

Paper 246-2008

Raising the Bar in Generating Bar Charts and More. A Push-of-the-Button Approach in SAS®

Rita Tsang, Averion International, Southborough, Massachusetts

ABSTRACT

It is cliché, nonetheless true, that we all want SAS programs that are as easy to use as a push-of-a-button. This holds especially true for generating SAS graphics, like bar charts, scatter plots, line plots, and Kaplan-Meier plots. This paper will introduce a graph macro that can generate a variety of SAS graphical displays by inputting simple graphical attributes, such as the input data set and the axis variables.

INTRODUCTION

With a generic graph macro, a user does not need to have any prior knowledge of **SAS/GRAPH**. This directly translates into time and cost efficiency in the production environment. In addition, we can be more assured that the graphical displays produced are consistent in quality once the graph macro is validated.

The **gengph** macro will dynamically generate various types of popular graphical displays, such as bar charts, scatter plots, scatter plots with regression line, and Kaplan-Meier time to event plots based on the user input. This paper will walk you through the parameters in this **gengph** macro. Examples from clinical data will be used to demonstrate how this macro works.

An explanation of what's behind the scene will be provided in this paper. Specifically, the **symbol, axis, legend statements** in **SAS/GRAPH**, and the SAS source code for generating the various graphical displays will also be explored in depth.

The **gengph** macro has the following macro parameters:

```
%gengph( inds=,
         xvara=,
         xlabel=,
         xrange=,
         xfmt=,
         yvara=,
         ylabel=,
         yrange=,
         yfmt=,
         byvara=,
         bylabel=,
         byfmt=,
         gproc=,
         refl=);
```

Here is the definition of the macro parameters.

INDS: denotes the name of the input data set.

For the x-axis:

XVARA: Variable for the X-axis in the input data set

XLABEL: Label for the X-axis

XRANGE: Optional range for the X-axis

XFMT: Optional user-defined format for the X-axis

For the y-axis:

YVARA: Variable for the Y-axis in the input data set

YLABEL: Label for the Y-axis

YRANGE: Optional range for the Y-axis

YFMT: Optional user-defined format for the Y-axis

An additional variable may be used:

BYVARA: Optional classification variable in the input data set
BYLABEL: Label for the classification variable
BYFMT: Optional user-defined format for the classification variable

GPROC: Specify the graphical display:

BCHART - a vertical bar chart

SPLOTR - a scatter plot of X and Y with a regression line

SPLOT - a scatter plot

LPLOT - a line plot

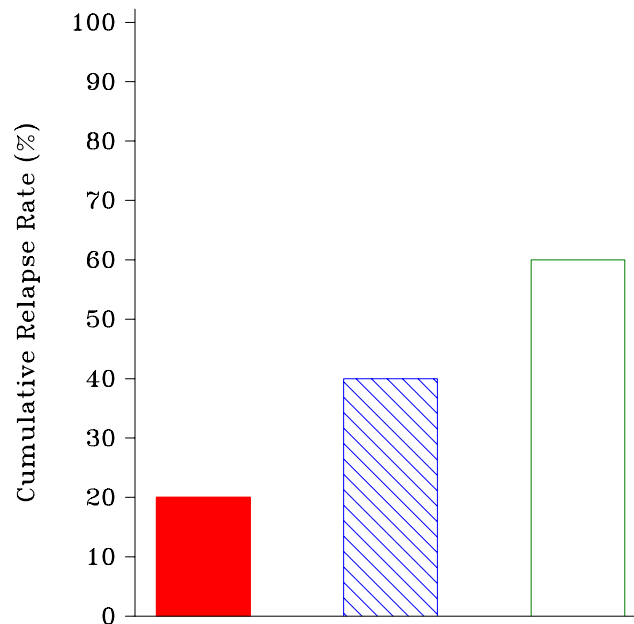
KMPLOT - a Kaplan-Meier time to event plot. For KMPLOT, the macro parameters X RANGE and BYVARA must be provided.

REFL: Optional reference line for the X or Y axis

EXAMPLE 1

The following macro call will create a bar chart of cumulative relapse rate by treatment.

Cumulative Relapse Rate at Week 4 by Treatment



Treatment AAAAAAA BBBBBBB Vehicle

The input data set is called **all1**, which contains the variable for the treatment group (**treat**), and the variable for the cumulative relapse rate (**percent**). A user-defined format **rantrt** is specified for the treatment variable, with decodes of 1=AAAAAAA, 2=BBBBBBB, 3=Vehicle. The label for the x-axis is **Treatment**. We define the range of the cumulative relapse rate value from **0 to 100**, with an increment of **10**. The label for the y-axis is **Cumulative Relapse Rate (%)**. The graphical display specified in this macro is a **bar chart (bchart)**.

