PharmaSUG 2012 - Paper DG05

A Multifaceted Approach to Generating Kaplan-Meier and Waterfall Plots in Oncology Studies

Stacey D. Phillips, PharmaNet/i3, Madison, Wisconsin

ABSTRACT

The new Statistical Graphics (SG) procedures and Graph Template Language (GTL) available in SAS® 9.2 have greatly increased the flexibility and ease with which graphs and figures are generated from SAS. This paper will demonstrate multiple methods of generating the commonly utilized Kaplan-Meier curve and waterfall plot from oncology data. Examples and discussion will range from traditional SAS/GRAPH® procedures such as PROC GPLOT to Statistical Graphics (PROC SGPLOT) and the Output Delivery System (ODS) with GTL.

INTRODUCTION

Clear and effective graphics are a popular way to explore, analyze and display data of all types. Creating graphics in SAS has historically been a challenging and complicated task due to the complex nature of the SAS/GRAPH syntax and the limited options for output format types. With the release of SAS 9.2 there are many new options for displaying data that require levels of SAS expertise ranging from novice to expert. This paper will be a general overview of several of these new options and will describe multiple methods of developing graphics ranging from the "old" methods of SAS/GRAPH with PROC GPLOT to the "new" methods involving ODS with Graph Template Language (GTL) . Two types of figures that are commonly used in oncology studies (Kaplan-Meier curves and waterfall plots) will demonstrate the multiple approaches to graphics creation. Some knowledge of SAS/GRAPH and statistical graphics procedures is helpful but not necessary to understand this paper.

Old method – SAS/GRAPH with PROC GPLOT

SAS/GRAPH is a powerful and flexible way to create high-quality graphics but it can be difficult and cumbersome to learn the different syntax. Further, customizing the graphs to the extent that is often required in the pharmaceutical industry requires an intricate and detailed knowledge of the Annotate Facility. Still it is a great option for users utilizing SAS versions less than 9.2 and those who want complete control over their graphic outputs. The most common way to generate both Kaplan-Meier curves and waterfall plots using SAS/GRAPH is a combination of PROC GPLOT and the Annotate Facility. "Approach 1" demonstrates this method in the examples below for both K-M curves and waterfall plots. This approach often uses the most lines of code and often requires more data manipulation than the subsequent examples that utilize the SAS Output Delivery System (ODS).

New method – SAS Output Delivery System (ODS)

The SAS Output Delivery System (ODS) creates statistical graphs in three ways: (Kuhfeld, 2010, pg. 1)

- **Automatically created graphs:** With the release of SAS 9.2, SAS/STAT® procedures use ODS Graphics to produce graphs as automatically as they produce tables. You only need to enable ODS Graphics with the **ods graphics on** statement to get default graphical output. Some graphs require you to additionally specify one or more simple options.
- **SG procedures:** The SG (statistical graphics) procedures (SGPLOT, SGSCATTER, and SGPANEL) provide a simple and convenient syntax for producing many types of statistical graphs. They are particularly convenient for exploring and presenting data.
- **The GTL:** The GTL (graph template language) and PROC SGRENDER provide a powerful syntax for creating custom graphs. You can also modify the templates.

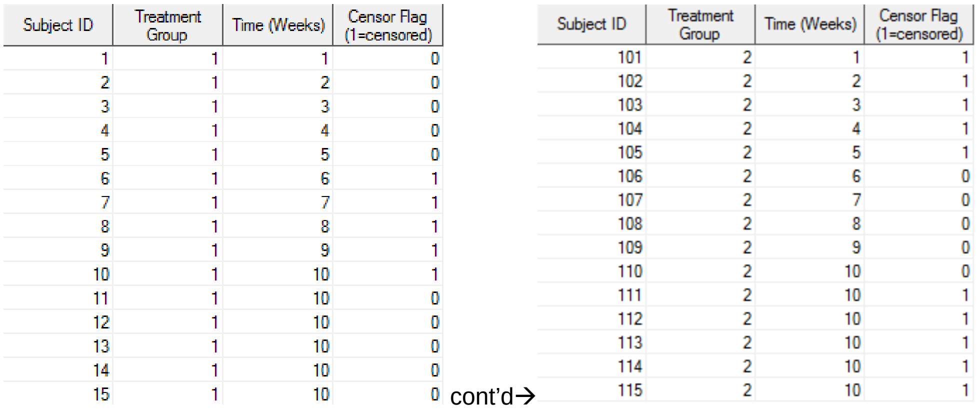
The first method is very simple and requires no knowledge of graph programming as these types of graphs are generated with a simple statement in a regular SAS/STAT procedure. The automatically generated graphs are ideal for exploring and informally analyzing data. The second method is slightly more involved than the first but since the syntax of the SG procedures is fairly simple it's much easier to generate a plot quickly than in older methods involving SAS/GRAPH. Like the automatically generated graphs, those involving the SG procedures are ideal for exploring data in addition to also producing presentation-quality graphics. The GTL method is the most flexible of all though does require more advanced knowledge of template language and/or the ability to search documentation for relevant syntax.

