

Customizing the Graph Templates for a Kaplan-Meier Failure Plot

Hugh Geary, Novella Clinical, a Quintiles company, Columbus, OH

ABSTRACT

Although PROC LIFETEST in SAS/STAT® 13.1 is generally used to generate Kaplan-Meier Survival Plots it can also generate 'Failure' plots. Survival here denotes the time to an event, such as death, and the plot is descending because subjects are removed from the count as they reach the event. In a failure plot subjects are added to the count as they reach the event and they have failed because they failed to avoid reaching the event (of death). The failure plot is an ascending curve. The failure plot is also used to display the 'Time to Recovery' which is a more positive frame of reference.

For the survival plot we have the option of modifying templates directly or using macros provided in a paper by Warren F. Kuhfeld and Ying So¹. However for the failure plot we must modify the failure plot templates directly. This paper explores modifying the failure templates.

INTRODUCTION

Both the survival and failure curves can be controlled in the same way as far as using the procedure options of the "plots=survival(...)" statement. The options that do apply to both include: the Hall-Wellner and/or equal-precision bands, p-values, and an at-risk counts table. Additionally the ability to place the at-risk table beneath the figure instead of inside the figure, to specify the time points for the at-risk counts independently of the time points for the survival/failure curves, and to specify that the time points must coincide with those specified for the at-risk table. One example could be:

```
proc lifetest data=sashelp.BMT;
    plots=survival(cb=hw test atrisk(atrisktickonly maxlen=13
        outside(0.15))=0, 500, 1000 to 2000 by 200);
    time T * Status(0);
    strata Group;
run;
```

To generate the "Failure" plot another parameter must be added as, plots=survival(failure ...) (that is, 'failure' in addition to any of the afore mentioned parameters of survival).

If we do not use the procedure, plots=survival(), at all, PROC LIFETEST will still run, but will produce only the survival curves with the censor markers and the defaults for the legend, axes labels, spacing for time points, and their labels.

However the template has to be modified if we need to do such things as change the default values for titles, axes labels, time points, labels for the time points, the legend position, the legend label, or construct a custom-designed inset table inside to display other information, such as, counts, medians, and confidence intervals.

For survival plots the templates are available in macro form and can be modified using macro variables. But for failure plots we must get a copy of the template, modify it, store it temporarily, run PROC LIFETEST, and then delete it.

There are two graph templates for the failure option of PROC LIFETEST, one for having the at-risk counts displayed within the figure and the other for displaying them beneath the horizontal axis label. A copy of either can be customized, stored in one's own project folder, and then used in place of the originals. There are no macros known (to the author) to be available for easily modifying the failure templates. But terms within these locally stored templates can be replaced by macro variables and even sections of a template can be rewritten in order to create modified templates. These modified versions can then be %included ahead of the call to PROC LIFETEST.

The data used here are in sashelp.BMT and come from a study of bone marrow transplant patients. There are three groups as specified by the variable, **Group**: ALL (acute lymphoblastic leukemia), AML (acute myelocytic leukemia)-Low Risk, and AML-High Risk. The variable, **T**, is time in days until death, relapse, or end of the study.

CREATION OF THE LOCAL TEMPLATE

To obtain a copy of the template we need to run PROC LIFETEST with trace on and with the parameters that we expect to use:

Customizing the Graph Templates for a Kaplan-Meier Failure Plot, continued

```
ods graphics on;
ods trace output;
proc lifetest data=sashelp.BMT plots=survival(cb=hw failure test atrisk(maxlen=13));
  time T * Status(0);
  strata Group / order=internal;
run;
ods trace off;
```

The trace output will appear in the log:

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

This is the exact name of the template that PROC LIFESTEST will look for to use whenever it is run. If 'outside' had been a parameter, such as in,

```
proc lifetest data=sashelp.BMT plots=survival(failure cb=hw test atrisk
(maxlen=13 outside((0.15))=0 to 2000 by 200);
```

then the other template's name would have appeared as:

```
Template:  Stat.Lifetest.Graphics.ProductLimitFailure2
```

To obtain the PROC TEMPLATE code, run the following. The code will appear in the log.

```
proc template;
  source Stat.Lifetest.Graphics.ProductLimitFailure;
run;
```

In order to modify the failure plot, copy the template code from the log into a file and insert

```
proc template;
```

as the first statement of the template. Modify the code as needed and then %INCLUDE it into the program ahead of the PROC LIFESTEST failure statement and run the program.