****user-defined format;**

```

proc format;
  value rantrt
    1='AAAAAAA'
    2='BBBBBBB'
    3='Vehicle';
run;

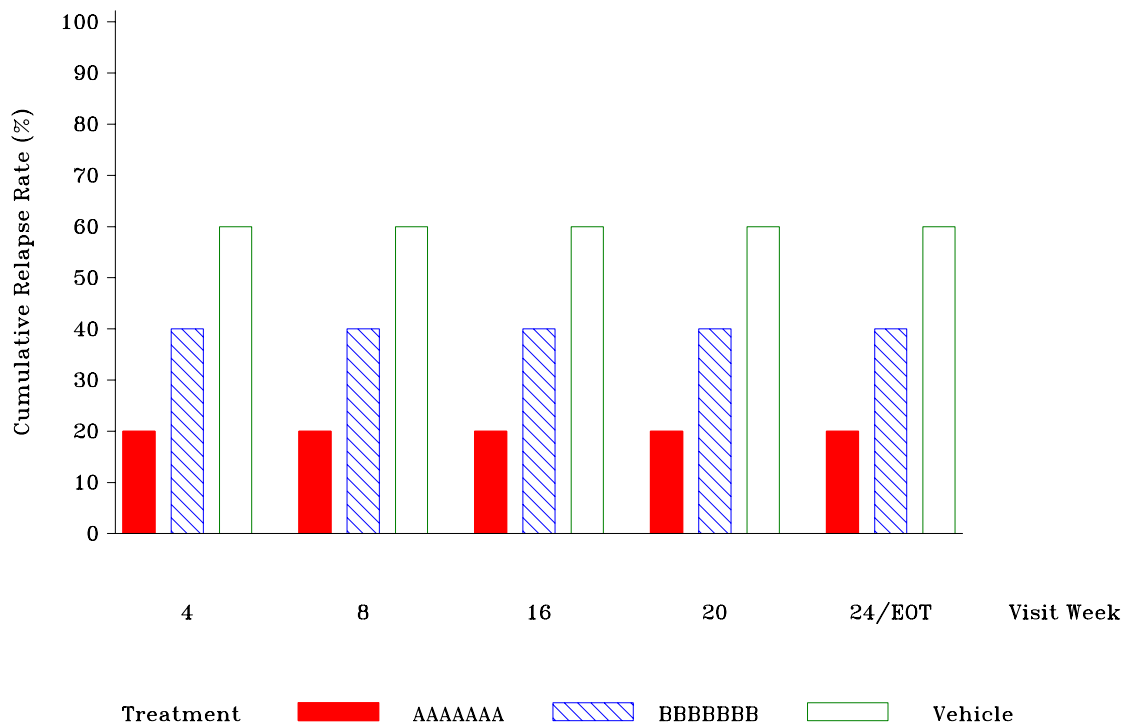
**call the gengph macro for the bar chart;
%gengph(inds=all1,
  xvara=treat,
  xlabel=%str(Treatment),
  xrange=,
  xfmt=rantrt,
  yvara=percent,
  ylabel=%bquote(Cumulative Relapse Rate (%)),
  yrange=%str(0 to 100 by 10),
  yfmt=,
  byvara=,
  bylabel=,
  byfmt=,
  gproc=bchart,
  refl=);

```

EXAMPLE 2

The following macro call will create a bar chart of cumulative relapse rate by treatment and visit week.

Cumulative Relapse Rate over time by Treatment



In this example, we have an additional time variable (**week**) in the **all1** data set. A user-define format **wk** is specified for the time variable, with decodes of 4=4, 8=8, 16=16, 20=20, 24=24/EOT. The label is **Visit Week**

```
**user-defined format;
```

```
proc format;  
  value rantrt  
    1='AAAAAAA'  
    2='BBBBBBB'  
    3='Vehicle';
```

```
Value wk  
  4='4'  
  8='8'  
 16='16'  
 20='20'  
 24='24/EOT';
```

```
run;
```