KAPLAN-MEIER CURVES

Kaplan-Meier (K-M) survival curves are often used in oncology studies as a graphic display of survival information for subjects involved in clinical trials. K-M curves are used to estimate the survival of patients in a study after receiving study drug and can demonstrate where and at what point subjects drop off of a study. An important advantage of the

Kaplan–Meier curve is that the method can take into account some types of censored data, particularly right-censoring, which occurs if a patient withdraws from a study, i.e. is lost from the sample before the final outcome is observed.

For all four K-M examples below we will utilize the same sample data of 30 subjects in two treatment groups (indicated by the variable TRTGROUP) that contains the subject number (SUBJECT), a time variable (WEEK) and a censor flag (CENSOR) where CENSOR=1 indicates a censored observation and CENSOR=0 indicates an uncensored observation.



Subject ID	Treatment Group	Time (Weeks)	Censor Flag (1=censored)
1	1	1	0
2	1	2	0
3	1	3	0
4	1	4	0
5	1	5	0
6	1	6	1
7	1	7	1
8	1	8	1
9	1	9	1
10	1	10	1
11	1	10	0
12	1	10	0
13	1	10	0
14	1	10	0
15	1	10	0
101	2	1	1
102	2	2	1
103	2	3	1
104	2	4	1
105	2	5	1
106	2	6	0
107	2	7	0
108	2	8	0
109	2	9	0
110	2	10	0
111	2	10	1
112	2	10	1
113	2	10	1
114	2	10	1
115	2	10	1

cont'd→

Display 1. Data set named EVENT_DATA for K-M curve examples

APPROACH 1 – KAPLAN MEIER SURVIVAL CURVE USING SAS/GRAPH AND PROC GPLOT

Before the advent of SAS 9.2 a common method for deriving a K-M curve using SAS was with the SAS/GRAPH application utilizing the GPLOT procedure. This method can be very robust and flexible particularly when used with the Annotate Facility to customize the graphs and add details not part of the default procedure. The problem with using this method is that it can be difficult to understand and create for the novice programmer as SAS/GRAPH syntax is so very different from Base SAS syntax that it's almost likely learning a new language! The Annotate Facility is very powerful for customization of the graphs but is also very involved and takes much effort to learn. Still, if you are working in an earlier version of SAS than 9.2 then this approach will likely be the best that is available to you.

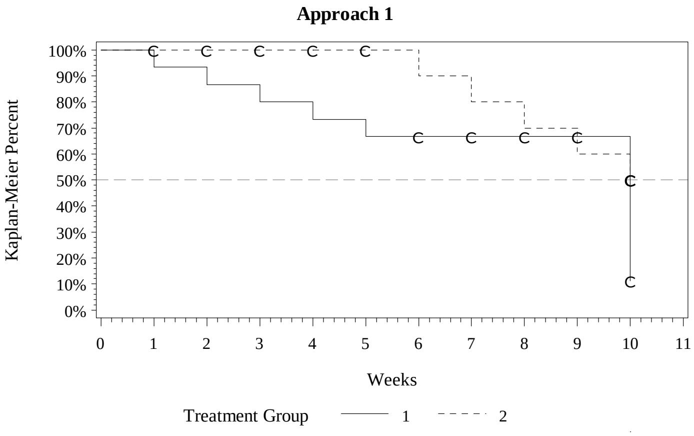


Figure 1. Approach 1: K-M curve with SAS/GRAPH and PROC GPLOT

```
   trtgroup     week     _CENSOR_     SURVIVAL     SDF_LCL     SDF_UCL     STRATUM

   1              0         .          1.00000     1.00000     1.00000        1
   1              1         0          0.93333     0.80710     1.00000        1
   1              2         0          0.86667     0.69464     1.00000        1
   1              3         0          0.80000     0.59758     1.00000        1
   1              4         0          0.73333     0.50954     0.95712        1
   1              5         0          0.66667     0.42811     0.90523        1
   1              6         1          0.66667        .           .           1
```

```
(cont’d)
trtgroup     week     _CENSOR_     SURVIVAL     SDF_LCL     SDF_UCL     STRATUM
1             7          1         0.66667         .           .           1
1             8          1         0.66667         .           .           1
1             9          1         0.66667         .           .           1
1            10          0         0.11111      0.00000     0.31385        1
1            10          1            .            .           .           1
2             0          .         1.00000      1.00000     1.00000        2
2             1          1         1.00000         .           .           2
2             2          1         1.00000         .           .           2
2             3          1         1.00000         .           .           2
2             4          1         1.00000         .           .           2
2             5          1         1.00000         .           .           2
2             6          0         0.90000      0.71406     1.00000        2
2             7          0         0.80000      0.55208     1.00000        2
2             8          0         0.70000      0.41597     0.98403        2
2             9          0         0.60000      0.29636     0.90364        2
2            10          0         0.50000      0.19010     0.80990        2
2            10          1            .            .           .           2
2            10          1            .            .           .           2
2            10          1            .            .           .           2
2            10          1            .            .           .           2
2            10          1            .            .           .           2
```

