT GROUP A	blue	blue
mycolor	TREATMENT GROUP B	red	red

Dataset 3. Discrete Attribute Dataset called MYATTR2

Then we utilize the discrete attribute map dataset MYATTR2 in PROC SGPLOT to generate Figure 3.

```
proc sgplot data=fin noborder dattrmap=myattr2;
   format trtgrpn ztrtfmt. agegr1n zage.;
   vbar agegr1n/ response=mean_weight group=trtgrpn groupdisplay=cluster attrid=mycolor
                 clusterwidth=0.7 barwidth=0.9 outlineattrs=(thickness=2)
                 transparency=0.2;
   ...
run;
```

Program 6. Code of employing discrete attribute map dataset in PROC SGPLOT

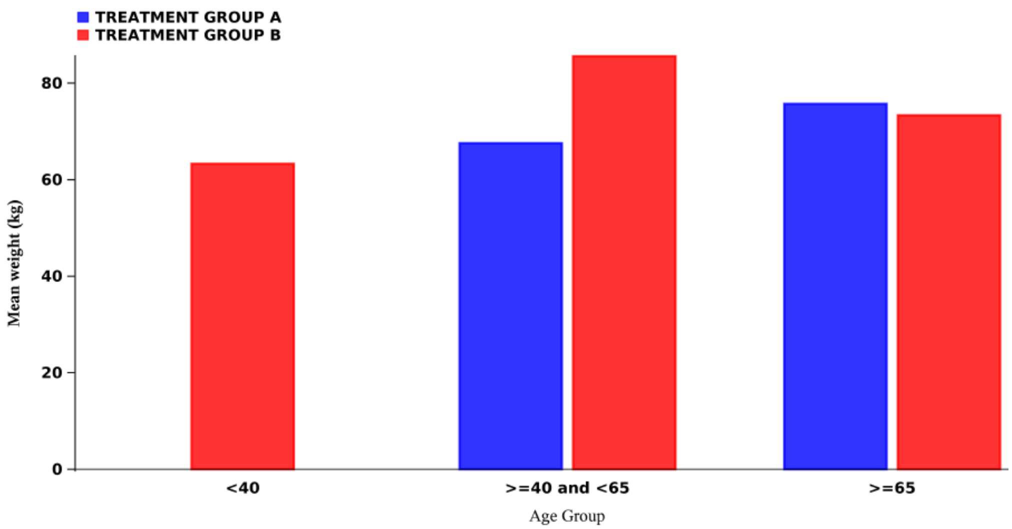

Figure 3. Mean Weight by Treatment Group and Age Group

Using discrete attribute maps allow for consistent application of the same attributes across different groups, enhancing comparability and facilitating more effective data analysis and interpretation.

SG AXIS TABLE

The XAXISTABLE and YAXISTABLE statements are utilized to generate axis tables that present data values at designated locations along the respective axes. Below are the options available for use with these axis table statements:

- The LOCATION=OUTSIDE|INSIDE option specifies whether the axis table is placed outside or inside the axis are.
- The Position= option specifies the position of the axis table.
  - X axis table: Valid values are BOTTOM (default) and TOP.
  - Y axis table: Valid values are LEFT (default) and RIGHT.
- The LABEL= option can be sued to provide a text string for table label.
- The CLASS= option can create a separate axis table for each unique value of specified variable.
- The CLASSDISPLAY=STACK|CLUSTER option specifies how the class values are displayed.

The SGPLOT procedure can contain multiple axis table statements. The only required argument is a list of one or more variables to be displayed. Here is the axis table syntax.

- xaxistable variable <…variable-n> / <options>;
- yaxistable variable <…variable-n> / <options>;

EXAMPLE 4: USING XAXISTABLE

This example illustrates the attainment of a specific level of reduction in individual symptom scores. The number of participants in each treatment group, along with baseline score data, is presented in the X axis table. We use below codes with xaxistable to generate Figure 4.

```
proc sgplot data=fin noborder;
   format trt01pn ztrtfmt. symptom zsfmt.;
   styleattrs datacolors=(black white) datacontrastcolors=(black);
   vbar symptom/response=pct group=trt01pn groupdisplay=cluster clusterwidth=0.7
             barwidth=0.9 outlineattrs=(thickness=2) transparency=0.2;
   xaxis display=(noticks noline nolabel) valueattrs=(weight=bold Family=Courier size=9);
   yaxis label='Proportion of Participants' labelattrs=(weight=bold size=10)
             valueattrs=(weight=bold Family=Courier size=9) ;
   keylegend/position=topleft across=1 noborder title='' valueattrs=(weight=bold
           Family=Courier size=9) autoitemsize;
   xaxistable nx meanx medianx/class=trt01pn classdisplay=cluster
                            valueattrs=(weight=bold Family=Courier size=9)
                            labelattrs=(weight=bold Family=Courier size=9)
                            location=inside;
run;
```