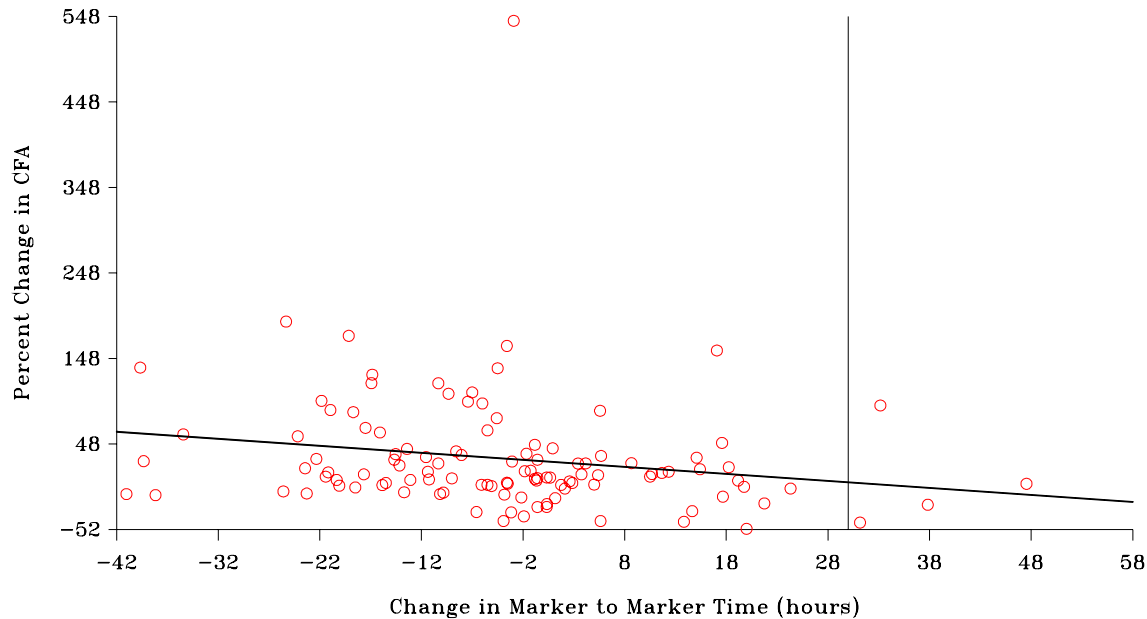
```
**call the gengph macro for the bar chart;
```

```
%gengph(inds=all1,  
  xvara=treat,  
  xlabel=%str(Treatment),  
  xrange=,  
  xfmt=rantrt,  
  yvara=percent,  
  ylabel=%bquote(Cumulative Relapse Rate (%)),  
  yrange=%str(0 to 100 by 10),  
  yfmt=,  
  byvara=week,  
  bylabel=%str(Visit Week),  
  byfmt=wk,  
  gproc=bchart,  
  refl=);
```

EXAMPLE 3

The following macro call will create a scatter plot with a regression line for Percent Change in CFA (Y) and Change in Marker to Marker Time (X). A reference line of X=30 will also be added.

Correlation of Percent Change in CFA with Change in Marker to Marker Time



Regression Equation:
 $y = 27.87919 - 0.819191 * X$

The input data set is called **splot**, which contains the variable for the **change in marker to marker time (x)**, and the variable for **percent change in CFA (pctc_cfa)**. The graphical display specified in this macro is a **scatter plot with regression line (splotr)**. The parameter to draw an optional line is **refl**. To draw **x=30**, simply input **refl=%str(href=30)**.

```

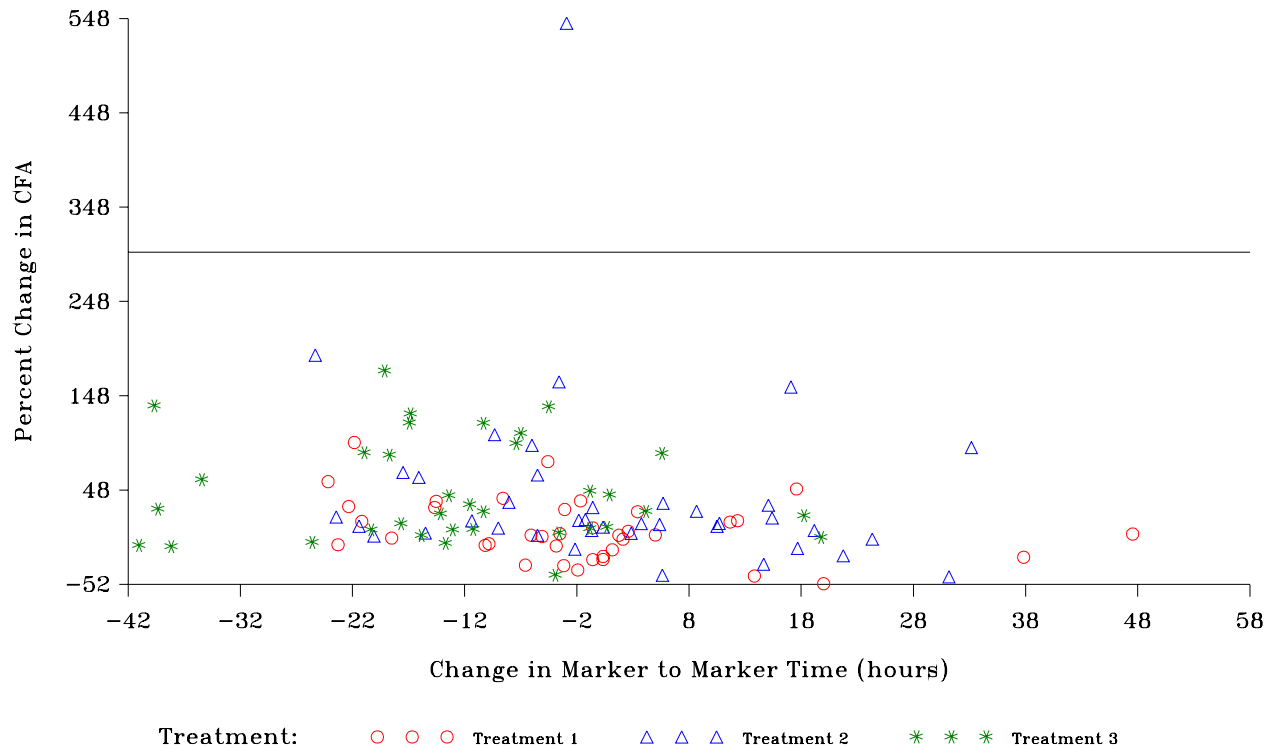
**call the gengph macro for the scatter plot with regression line;
%gengph(inds=splot,
  xvara=x,
  xlabel=%str(Change in Marker to Marker Time (hours)),
  xrange=,
  xfmt=,
  yvara=pctc_cfa,
  ylabel=%str(Percent Change in CFA),
  yrange=,
  yfmt=,
  byvara=,
  bylabel=,
  byfmt=,
  gproc=splotr,
  refl=%str(href=30));