When finished with running with this version of the template, this modified version of the template needs to be removed from the system because it has the same name as the system version. Use the following code to delete it.

```
proc template;
  delete Stat.Lifetest.Graphics.ProductLimitFailure;
run;
```

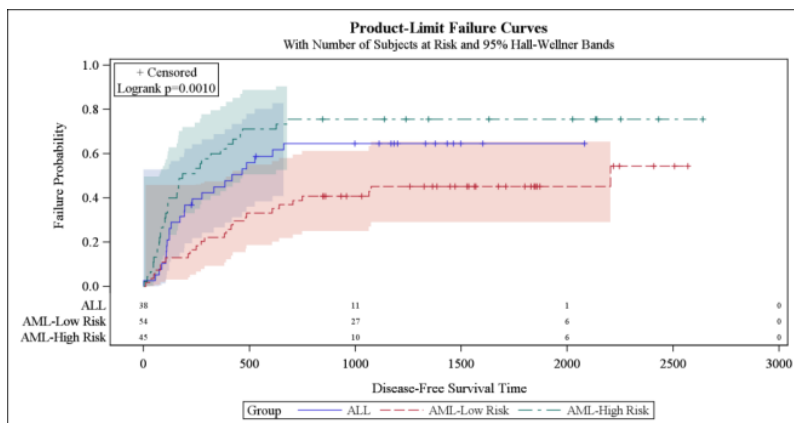


Figure 1 A Kaplan-Meier Failure Plot before making any modifications

EXAMPLES OF TEMPLATE MODIFICATIONS

Next we examine specific lines of the template code and indicate how they may be modified to change a feature of the output figure.

The template is complex. To decipher it helps to note that roughly the first half of the code is for plots that have a single strata and the 2nd half is for multiple overlaid strata. Nearly every term that appears in one half also appears in the other half. If it is known that all applications for a given project will have two or more strata then the first half of the code that is between the two statements can be eliminated provided it is done precisely so as to not cause a syntax error.

```
if (NSTRATA=1) and
else           (which occurs about half through the code)
```

TITLE LINES FOR THE PLOT

The first few lines of the template code are:

```
proc template;
define statgraph Stat.Lifetest.Graphics.ProductLimitFailure;
dynamic NStrata xName maxTime plotAtRisk plotCensored plotCL plotHW plotEP
labelCL labelHW labelEP yMin xtickVals xtickValFitPol method StratumID
classAtRisk plotTest GroupName Transparency SecondTitle TestName pValue
_byline_ _bytitle_ _byfootnote_;
BeginGraph;    ** The code between the next two lines has been omitted **;
if (NSTRATA=1)
                (single strata code omitted here)
else
  entrytitle METHOD " " "Failure Curves";
  if (EXISTS(SECONDTITLE))
    entrytitle SECONDTITLE / textattrs=GRAPHVALUETEXT;
endif;
```

Values for the dynamic variables are supplied by PROC LIFETEST. Next are the entrytitle lines. The title in Figure 1 can be replaced by replacing “METHOD ” “Failure Curves” with, say, “A Kaplan-Meier Plot for BMT Survival Analysis” so that the entrytitle statement would be:

```
entrytitle "A Kaplan-Meier Plot for BMT Survival Analysis";
```