Output 1. Output data set “SURVIVAL_DATA” from LIFETEST procedure used to generate K-M curve.

```
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  GET LIFETEST OUTPUT TO BE USED IN SURVIVAL CURVE
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~;
proc lifetest data=event_data method=KM CONFTYPE=LINEAR outsurv=survival_data ;
  strata trtgroup / test=(LOGRANK);
  time week*censor(1);
run;

/* X-AXIS, Y-AXIS AND LEGEND DETAILS*/
axis1   label=(a=90 "Kaplan-Meier Percent") order  =(0 to 1 by .1);
axis2   label=("Weeks")                     order  = (0 to 11 by 1 );
legend1 label=("Treatment Group") position=(bottom center outside);

***** ANNOTATE THE CURVE ONTO THE GRAPH;
proc sort data=survival_data out=_anno;
  by trtgroup week;
run;

data anno_a;
  set _anno(where=(_censor_ ne 1));
  by trtgroup week;
  if last.trtgroup;
run;

data anno_b;
  set _anno;
  by trtgroup week;
  if last.trtgroup;
run;

data anno_lines;
    length function $8 xsys ysys $1 x y 8;
  merge anno_a anno_b (rename=(week=lasttime));
  by trtgroup;
  if trtgroup=1 then do;
    color='black';
    line=1;
  end;
  if trtgroup=2 then do;
    color='black';
```

```
    line=2;
  end;
  function='move' ;  xsys='2';  ysys='2';  x=week ;  y=survival; output;
  function='draw';  x=lasttime;  y=survival;  output;
run;

**** ANNOTATE CENSOR "C" ONTO GRAPH;
data anno_cens;
  length function $8 text $200;
  %annomac ;
  set _anno;
  by trtgroup week;
  function='label';

  xsys='2';
  ysys='2';
  position='5';

  if survival=. then survival=subsur;
  subsur=survival;
  retain subsur ;

  *** Display  'C' when subject is censored ;
  if _censor_ then do;
     x=week;
     y=survival;
     %LABEL(x,  y+0.01,'C',    black, ., ., 1.0 ,Albany AMT ,  5);
  end;
run;

*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 ALL ANNOTATE DATA SETS TOGETHER
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~;
data anno;
  set anno_lines anno_cens;
run;

SYMBOL1 INTERPOL=STEPJL LINE=1 VALUE=none MODE=include color=BLACK;
SYMBOL2 INTERPOL=STEPJL LINE=2 VALUE=none MODE=include color=BLACK;

proc gplot data=survival_data anno=anno;
   plot survival*week=trtgroup
        /vref=.5
         lvref=3
         vaxis=axis1
         haxis=axis2
         legend=legend1;
   where survival ne .;
   format survival percent8.0;
   title2 'Approach 1';
run;
quit;
```

APPROACH 2 – KAPLAN-MEIER CURVE USING DEFAULT PLOT FROM PROC LIFETEST

In this second approach we take the most basic option available in ODS graphics and specify that a graph be generated directly from PROC LIFETEST using the “PLOTS=SURVIVAL” statement. This option is great for the SAS programmer with little SAS/GRAPH experience who would like a “quick and dirty” view of their data. Note how much simpler this option is than the example in Approach 1 which required plotting data points on a graph and then interpolating the lines in addition to significant customization via the Annotate Facility. A similar graph is generated in Approach 2 with far less code and no knowledge whatsoever of SAS/GRAPH. A disadvantage to this approach is the relative lack of flexibility with regards to the appearance of the output. For example, the title of “Product-Limit Survival Estimates” is the default title and to be changed would need to be edited in PROC TEMPLATE which is a somewhat complicated modification. We will look further at PROC TEMPLATE in Approach 4 later in this paper. Still, if a basic graph to explore data is needed then this approach might make the most sense.

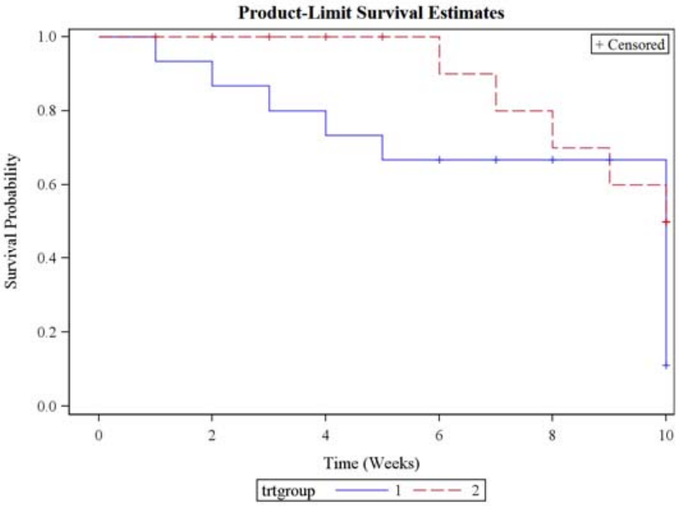


Figure 2. Approach 2: K-M curve using default plot from PROC LIFETEST

```
*CODE NOTES: "PLOTS=SURVIVAL" TELLS LIFETEST TO CREATE A SURVIVAL PLOT.
             "ODS SELECT" STATEMENT TELLS SAS TO INCLUDE ONLY THE SURVIVAL PLOT
             IN THE ODS OUTPUT.;
proc lifetest data=event_data method=KM CONFTYPE=LINEAR outsurv=survival_data
             plots=survival;
  strata trtgroup / test=(LOGRANK);
  time week * censor(1) ;
  ods select Lifetest.SurvivalPlot;
run;
```