```

EXAMPLE 4

The following macro call will create a scatter plot of Percent Change in CFA (Y) and Change in Marker to Marker Time (X) by Treatment. A reference line of Y=300 will also be added.

Correlation of Percent Change in CFA with Change in Marker to Marker Time



An additional variable, **treatment (trt)**, is added to this example. A user-defined format is specified for treatment (**trtf**), with the decodes of 1=Treatment 1, 2=Treatment 2, 3=Treatment 3. For the reference line **y=300**, simply input **refl=%str(vref=300)**.

```

**user-defined format;
proc format;
  value trtf
    1='Treatment 1'
    2='Treatment 2'
    3='Treatment 3';
run;

**call the gengph macro for the scatter plot;
%gengph (inds=splot,
  xvara=x,
  xlabel=%str(Change in Marker to Marker Time (hours)),
  xrange=,
  xfmt=,
  yvara=pctc_cfa,
  ylabel=%str(Percent Change in CFA),
  yrange=,

```

```

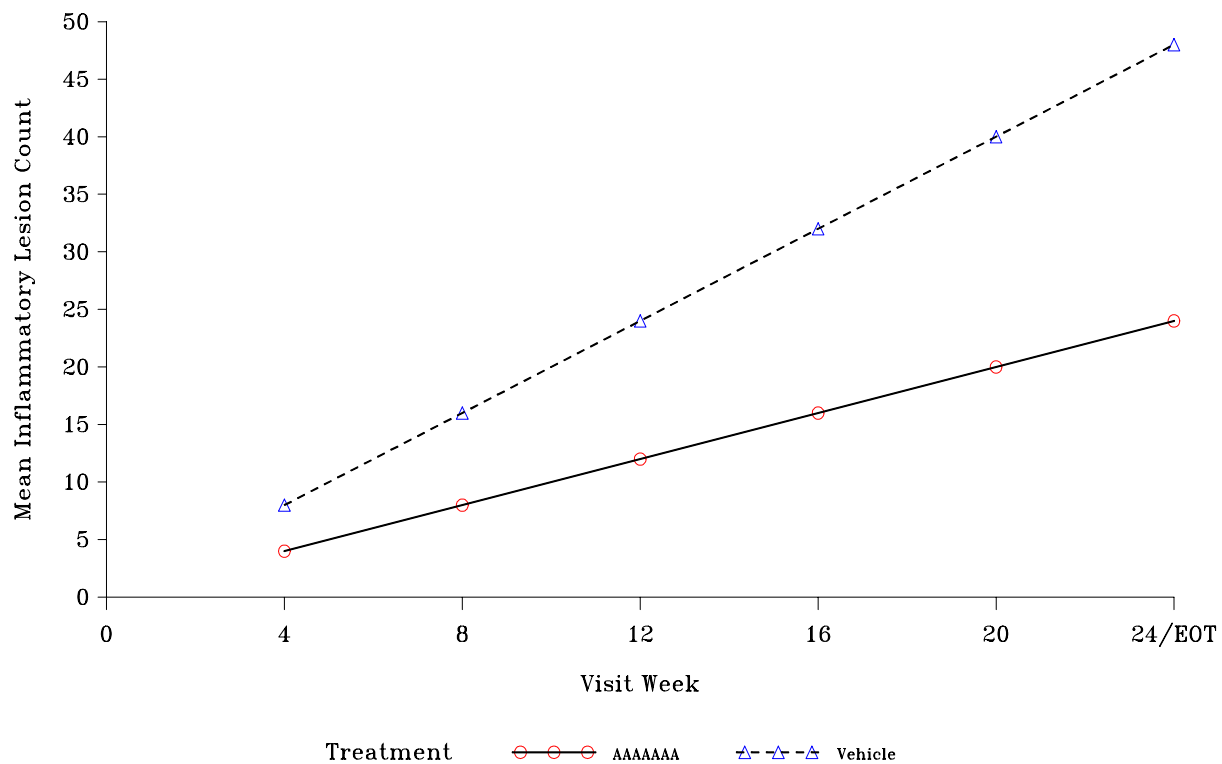
yfmt=,
byvara=trt,
bylabel=%str(Treatment: ),
byfmt=trtf,
gproc=splot,
refl=%str(vref=300));