Program 7. Code of employing xaxistable in PROC SGPLOT

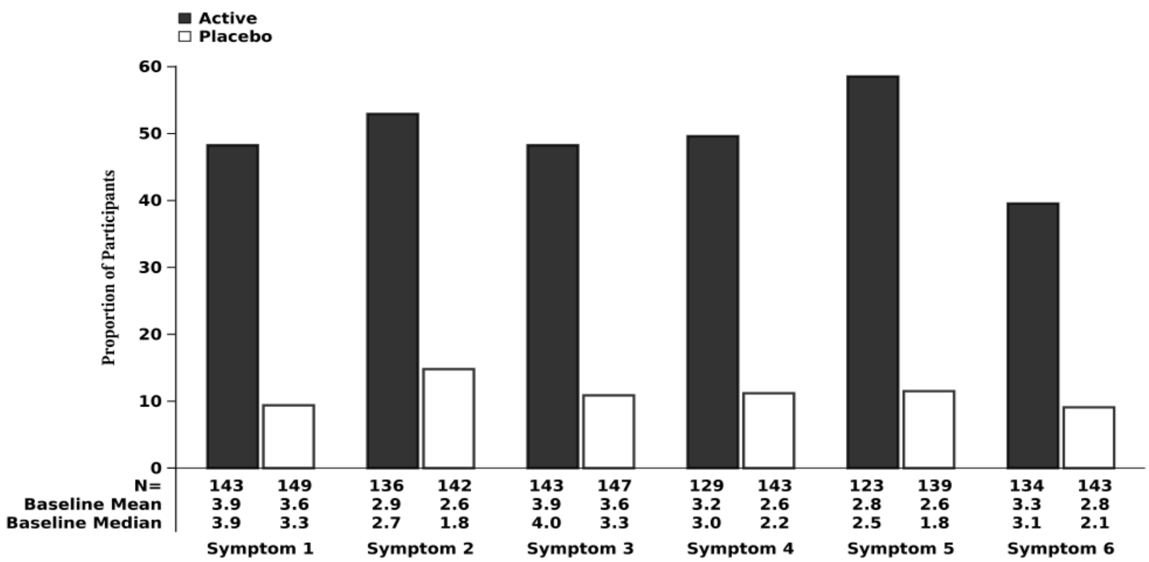

Figure 4. Proportion of Participants achieving >=50% reduction in individual symptom score

Using XAXISTABLE in this example allows for improved clarity in graphical presentations by enabling the display of detailed data alongside visual elements, facilitating better comparisons within the plotted results.

PROC SGPANEL

The SGPANEL procedure simplifies the creation of complex, multi-panel graphics that previously required extensive effort. It introduces built-in paneling capabilities, eliminating the need for cumbersome workarounds. While the concept of paneling has been explored in numerous SAS Global Forum papers—often implemented using PROC GREPLAY, SGPANEL streamlines the process, making it easier to generate multiple, related graphs without the associated challenges.

The PANELBY statement in SAS is used within PROC SGPANEL to create multi-panel displays of data based on one or more classification variables. It determines how the data is split into separate panels within a single graph layout.

Key Features of PANELBY:

- Splits data into separate panels based on the values of one or more categorical variables.
- Allows control over layout options such as number of columns, rows, and spacing.
- Supports options for uniform axis scaling across panels.

- Simplifies the visualization of grouped data without requiring PROC GREPLAY.

For example, to display a figure by treatment groups shown on the same page (See Figure 5), we need to create multi panels layout to present the data effectively. The basic syntax is as follows:

```
proc sgpanel data=final noautolegend;
    panelby tmt / columns=1 sort=data uniscale=column novarname;
    rowaxis label="Measured Value" min=40 max=80;
    scatter X=avisitn Y=aval/group=SUBJID name='A' markerattrs=(symbol=circlefilled);
    series X=avisitn Y=aval/ group=SUBJID break curvelabel name="grouping"
                         markerattrs=(symbol=circlefilled size=11);
    colaxis values= (1 to 25 BY 1) label="Visit" display=all valueshint
            fitpolicy=rotate;
    discretelegend 'A'/TITLE="Subjects" exclude=("");
    format avisitn emafmt.;
run;
```

Program 8. Code of PROC SGPANEL

Explanation:

- panelby tmt: - Creates separate panels for each unique value in tmt.
- / columns = 1: Arranges panels into 1x1 grid.
- scatter x=avisitn y=aval: Plots scatter plots in each panel.
- series x=avisitn y=aval: Plots line plots in each panel.