APPROACH 3 – KAPLAN-MEIER SURVIVAL CURVE USING PROC SGPLOT

Similar to Approach 1 above we use the PROC LIFETEST output with a SAS procedure but this time utilize the statistical graphic procedure of SGPLOT. Approach 3 is still a pretty simple example of how to generate a K-M plot although is made slightly more complicated by the use of an additional DATA step to flag the censored variables with a “C” rather than the default “+”. To generate the K-M curve below with PROC SGPLOT we create the plot in two steps. The first step is to generate the actual lines of the plot which is achieved using the STEP statement that tells SAS to generate a step plot using the variable TIME on the x-axis and SURVIVAL on the y-axis. The “GROUP=” option tells SAS to generate separate lines based on the STRATUM variable (treatment group) from the PROC LIFESTEST output data set. The second step in the PROC SGPLOT approach is to tell SAS to flag the censored observations and this is achieved using the SCATTER statement in PROC SGPLOT. With this statement, a “C” is placed on graph to indicate the censored observations. We were able to specify the “MARKERCHAR=” option as a “C” in the previous DATA step otherwise the default “markerchar” from SAS would’ve been used.

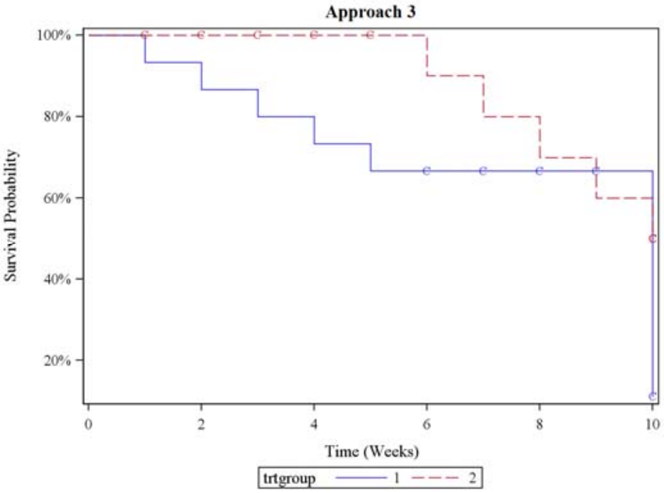


Figure 3. Approach 3: K-M curve using PROC SGPLOT

```
*CODE NOTES: IN THIS EXAMPLE WE USE "ODS OUTPUT" STATEMENT RATHER THAN
              "OUTSURV=" TO OUTPUT LIFESTEST RESULTS.;
ods output survivalplot = new_survival;
proc lifetest data=event_data method=KM CONFTYPE=LINEAR ;
    time week*censor(1) ;
    strata trtgroup/test=(LOGRANK);
run;

/*Flag the censored records as "C" for use in scatter plot below*/
data new_survival2;
    length flag $1;
  set new_survival;
  if event=0 & _n_ ne 1 then flag = 'C';
  format survival percent8.0;
run;

*CODE NOTES: "TMPLOUT=" STATEMENT GENERATES PROC TEMPLATE CODE IN APPROACH 4.;
proc sgplot data=new_survival2 TMPLOUT="C:\Users\staceyp\km.sas";
  step x=time y=survival / group  = stratum;
  scatter x=time y=censored / markerchar=flag group=stratum;
  title 'Approach 3';
run;
```

APPROACH 4 – KAPLAN-MEIER SURVIVAL CURVE USING PROC TEMPLATE AND PROC SGRENDER (GTL)

This final example demonstrates how to further customize a K-M curve with PROC TEMPLATE using the example in Approach 3 as the base code. Specifically we determine that we would like to do a little more customizing than PROC SPLOT allows for so decide to use the approach where we modify the underlying template code and then use this new template in combination with PROC SGRENDER to generate a more customized plot. Fortunately SAS has a very helpful option called “TMPLOUT=” that gives us a great start. Rather than generating the PROC TEMPLATE code from scratch this option generates the underlying PROC TEMPLATE code used in the PROC SGPLOT procedure and sends it to the program in the path that you specify. This code can then be added to or modified to suit the needs of your final output. In the example below, the PROC TEMPLATE code from Approach 3 is shown with the slight modification to the label in the plot legend (see “TMPLOUT=” on PROC SGPLOT statement in Approach 3 example). Using the DISCRETELEGEND statement this was easily changed from the default variable name of “trtgroup” to “Treatment Group.” It’s also important to note that the customization that was done in the DATA step in Approach 3 (adding “C” for censored variables) does NOT carry down using the “TMPLOUT=” option so the default “markerchars” are shown. While this example is fairly basic and little modification was actually done using PROC TEMPLATE this approach can be very powerful where significant customization is required. More advanced knowledge of ODS and GTL is needed here but with the “TMPLOUT=” option even the newest graphics programmer can make changes to the template easily.