```

EXAMPLE 5

The following macro call will create a line plot of Mean Inflammatory Lesion Counts by Visit and Treatment.

Mean Lesion Counts at Scheduled Evaluations by Treatment



The input data set is called **stat**, which contains the variables for **Visit Week (week)**, **Mean Inflammatory Lesion Count (mean)**, and **Treatment (treatn)**. A user-defined format **wk** is specified for **visit week**. We define the range of **week** from **0 to 24** with an increment of 4. A user-defined format **rantrt** is specified for **treatn**. The graphical display specified in this macro is a **line plot (lplot)**.

```

**user-defined format;
proc format;

```

```
value rantrt
  1='AAAAAAA'
  2='Vehicle';

Value wk
  0='0'
  4='4'
  8='8'
  16='16'
  20='20'
  24='24/EOT';

run;
```

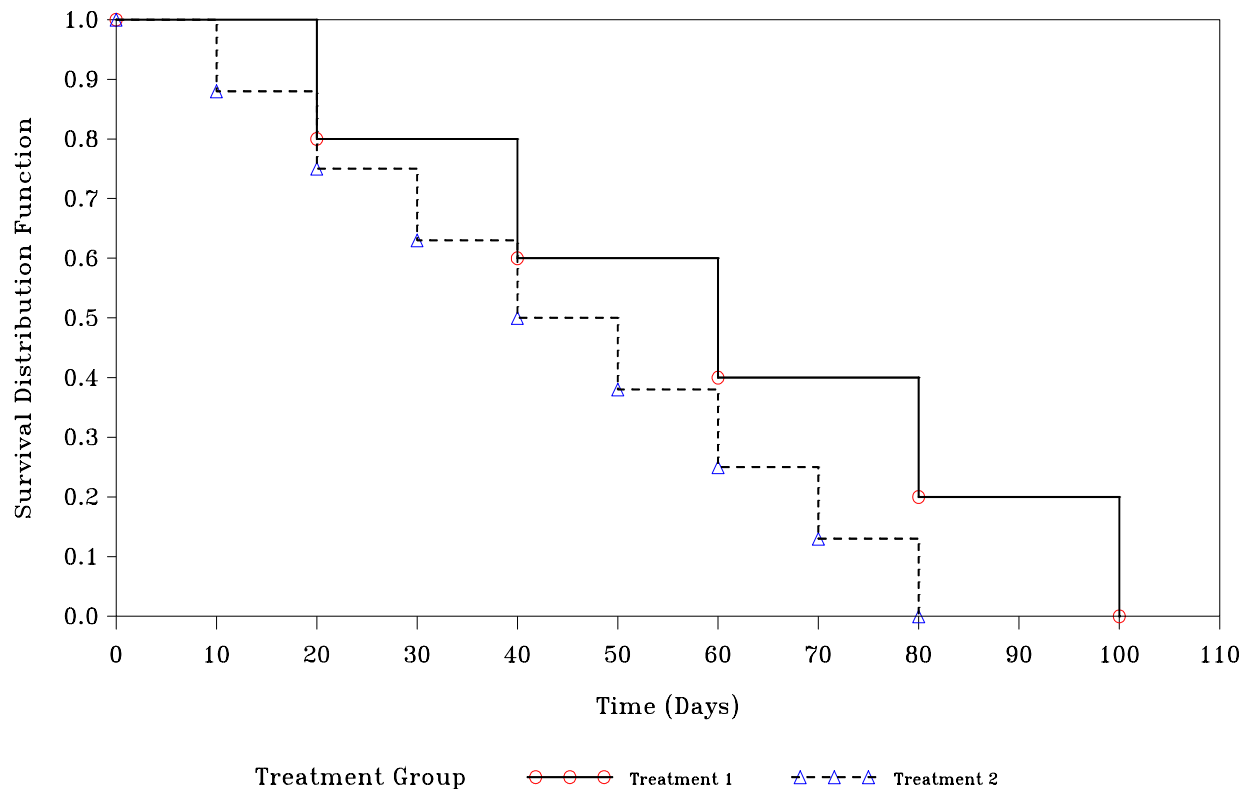
```
**call the gengph macro for the line plot;
```

```
%gengph(inds=stat,  
  xvara=week,  
  xlabel=%str(Visit Week),  
  xrange=%str(0 to 24 by 4),  
  xfmt=wk,  
  yvara=mean,  
  ylabel=%str(Mean Inflammatory Lesion Count),  
  yrange=,  
  yfmt=,  
  byvara=treatn,  
  bylabel=%str(Treatment),  
  byfmt=rantrt,  
  gproc=lplot,  
  refl=);
```


