

## Super Static SG Graphs

Kriss Harris, SAS Specialists Limited

### ABSTRACT

Would you like to know how to utilize ODS objects to produce graphs just the way you want them? Such as with coloring and annotation?

This submission will show you how to create the graphs you want through using a Kaplan-Meier example.

### INTRODUCTION

Outputting (Output Delivery System) ODS output objects can help you create your desired graphs because they contain the necessary data for your graphs. You will see examples of how you can use ODS objects to help make the Kaplan-Meier Plot. Knowing how to create ODS output objects will also help you not only to create Kaplan-Meier plots, but also other plots such as forest plots and least-square means plots.

### OBTAIN ODS OBJECT

One way to help know which object to obtain is to use the ODS TRACE option immediately before the statistical procedure that you are using. In our case, the LIFETEST procedure is used because we want to perform survival analysis and view the Kaplan-Meier curve. The ODS TRACE ON option writes to the SAS log a record of each output object that is created from the procedures, and the ODS TRACE OFF suppresses the writing of the output objects to the SAS log. Please see Program 1 below:

#### Program 1: Program to create Proc LIFETEST ODS Objects

```
ods trace on;
proc lifetest data = adam.adtteeff
  plots=survival(atrisk=0 to 210 by 30);
  time aval * cnsr(1);
  strata trtpn;
run;
ods trace off;
```

### OBTAIN SURVIVAL PLOT OBJECT

After running Program 1 you will see output objects in the log, and Output 1 is an extract of the ODS objects from the log.

#### Output 1: ODS Object Extract

```
-----
Output Added:
-----
Name:      SurvivalPlot
Label:     Survival Curves
Template:  Stat.Lifetest.Graphics.ProductLimitSurvival
Path:      Lifetest.SurvivalPlot
-----
```

In Output 1, you can see that the output object SurvivalPlot is created within the LIFETEST procedure. The SurvivalPlot object is helpful for creating the Kaplan-Meier plots. In order to create a data set for the SurvivalPlot output object, you can simply use ODS OUTPUT SURVIVALPLOT = <SURVIVALPLOT> before Proc LIFETEST as shown in Program 2.

### Program 2: Creating Datasets from ODS Objects

```
ods output SurvivalPlot = SurvivalPlot;
ods output HomTests=HomTests (where=(test="Log-Rank") );
proc lifetest data = adam.adtteeff plots=survival(atrisk=0 to 210 by 30);
  time aval * cnsr(1);
  strata trtpn;
run;
```

Program 2, shows an example of how to create the output objects using the LIFETEST procedure. The SurvivalPlot object helps create the data necessary to produce the Kaplan-Meier plot. The HomTests object helps to make the dataset which has the Log-Rank  $p$ -value. The statement TIME AVAL \* CNSR(1) is used to indicate the Survival time, and to indicate if any time points where censored. In this example, a value of 1 within the CNSR variable indicates that the time point is censored. The statement STRATA TRTPN is used to produce a Kaplan-Meier curve for each of the three treatments.