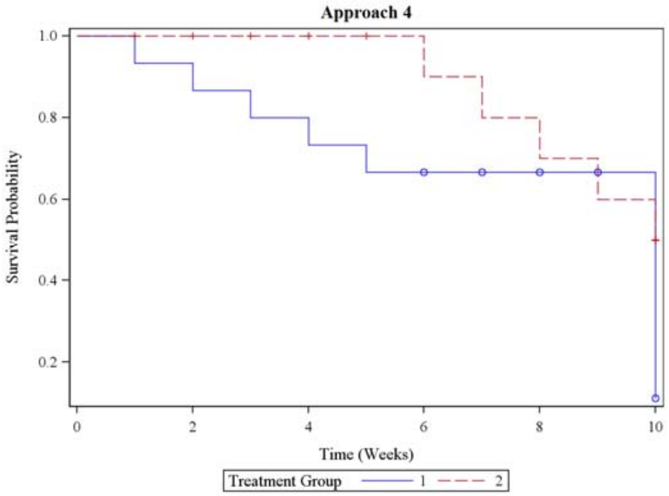


Figure 4. Approach 4: K-M curve using PROC TEMPLATE and PROC SGRENDER

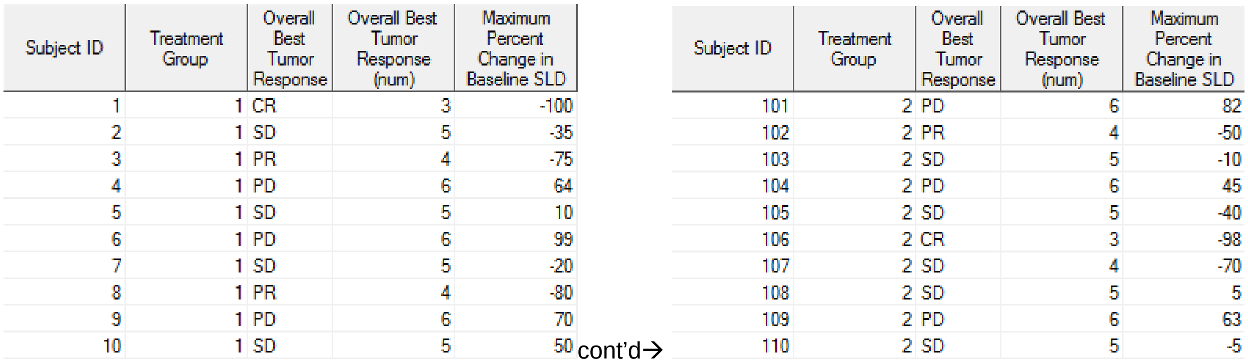
```
*CODE NOTES: PROC TEMPLATE CODE GENERATED WITH "TMPLOUT=" IN APPROACH 3.;
proc template ;
  define statgraph sgplot;
  begingraph;
  EntryTitle "Approach 4" /;
  layout overlay;
       StepPlot X=Time Y=Survival / primary=true Group=Stratum
              LegendLabel="Survival Probability" NAME="STEP";
       ScatterPlot X=Time Y=Censored  / Group=Stratum MarkerCharacter=flag
              LegendLabel="Censored" NAME="SCATTER";
       DiscreteLegend "STEP"/ title="Treatment Group";
  endlayout;
  endgraph;
  end;
run;

proc sgrender data=new_survival template=sgplot;
run;
```

WATERFALL PLOT

Waterfall plots are becoming more and more popular due their easy-to-understand representation of data. This type of chart is often used to demonstrate how a value increases or decreases for a particular variable or parameter of interest. In the below examples we will use tumor lesion and tumor response data to demonstrate how individual subjects responded to the study drug both qualitatively (tumor response as determined by an independent reviewer) and quantitatively (numeric change in tumor lesion size). Waterfall plots are especially effective at demonstrating the effects of the study drug for a large population because of their “at-a-glance” nature. Waterfall plots start with the “worst” data on the left and moves through the “best” data to the right. In this example each vertical bar represents one subject and that bar demonstrates three types of information: treatment group assignment (noted below the x-axis), qualitative response (indicated by the color of the bar and legend below the x-axis) and quantitative response (numeric representation of the increase/decrease of tumor size).

For all three examples below we will utilize the same sample data of 20 subjects in two treatment groups (indicated by the variable TRTGROUP) which contains the subject number (SUBJECT), the overall best tumor response (RESP – numeric and RESPCD - character) and the maximum percent change from baseline for sum of longest diameters (SLD).



Subject ID	Treatment Group	Overall Best Tumor Response	Overall Best Tumor Response (num)	Maximum Percent Change in Baseline SLD
1	1	CR	3	-100
2	1	SD	5	-35
3	1	PR	4	-75
4	1	PD	6	64
5	1	SD	5	10
6	1	PD	6	99
7	1	SD	5	-20
8	1	PR	4	-80
9	1	PD	6	70
10	1	SD	5	50

cont’d→

Subject ID	Treatment Group	Overall Best Tumor Response	Overall Best Tumor Response (num)	Maximum Percent Change in Baseline SLD
101	2	PD	6	82
102	2	PR	4	-50
103	2	SD	5	-10
104	2	PD	6	45
105	2	SD	5	-40
106	2	CR	3	-98
107	2	SD	4	-70
108	2	SD	5	5
109	2	PD	6	63
110	2	SD	5	-5

Display 2. SAS data set used for waterfall plot examples

APPROACH 1 – WATERFALL PLOT USING SAS/GRAPH AND PROC GPLOT

Much like Approach 1 in the Kaplan-Meier portion of this paper we will take a similar approach to creating a waterfall plot by generating our required data and then using a SAS/GRAPH procedure to generate the plot. SAS/GRAPH is a tried-and-true method of generating presentation quality graphics but the code for SAS/GRAPH procedures is so dissimilar to Base SAS code that it requires an investment in time that a programmer might not have. Further, while the Annotate Facility is powerful it also takes a lot to learn so a new approach to generating a waterfall plot is ideal. The SAS/GRAPH code for this example isn’t too onerous and in fact, we don’t need to use the Annotate Facility at all to achieve it. The basic procedure is to sort our underlying data in descending order of SLD (see data above) so that the maximum change values plot first. Based on this order we assign an ordinal observation number (x-axis variable) that gets plotted against the SLD value (y-axis variable) and finally plotted against a classification variable (best tumor response). Some minor customization is achieved using AXIS and LEGEND statements.