EXAMPLE 6

The following macro call will create a Kaplan-Meier plot of the survival distribution function of mortality by time to death and treatment group.

Kaplan–Meier Estimates of All Cause Mortality by Treatment Group



The input data set is called **main**, which contains the variables for **Time (t_death)**, **Survival Distribution Function (death)**, and **Treatment Group (rantrt_)**. A user-defined format **trtf** is specified for **Treatment Group**. We define the range of **time** from **0 to 110** with an increment of 10. We define the **Survival Distribution Function** from **0 to 1** with an increment of 0.1. A user-defined format **rantrt** is specified for **treatn**. The graphical display specified in this macro is a **Kaplan-Meier (kmplot)**.

```
**user-defined format;
proc format;
  value trtf
    1='Treatment 1'
    2='Treatment 2';
run;
```

```
%gengph(inds=main,
```

```

xvara=t_death,
xlabel=%nrstr(Time (Days)),
xrange=%str(0 to 110 by 10),
xfmt=,
yvara=death,
ylabel=%str(Survival Distribution Function),
yrange=%str(0 to 1 by 0.1)
yfmt=,
byvara=rantrt__,
bylabel=%str(Treatment Group),
byfmt=trtf,
gproc=kmpplot,
refl=);

```

WHAT'S BEHIND THE SCENE?

SYMBOL, AXIS, AND LEGEND STATEMENTS

The **symbol**, **axis**, and **legend** statements in **SAS/GRAPH** form the backbone of SAS graphics. The **gengph** macro will dynamically assign these statements based on the user input.

The **symbol** statements in **SAS/GRAPH** define the attributes of the plot line and symbols in scatter plots, line plots, and Kaplan-Meier time to event plots. These graphical displays use similar **symbol** statements. It is the **interpol** option that sets them apart. By varying the input value for the macro parameter **grproc** in the **gengph** macro, we can dynamically change the **symbol statements** for the specified type of graphical display.

A macro variable **inter** is assigned based on the graphical display requested in the **gengph** macro. The macro variable **inter** is initialized to **none** as the default for a scatter plot. For a line plot, its value will be changed to **join**. Its value will be changed to **rl** for a scatter plot with regression line, and **steplj** for a Kaplan-Meier time to event plot.

```

**For a scatter plot;
%let inter=none;

**For a line plot;
%if "%upcase(&gproc)" = "LPLOT" %then %do;
  %let inter=join;
%end;

**For a scatter plot with regression line;
%if "%upcase(&gproc)" = "SPLOTR" %then %do;
  %let inter=rl;
%end;

**For a Kaplan-Meier time to event plot;
%if "%upcase(&gproc)" = "KMPLOT" %then %do;
  %let inter=steplj;
%end;

```

The **interpol** option, **i** for short, in the **symbol** statement will reflect the value of the macro variable **inter** assigned earlier. Five symbol statements with different line and symbol attributes have been pre-assigned.

```

symbol1 value=circle h=0.17 in cv=red ci=black width=4 i=&inter l=1;
symbol2 value=triangle h=0.17 in cv=blue ci=black width=4 i=&inter l=2;
symbol3 value=: h=0.17 in cv=green ci=black width=4 i=&inter l=3;
symbol4 value=square h=0.17 in cv=black ci=black width=4 i=&inter l=4;
symbol5 value=dot h=0.17 in cv=black ci=black width=4 i=&inter l=5;