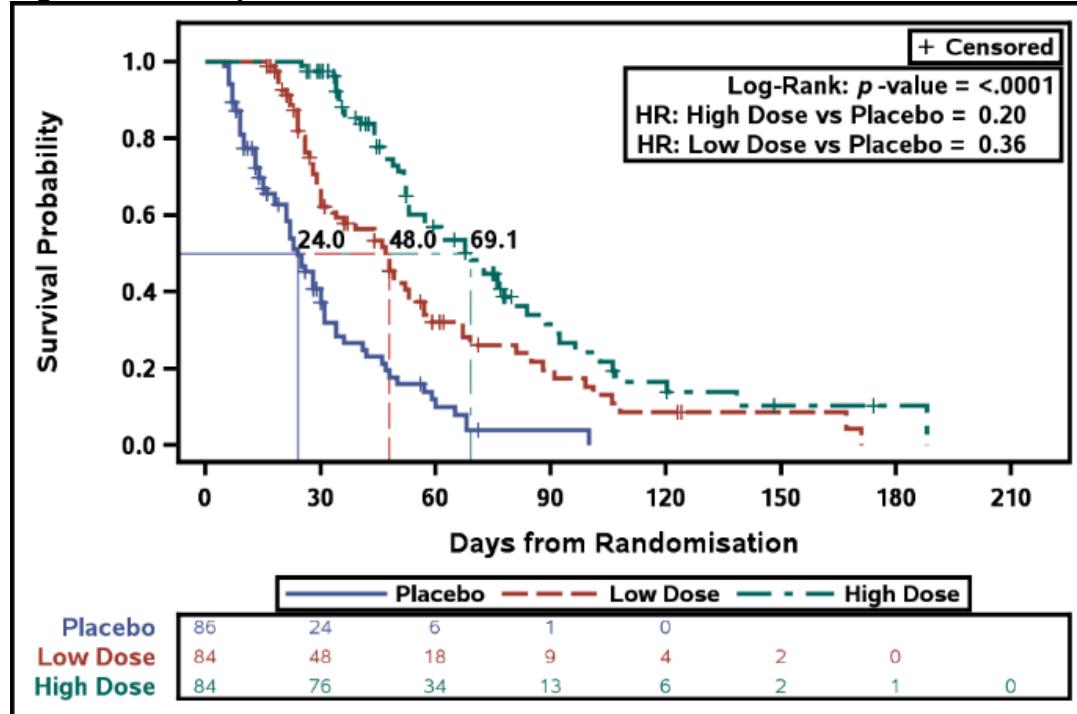
It is worth noting that Proc LIFETEST can also produce a survival plot automatically. However, the survival plot may not meet your needs. For example, you may want to show different axis tick values than those that Proc LIFETEST automatically generates, as seen in our example where we want the axis tick values to go from 0 to 210 by 30 instead of 0 to 200 by 50 (which was the default). You may wish to have Kaplan-Meier curves that have different style attributes, or you may want to add the Log-Rank  $p$ -values and hazard ratios to the plot. Therefore, it is helpful to obtain the survival plot data in those situations and then create your custom survival plot using SGPlot or Graph Template Language (GTL).

## CREATING KAPLAN-MEIER GRAPH

Now that you have the dataset to produce a Kaplan-Meier Curve, you will be shown the GTL code to produce a Kaplan-Meier Curve, as shown in Figure 1. Figure 1 shows the Kaplan-Meier Curves for each of the three treatment groups. The blue curve denotes the placebo group, the low dose group has a red curve, and the high dose group has a green curve. The plot also displays the censored legend, and there is a box that shows the Log-Rank  $p$ -value and the hazard ratios between the treated groups and Placebo.

At the bottom of the plot, you can see the patients at risk numbers beneath the Kaplan-Meier curves. You can see that at the start of the trial, there were 86 patients in the placebo group and 84 patients in the low and high dose groups.

**Figure 1: Kaplan-Meier Curve**



### STEPS TAKEN TO PRODUCE KAPLAN-MEIER GRAPH

There are a few steps taken to produce a Kaplan-Meier graph, as shown in Figure 1. You have already seen steps 1 and 2. The steps are as follows:

1. Use a time-to-event dataset, for example, ADTTE.
2. Use the LIFETEST procedure on your time-to-event dataset to obtain the Kaplan-Meier survival data set and get the median survival times.
3. Use the PHREG procedure to obtain the hazard ratios.
4. After you have created your hazard ratio and median survival times data sets, you can use the INTO clause from the SQL procedure or use the CALL SYMPUT routine to create macro variables which have the hazard ratios and the median survival times as the macro variable values. You can use any method you are most comfortable with to create the macro variables. The main thing is that you have these values stored as macro variables for use when creating the graph.
5. After you have created your macro variables representing the hazard ratios and the median survival times, and you have your survival data set of the survival probability against time, you can then use GTL to produce a plot.

### CODE FOR CREATING KAPLAN-MEIER GRAPH

As you know, it can be easier to break things down, and so you will be shown how to create the Kaplan-Meier plot in small steps. The first step you will see is how to create the Kaplan-Meier curves and censored markers, as shown in Program 3.