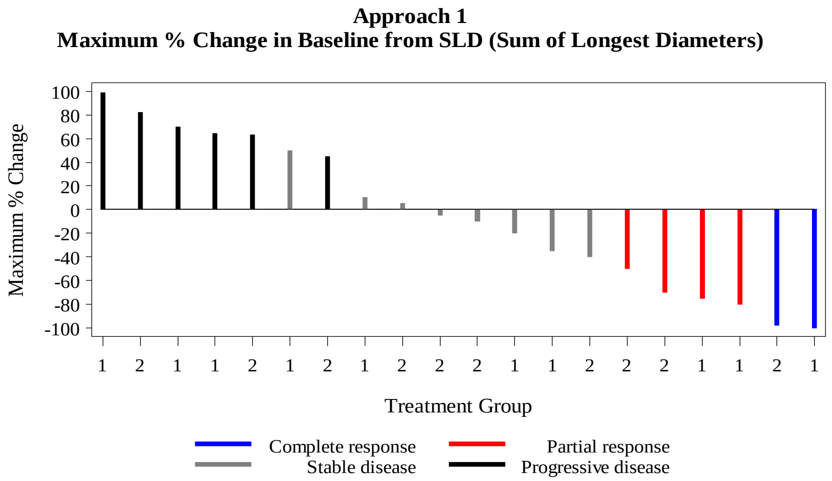


Figure 5. Approach 1: Waterfall plot using SAS/GRAPH and PROC GPLOT

```
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  GET DATA FOR PROC GPLOT :
  SORT IN DESCENDING ORDER AND CREATE A VARIABLE FOR THE ORDERED OBSERVATIONS
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~;
      proc sort data=lesion_data;
        by descending sld subject;
      run;

      data lesion_data1;
        set lesion_data;
        obs=_n_; /*To maintain the order of the data for plotting*/
      run;

      /*This format will be used in examples for Approach 2 also*/
      data fmt(keep=start label fmtname);
          length label $6;
        set lesion_data1;
        fmtname='fmt';
        label=trtgroup;
        start=obs;
      run;

      proc format cntlin=fmt;
      run;

      proc format;
        value rsrespcd
          1="Unevaluable"
          2="Not done"
          3="Complete response"
          4="Partial response"
          5="Stable disease"
          6="Progressive disease";
      run;

      axis1 label=("Treatment Group")offset=(1,1) minor=none;
      axis2 label=(angle=90 "Maximum % Change")   minor=none;
      legend1 label=(" ");
      symbol1  pointlabel=none interpol=needle  ci=blue  line=1  width=6;
      symbol2  pointlabel=none interpol=needle  ci=red   line=1  width=6;
      symbol3  pointlabel=none interpol=needle  ci=gray  line=1  width=6;
      symbol4  pointlabel=none interpol=needle  ci=black line=1  width=6;
```

```
 *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  APPROACH 1: SAS/GRAPH WITH PROC GPLOT
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~;
Proc gplot data=lesion_data1 ;
     plot sld*obs=respcd / legend=legend1
                         haxis=axis1
                         vaxis=axis2
                         hminor=0
                         vminor=1 ;
     format respcd rsrespcd.  obs  fmt.;
     title2 "Approach 1";
     title3 "Maximum % Change in Baseline from SLD (Sum of Longest Diameters)";
run;
Quit;
```

APPROACH 2 – WATERFALL PLOT USING PROC SGPLOT

In the second approach to creating a waterfall plot we again utilize the statistical graphics procedure SGPLOT which results in a similar graph to the one generated in Approach 1. Approach 2 demonstrates an alternative method to creating a waterfall plot by using PROC SGPLOT though it doesn't actually reduce the amount of code required as it did in the Kaplan-Meier examples. The key statements required for a waterfall plot in PROC SGPLOT are the VBAR statements and the XAXIS and YAXIS statements. You can see from the example that the incoming data needs to be rescaled from the data in Approach 1 and that additional data manipulations need to be done to "trick" SAS into providing the treatment group values across the bottom. Specifically, spaces are added to the treatment group values so that SAS thinks they're distinct values. If we don't do this then PROC SGPLOT will create a stacked bar chart rather than a chart with separate bars for each observation. Please note that in SAS 9.3 a waterfall plot option has been added though the online documentation indicates this option is still in preproduction.