```

The **axis** statements in **SAS/GRAPH** define the value ranges and labels of the x-axis and the y-axis. In the **gengph** macro, the **axis** statements can be dynamically assigned based on user input for the macro parameters **xrange**, **yrange**, **xlabel**, and **ylabel**. The **gengph** macro can also find the value ranges based on the input data if no value ranges are provided. Essentially the value ranges are defined by the macro variables **miny**, **maxy**, **byvaly**, **minx**, **maxx**, **byvalx** created in the **gengph** macro. For the scope of this paper, the source code for generating these value ranges will not be discussed here.

```

**Defining the Y-axis;
%if "&yrange" = "" %then %do;
  axis1 minor=none order=(&miny to &maxy by &byvaly) offset=(0 pct)
  label=(angle=90 font=triplex h=0.175 in "&ylabel") length=4 in;
%end;
%else %do;
  axis1 minor=none order=(&yrange) offset=(0 pct)
  label=(angle=90 font=triplex h=0.175 in "&ylabel") length =4 in ;
%end;

**Defining the X-axis;
%if "&xrange" = "" %then %do;
  axis2 minor=none order=(&minx to &maxx by &byvalx)
  offset=(0 pct) label=(font=triplex h=0.175 in "&xlabel") ;
%end;
%else %do;
  axis2 minor=none order=(&xrange)
  offset=(0 pct) label=(font=triplex h=0.175 in "&xlabel") ;
%end;

```

If an additional classification variable is specified, e.g. treatment group, a legend statement will be assigned.

```

%if "&bylabel" ne "" %then %do;
  legend1 label=(f=triplex h=0.175 in "&bylabel" j=1) value=(f=triplex h=0.12 in
  j=1);
%end;

```

SCATTER PLOTS, SCATTER PLOTS WITH REGRESSION LINE, LINE PLOTS

The **gengph** macro utilizes the **GLOT** procedure in **SAS/GRAPH** to create scatter plots, scatter plots with regression lines, and line plots, when the macro values **SLOT**, **SLOT**, and **LLOT** are provided for the parameter **gproc** respectively.

The **GLOT** syntax is exactly the same for **SLOT**, **SLOT** and **LLOT**. Scatter plots with a regression line (**SLOT**) has an additional option **regegn** in the **PLOT** statement to generate a linear regression line for the Y and X variables.

Conditional branching will be employed in the **gengph** macro to apply user formats for the x, y, and/or an optional classification variable. An optional reference line can also be specified as in the **PLOT** statement based on the user input. For example, a **refl** value of **vref=300** will add a **vref** option in the **PLOT** statement to draw a line at 300 on the y-axis. Similarly, a **refl** value of **href=300** will add a **href** option in the **PLOT** statement to draw a line at 30 on the x-axis.

A legend will be specified as an option, i.e. **nolegend** or **legend=legend1**, in the **PLOT** statement if a classification variable is present.

```

%if "%upcase(&gproc)" = "SPLOT" or "%upcase(&gproc)" = "LPLLOT" %then %do;
  proc gplot data=&inds;

    **Add user-defined formats;
    %if "%upcase(&xfmt)" ne "" %then %do ;
      format &xvara &xfmt.;
    %end;
    %if "%upcase(&yfmt)" ne "" %then %do ;
      format &yvara &yfmt.;
    %end;

    **The following code will be executed when no classification variable
    is specified;
    %if "%upcase(&byvara)" = "" %then %do;
      plot &yvara*&xvara/noframe vaxis=axis1 haxis=axis2 &refl nolegend;
    %end;

    **The following code will be executed when a classification variable
    is specified;
    %else %do;
      plot &yvara*&xvara=&byvara/noframe vaxis=axis1 haxis=axis2
        &refl legend=legend1;
    %end;
  run;
  quit;
%end;

```

The following code will be executed when **SPLOTR** is provided for the macro parameter **gproc**.

```

%if "%upcase(&gproc)" = "SPLOTR" %then %do;
  proc gplot data=&inds;
    %if "%upcase(&xfmt)" ne "" %then %do ;
      format &xvara &xfmt.;
    %end;
    %if "%upcase(&yfmt)" ne "" %then %do ;
      format &yvara &yfmt.;
    %end;
    %if "%upcase(&byvara)" = "" %then %do;
      plot &yvara*&xvara/noframe regeqn vaxis=axis1 haxis=axis2
        &refl nolegend;
    %end;
    %else %do;
      plot &yvara*&xvara=&byvara/noframe regeqn vaxis=axis1 haxis=axis2
        &refl legend=legend1;
    %end;
  run;
  quit;
%end;