#### Program 3: Creating Kaplan-Meier Curves and Censored Markers

```
stepplot x = time y = survival / ①
group = stratum
name="Survival"
legendlabel="Survival";
```

```

scatterplot x=time y=censored / ②
  markerattrs=(symbol=plus)
  group=stratum;

```

❶ The STEP PLOT statement produces the Kaplan-Meier curves, and within the STEP PLOT statement, the GROUP = STRATUM option is used to produce the Kaplan-Meier curves for each treatment. Therefore, in the STRATUM column, there are three values because there are three Kaplan-Meier curves on the plot.

❷ The SCATTERPLOT statement produces the plus-shaped markers which represent the subjects with censored time-points. You can see that the SYMBOL=PLUS option is used within the MARKERATTRS option to create the plus-shaped symbols. Again, the GROUP option is used to display censored time-point values for each of the three treatments.

Program 4, shows how you can create the censored legend. You can use the SCATTERPLOT statement in conjunction with the DISCRETELEGEND statement to produce the censored legend.

#### Program 4: Creating Censored Legend

```

scatterplot x=time y=censored /
  markerattrs=(symbol=plus color=black) } ❶
  name="Censored"; }

discretelegend "Censored" / } ❷
  location = inside
  autoalign = (topright);

```

❶ In the SCATTERPLOT statement, the MARKERATTRS option denotes the attributes which are shown in the legend. You can see that the options SYMBOL = PLUS and COLOR = BLACK are specified, and so these attributes will appear in the legend.

In Figure 1, there are three colors: blue, red, and green. However, instead of having a blue, red, and green plus marker, all that is needed is to have one marker in the legend to indicate the censored value. Therefore, a different color was chosen, that is, COLOR = BLACK, to represent the subjects with censored values across all three treatment groups.

The NAME option is used within the SCATTERPLOT statement to reference this plot statement in subsequent template statements. And it's used in our example so that the particular scatterplot statement can be eventually referenced in the DISCRETELEGEND statement. Hence, the attributes of the markers displayed in the legend will be the attributes that we specified (earlier in the SCATTERPLOT statement).

❷ You can see that the option NAME="CENSORED" is in the SCATTERPLOT statement, and in the DISCRETELEGEND statement, you can also see the value "CENSORED" is being used. That "CENSORED" text is the link between the SCATTERPLOT and the DISCRETELEGEND statement.

Within the DISCRETELEGEND statement, the LOCATION = INSIDE option displays the legend inside the Graph Walls. The AUTOALIGN option displays the legend in the top right-hand corner of the graph.

Program 5 shows you how to produce the subject's at-risk table, which you saw located at the bottom of the graph. The AXISTABLE statement was used to produce the at-risk table.

### Program 5: Creating Subject At-Risk Table

```
layout overlay /  
  xaxisopts=(display=none  
    linearopts=(tickvaluesequence=(start=0  
      end=210 increment=30))) border=off;  
  
  axistable value=atrisk x=tatrisk /  
    class=stratum colorgroup=stratum;  
endlayout;
```



- ❶ In the AXISTABLE statement, you can use the CLASS option so that the subjects at risk are shown on a separate row for each treatment group.

The COLORGROUP=STRATUM option displays different color text for each of the groups in the STRATUM variable. Because the STRATUM variable is used here and the STRATUM variable was used in the GROUP options within the STEP PLOT and SCATTERPLOT statements, the subject at-risk values text will be the same color for each treatment as the color of the KM curves, and the color of the censored values.

You could have produced the AT-RISK table inside the graph, however because it was produced outside the graph the LAYOUT OVERLAY statement is also used.

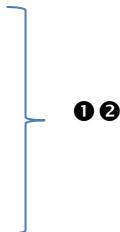
Essentially to produce the graph in Figure 1, a LAYOUT LATTICE with two rows has been used, that is a LAYOUT LATTICE with two LAYOUT OVERLAY statements.

The first LAYOUT OVERLAY contained the SERIESPLOT, SCATTERPLOT and DISCRETELEGEND statements, and the second LAYOUT OVERLAY contained the AXISTABLE statement.