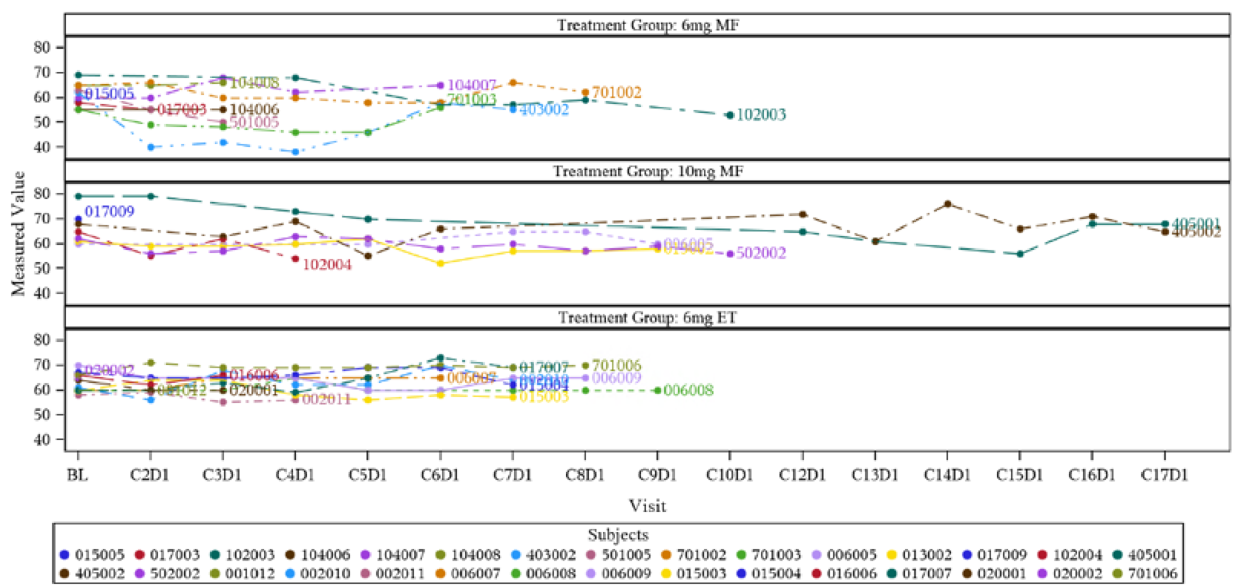

Figure 5. Spaghetti Plot by Treatment Group

As you can see this functionality is particularly valuable in clinical research, where visualizing treatment effects over time is crucial. The ability to maintain uniform axis scaling, control layout configurations, and integrate multiple plot types within the same panel makes **SGPANEL** an essential tool for biostatisticians and data analysts.

CONCLUSION

This paper explores the integration of SG annotations, SG discrete attribute maps, AXIS tables and PROC SGPANEL within PROC SG procedures to enhance visualizations. By utilizing these advanced features, analysts can produce graphical representations of data that are more informative, engaging, and insightful, facilitating improved decision-make and clearer communication of results. Through a series of practical examples and case studies, we aim to illustrate the effectiveness of these enhancements in various analytical scenarios.

REFERENCES

Sanjay Matange. 2014."Annotate your SGPLOT Graphs"

SAS Help Center "SG Annotation". Accessed March 20, 2025. SAS Help Center: SG Annotation

SAS Help Center "SG Discrete Attribute Maps". Accessed March 20, 2025. SAS Help Center: Using Discrete Attribute Maps

SAS Help Center "XAXISTABLE". Accessed March 20, 2025. SAS Help Center: Syntax: PROC SGPLOT XAXISTABLE Statement

SAS Help Center "SGPANEL Procedure". Accessed March 20, 2025. SAS Help Center: SGPANEL Procedure

CONTACT INFORMATION

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

Fengying Miao
fmiao@incyte.com
Incyte Corporation
1815 Augustine-cutoff, Wilmington, DE 19801

Vicky Yuan
vyuan@incyte.com
Incyte Corporation
1815 Augustine-cutoff, Wilmington, DE 19801