```

KAPLAN-MEIER TIME TO EVENT PLOTS

Similarly, the **gengph** macro utilizes the **GLOT** procedure in **SAS/GRAPH** to create the Kaplan-Meier time to event plots when **KMPLOT** is provided for the parameter **gproc**. Additional processing of the input data set, however, is required before the plots can be generated. Specifically the **LIFETEST** procedure will be used to generate a survivor function data set via **SAS ODS** before the **GLOT** procedure. Conditional branching will also be used to dynamically assign formats and labels.

```

%if "%upcase(&gproc)" = "KMPLLOT" %then %do;

  **Generate K-M survivor function using ODS;
  proc lifetest data=&inds method=km ;
    time &xvara*&yvara(0);
    strata &byvara;
    ods output productlimitestimates=lifeout;
  run;

  **Get maximum time to event;
  data _null_;
    call symput("_maxt",scan(scan("&xrange",2,"to"),1,"by"));
  run;

  **Carry forward any missing values in the survivor function;
  data lifeout;
    set lifeout;
    retain surv .;
    if survival>. then surv=round(survival,.01);
    format &xvara 8. surv 5.1;
  run;

  proc sort data=lifeout;
    by &byvara &xvara;
  run;

  **cap maximum time for display;
  data lifeout;
    set lifeout;
    by &byvara &xvara;
    if &xvara<&_maxt then output;
    if &xvara=&_maxt then do;
      if first.&xvara then output;
    end;
  run;

  **Generate the time to event plot;
  proc gplot data = lifeout;
    %if "%upcase(&xfmt)" ne "" %then %do;
      format &xvara &xfmt.;
    %end;
    %if "%upcase(&yfmt)" ne "" %then %do ;
      format surv &yfmt.;
    %end;
    %if "%upcase(&byfmt)" ne "" %then %do ;
      format &byvara &byfmt.;
    %end;
    plot surv*&xvara=&byvara/haxis = axis2 vaxis = axis1 legend=legend1;
  run;
  quit;
%end;

```

BAR CHARTS

The **gengph** macro utilizes the **GCHART** procedure in **SAS/GRAPH** to generate bar charts when the macro value **BCHART** is provided for the parameter **gproc**. Unlike the **GPLOT** procedure, the **pattern** statements are required for the **GCHART** procedure to define the attributes of the vertical bars. Similar to all other graphical displays, conditional branching will be employed to assign formats and labels for the axis variables or the additional classification variable.

```

%if "%upcase(&gproc)" = "BCHART" %then %do;
  pattern1 value=s color=red;
  pattern2 value=l1 color=blue;
  pattern3 value=e color=green;
  pattern4 value=x1 color=purple;
  pattern5 value=r1 color=orange;

  axis3 label=(angle=180 h=0.175 in f=triplex " ") value=none offset=(4 pct);

proc gchart data=&inds;
%if "&byvara" ne "" %then %do;
  label &byvara="&bylabel";
%end;

%if "%upcase(&xfmt)" ne "" %then %do ;
  format &xvara &xfmt.;
%end;
%if "%upcase(&yfmt)" ne "" %then %do ;
  format y &yfmt.;
%end;
%if "%upcase(&byfmt)" ne "" %then %do ;
  format &byvara &byfmt.;
%end;

label &xvara="&xlabel";
vbar &xvara / sumvar=y
      %if "&byvara" ne "" %then %do;
        group=&byvara
        subgroup=&xvara
        discrete
        width=2
        noframe
        maxis=axis3
        raxis=axis1
        space=1;
      %end;
      %else %do;
        subgroup=&xvara
        discrete
        width=5
        noframe
        maxis=axis3
        raxis=axis1
        space=5;
      %end;

run;
quit;
%end;

```

CONCLUSION

This paper has attempted to show that generating popular graphical displays, such as bar charts, scatter plots, line plots, and Kaplan-Meier time to event plots, is as easy as push-of-a-button with this generic graph macro. Examples in clinical trials have been used to demonstrate how this generic graph macro works.

This paper has also shown some source code behind the scene. The advantage of having a generic graph macro is that the user does not need to have any prior knowledge of SAS/GRAPH, which directly helps translate into time and cost efficiency. A generic graph macro provides an important first step for any programming environment.

ACKNOWLEDGEMENT

The author would like to express her appreciation to the following individuals for their invaluable comments and suggestions in this paper:

Shannon Harkness, GE Healthcare
C.H. Ho, Averion International
Jeff Orvedahl, Averion International

CONTACT INFORMATION

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

Rita Tsang
Averion International
225 Trunpike Road
Southborough, MA 01772
Rita.Tsang@averionintl.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.