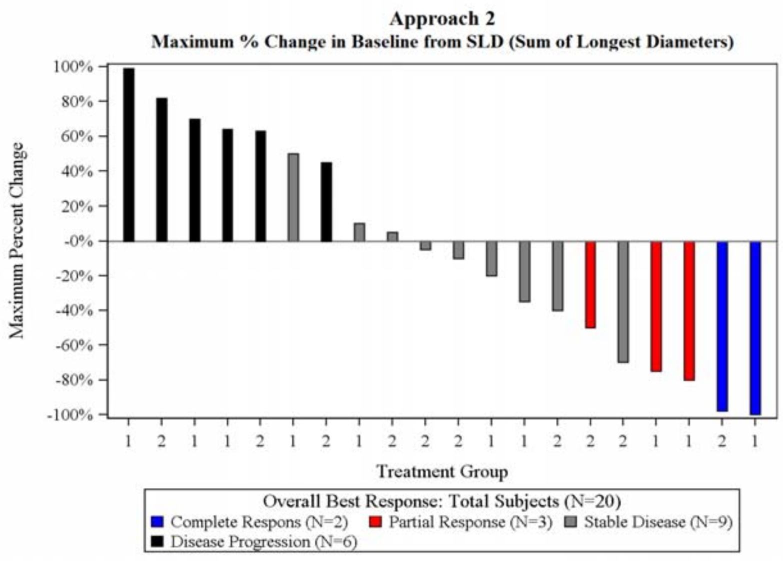


Figure 6. Approach 2: Waterfall plot using PROC SGPLOT

```
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  APPROACH 2: EXAMPLE USING SGPLOT
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~;
data lesion_data1;
  set lesion_data1;
  if sld ne . then sld=sld/100;  /*Rescale for the existing graph code*/
run;

proc sort data=lesion_data1;
  by obs sld trtgroup;
run;
```

```
/*Transpose into the format needed for SGPLOT*/
proc transpose data=lesion_data1 out=lesion2 (drop=_name_ );
  by obs sld trtgroup;
  id resp;
  idlabel resp;
  var sld;
run;

data lesion2;
  set lesion2;
  if obs ne . then trtgroup2=trim(left(put(obs,fmt.)));
  sortvar=1;
run;

proc sort;
  by sortvar obs;
run;

data lesion3;
    length spc test test2 $200;
    retain spc;
  set lesion2;
  by sortvar obs;

  /*Trick SAS into thinking these are separate values by adding spaces*/
  if first.sortvar then test=trim(left(trtgroup2));
  spc='-'||trim(left(spc));
  test=trim(left(spc))||trim(left(trtgroup2));
  if test ne '' then test2=tranwrd(test,'-','');
  label test2="Treatment Group";
run;

*CODE NOTES: "TMPLOUT=" STATEMENT GENERATES PROC TEMPLATE CODE FOR APPROACH 3.;
proc sgplot data=lesion3 TMPLOUT="C:\Users\staceyp\waterfall.sas";
    keylegend "cr" "pr" "sd" "pd"/ location=outside position=bottom
    title='Overall Best Response: Total Subjects (N=20)';
vbar test2 / response=cr fillattrs=(color=blue)
             legendlabel="Complete Response(N=2)" barwidth=.3 name='cr';
vbar test2 / response=pr fillattrs=(color=red)
             legendlabel="Partial Response (N=3)"  barwidth=.3  name='pr';
vbar test2 / response=sd fillattrs=(color=gray)
             legendlabel="Stable Disease (N=9)" barwidth=.3 name='sd';
vbar test2 / response = pd fillattrs=(color=black)
             legendlabel="Disease Progression (N=6)" barwidth=.3 name='pd';
xaxis type=discrete discreteorder=data ;
yaxis values=(-1.0 to 1.0 by .20) label='Maximum Percent Change'
             tickvalueformat=percentn8.0;
title1 "Approach 2";
title2 "Maximum % Change in Baseline from SLD (Sum of Longest Diameters)";
run;
quit;
```

APPROACH 3 – WATERFALL PLOT USING PROC TEMPLATE AND PROC SGRENDER

The final approach to generating a waterfall plot is similar to the final approach in the K-M examples and involves outputting the underlying PROC TEMPLATE code using “TMPLOUT=” and modifying accordingly. The PROC TEMPLATE code looks complicated at first glance though it is fairly obvious how the axes, labels and tickmarks are specified via the LAYOUT OVERLAY and BARCHARTPARM statements. For this example, we determine that we’d like to add a legend to the chart indicating the meaning of the treatment group values across the x-axis. After consulting the SAS Graphic Template Language documentation we determine that adding a legend is easy with ENTRY statements within the LAYOUT OVERLAY statements.