Now you will see how to create the Log-Rank *p*-value, and hazard ratio summary statistics table. You can create those statistics by using MVAR and ENTRY statements as shown in Program 6.

### Program 6: Creating the Log-Rank *p*-value, and Hazard Ratio Summary Statistics Table

```
mvar log_rank_pvalue HazardRatio1 HazardRatio2; ❶  
  
layout gridded / columns=2 rows = 3 border = true  
  halign = right valign = top outerpad=(top=25px);  
  entry halign = right "Log-Rank: "  
    textattrs=(style=italic) "p"  
    textattrs=(style=normal) "-value = ";  
  entry halign = left log_rank_pvalue;  
  <Other Entry Statements>  
endlayout;
```



- ❶ Recall we created three macro variables beforehand, and those macro variables contain the macro variable values of the log-rank *p*-value, the hazard ratio of high dose vs. the placebo, and the hazard ratio of the low dose vs. the placebo.

The MVAR statement generates character tokens from the macro variables.

Then the character tokens can be further referenced in GTL, such as in the subsequent ENTRY statement.

- ❷ In the ENTRY statement, you can also use the TEXTATTRS options to specify the attributes of the text, and therefore we have different attributes for different text as seen in the “*p*” in “*p*-value”. You can see the TEXTATTRS option (STYLE=ITALIC) is used to make the *p* in *p*-value be displayed in an italic font.

Now you will see how to produce the lines which represent the median survival time and display labels that represent the median survival time. You can use the NMVAR, (MVAR) statement, and the DROPLINE statement to create the lines as shown in Program 7.

### Program 7: Creating Lines that Represent the Median Survival Time

```
nmvar MedianSurvival1 MedianSurvival2 MedianSurvival3; ❶  
mvar CMedianSurvival1 CMedianSurvival2 CMedianSurvival3;  
  
%do i = 3 %to 1 %by -1;  
    dropline y = 0.50 x = MedianSurvival&i / ❷  
        dropto = both ❸  
        lineattrs=(thickness=1px  
            color=graphdata&i:color  
            pattern=graphdata&i:linestyle) ❹  
        label=CMedianSurvival&i;  
    %end;
```

❶ The NMVAR statement is similar to the MVAR statement, but it creates numerical tokens instead of character tokens.

You may wonder why a numeric token was created. Numeric tokens were created because the DROPLINE statement, which was subsequently used, expects a numeric value instead of a character value in the X argument.

❷ The DROPLINE statement produces the horizontal and vertical lines representing the median and median survival time. A coordinate is used in the DROPLINE statement, i.e., Y = 0.5, the Median, and X = the Median Survival Time. You can also use macros within GTL, and therefore instead of writing the dropline statement three times, one for each treatment group, a macro is used to produce the three different DROPLINE statements.

❸ When the DROPTO = BOTH option is specified, a line is drawn from the coordinates specified to the Y AXIS, and also drawn from the coordinates to the X AXIS.

❹ The LINEATTRS option can be used to format the line attributes.

## CONCLUSION

Data from a procedure can be saved in ODS output objects to be further used. Creating a custom template and associating it with the necessary data allows you to create custom graphs. It is relatively simple to create Kaplan-Meier plots using SAS. The STEPLOT statement creates the Kaplan-Meier curves. The AXISTABLE statement creates the subjects at risk table, and the ENTRY statements in conjunction with macro variables values can be used to add the appropriate annotation. You can control the color of various items by using options such as LINEATTRS and MARKERATTRS.

## REFERENCES

Harris, K., & Watson, R. (2020). *SAS® Graphics for Clinical Trials by Example*. SAS Institute.

## RECOMMENDED READING

- *SAS® Graphics for Clinical Trials by Kriss Harris and Richann Watson*.

## CONTACT INFORMATION

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

Kriss Harris  
SAS Specialists Limited  
italjet125@yahoo.com

<http://www.krissharris.co.uk>

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

The full code for Figure 1, can be found at the following link:

<https://www.krissharris.co.uk/ATE/KM%20program%20for%20ATE.sas>