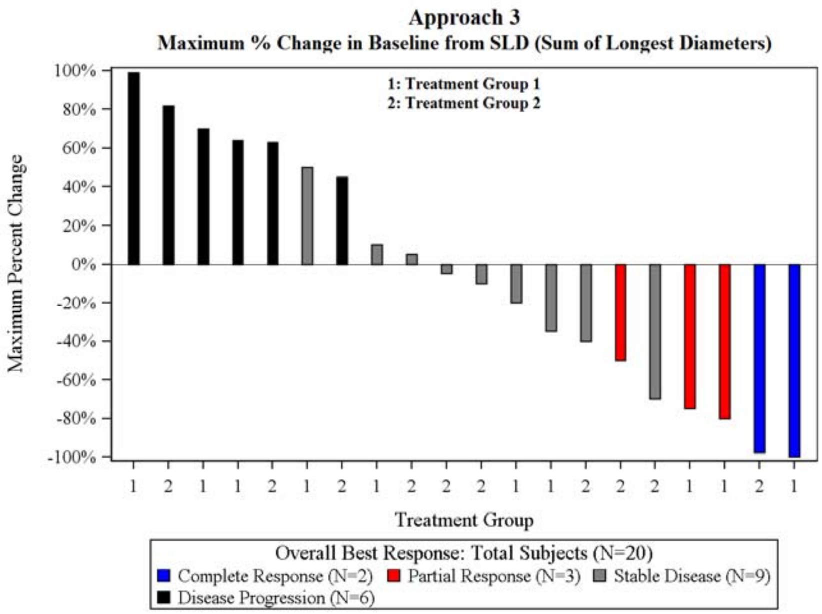


Figure 7. Approach 3: Waterfall plot using PROC TEMPLATE and PROC SGRENDER

```
*CODE NOTES: PROC TEMPLATE CODE GENERATED WITH "TMPLOUT=" IN APPROACH 2.;
proc template;
  define statgraph sgplot;
  dynamic _NEGATIVE_;
  dynamic _ticklist_;
  begingraph;
  EntryTitle "Approach 3" /;
  EntryTitle "Maximum % Change in Baseline from SLD (Sum of Longest
             Diameters)"/ textattrs=(size=GraphLabelText:fontsize);
  layout overlay / cycleattrs=true xaxisopts=(Label="Treatment Group"
  display=(tickvalues line label ticks ) type=discrete discreteopts=(
  tickValueList=_ticklist_ )) yaxisopts=( Label="Maximum Percent Change"
  type=auto linearopts=(tickvaluelist=( -1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8
  1 ) viewmin=-1 viewmax=1 TickValueFormat=PERCENTN8.0 ) );

  if (_NEGATIVE_)   ReferenceLine y=0;
  endif;
  BarChartParm X=test2 Y=CR / primary=true FillAttrs=(Color=BLUE) barwidth=0.3
             LegendLabel="Complete Response (N=2)" NAME="cr";
  if (_NEGATIVE_)   ReferenceLine y=0;
  endif;
  BarChartParm X=test2 Y=PR / FillAttrs=(Color=RED) barwidth=0.3
             LegendLabel="Partial Response (N=3)" NAME="pr";
  if (_NEGATIVE_)   ReferenceLine y=0;
  endif;
  BarChartParm X=test2 Y=SD / FillAttrs=(Color=GRAY) barwidth=0.3
               LegendLabel="Stable Disease (N=9)" NAME="sd";
  if (_NEGATIVE_)   ReferenceLine y=0;
  endif;
  BarChartParm X=test2 Y=PD / FillAttrs=(Color=BLACK) barwidth=0.3
             LegendLabel="Disease Progression (N=6)" NAME="pd";
  DiscreteLegend "cr" "pr" "sd" "pd" / Location=Outside valign=bottom
             Title="Overall Best Response: Total Subjects (N=20)";

  Layout Gridded / Border=false valign=top;
  Layout Gridded / columns=1 columngutter=5;
  Entry halign=left "1: Treatment Group 1" / textattrs=(size=9pt weight=bold);
  Entry halign=left "2: Treatment Group 2" / textattrs=(size=9pt weight=bold);
```

```
endlayout;
endlayout;
endlayout;
endgraph;
end;
run;

proc sgrender data=lesion3 template=sgplot;
run;
```

CONCLUSION

SAS has many options when it comes to analyzing and displaying data graphically and this paper summarizes different approaches for programmers in the oncology area looking to generate either Kaplan-Meier survival curves or waterfall plots. Novice graph programmers and those looking to simply explore their data might find the basic graphics from ODS and STAT procedures to be adequate while others needing more presentation quality and customized graphs might prefer the new statistical graphics procedures either with or without PROC TEMPLATE. Finally, SAS/GRAPH is effective for programmers using SAS 9.1 or earlier and those who want total control over their output including the customizability afforded via the Annotate Facility. Which approach to take depends on the programmer's level of expertise with the various methods and the requirements of the graph being generated. No matter what your programming level and output requirement there is sure to be an approach that works for you.

REFERENCES

Kuhfeld, Warren F (2010) "The Graph Template Language and the Statistical Graphics Procedures: An Example-Driven Introduction" *Proceedings of the Pharmasug 2010 Conference*. Cary, NC: SAS Institute. Available at http://www.pharmasug.org/cd/papers/SAS/TU-SAS01.pdf.

RECOMMENDED READING

- Base SAS® ODS Statistical Graphics Online Documentation: http://support.sas.com/rnd/base/topics/statgraph/ (link is largely for pre-production in v9.1 but has useful "Tip Sheets" for reference)
- SAS/STAT ODS User's Guide 9.2: http://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#odsgraph_toc.htm
- SAS/GRAPH® Online Documentation: http://support.sas.com/documentation/onlinedoc/graph/

CONTACT INFORMATION

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

Name: Stacey D. Phillips
Enterprise: Pharmanet/i3
Address: 3071 Cimarron Trl.
City, State ZIP: Madison, WI 53719
Work Phone: 608-271-6042
Fax: 608-271-6042
E-mail: sphillips@pharmanet-i3.com
Web: www.pharmanet-i3.com