VERTICAL AXIS LABEL AND TICK MARKS MODIFICATIONS

In Figure 1 the label on the vertical axis is “Failure Probability” and in the original template there is a section which has ‘Failure Probability’ in it:

```
layout overlay / xaxisopts=(shortlabel=XNAME offsetmin=.05
  linearopts=(viewmax=MAXTIME tickvaluelist=XTICKVALS
    tickvaluefitpolicy=XTICKVALFITPOL))
yaxisopts=(label="Failure Probability" shortlabel="Failure"
  linearopts=(viewmin=0 viewmax=1 tickvaluelist=(0 .2 .4 .6 .8 1.0)));
```

If this template is to be used for many figures and it is desired to have various labels on this axis, the text “Failure Probability” could be replaced by a macro variable, YY_label, and which could be assigned a value ahead of each execution of PROC LIFETEST, as below.

Also further assume that every failure plot that is going to be produced will need to have the y-axis tic marks at 0, .25, .5, .75 and 1. In this case the list, (0 .2 .4 .6 .8 1.0), can be replaced directly in the template as below.

```
layout overlay / xaxisopts=(shortlabel=XNAME offsetmin=.05
  linearopts=(viewmax=MAXTIME tickvaluelist=XTICKVALS
    tickvaluefitpolicy=XTICKVALFITPOL))
yaxisopts=(label=&YY_label shortlabel="Failure"
  linearopts=(viewmin=0 viewmax=1 tickvaluelist=(0 .25 .5 .75 1.0)));
```

SURVIVAL CURVE and CENSOR MARKERS

Below the statements for Hall-Wellner (PLOTBW=1) and Exact Precision (PLOTPE=1) and others, is the STEPPLOT statement for the survival curve, followed by a SCATTERPLOT statement for the censoring markers.

```

stepplot y=eval (1-SURVIVAL) x=TIME / group=STRATUM index=STRATUMNUM name="Failure"
           rolename=(_tip1=ATRISK _tip2=EVENT) tiplabel=(y="Failure Probability" _tip1=
           "Number at Risk" _tip2="Observed Events") tip=(x y _tip1 _tip2);
if (PLOTCEM)
  scatterplot y=eval (1-CENSORED) x=TIME / tiplabel=(y="Failure Probability")
             group=STRATUM index=STRATUMNUM markerattrs=(symbol=plus);
endif;

```

Note that the scatterplot is conditionally executed. The plus symbol is specified here to mark censored points. The name="Failure" parameter in the stepplot statement links the Legend to it. The Legend must also use "Failure" as seen in the legend statement below or there will be no legend displayed.

LEGEND ATTRIBUTES MODIFICATIONS

In Figure 1 the legend is below the figure as specified by the location parameter, "location=outside".

```
DiscreteLegend "Failure" / title=GROUPNAME location=outside;
```

It can be placed inside at the lower right corner and the title removed (or replaced) by using the following:

```
DiscreteLegend "Failure" / title='The Legend' location=inside across=1
  autoalign=(bottomright topright center);
```

The title in the legend is by default the label of the group variable and the title can be modified or completely removed as done above. With the legend being moved inside a location may be chosen for it. Position possibilities are: topright, right, topleft, left, center, etc. Autoalign will choose the first entry unless it determines that position is already too busy. If there is not enough space to have it in one horizontal line it may be laid out vertically by specifying one column wide by using: across=1 (or n columns wide by using across=n);

INSERTION OF TEXT WITHIN THE FIGURE

Beneath the legend statement are several layout areas as follows:

```

if (PLOTCEM)
  if (PLOTTEST)
    Layout code (to display information using entry statements)
  else
    Layout code
  endif;
else
  if (PLOTTEST)
    Layout code
  else
    Layout code
  endif;
endif;

```

Entry statements can be added into a layout-endlayout section such as:

```

if (PLOTTEST)
  layout gridded / rows=2 autoalign=(TOPLEFT BOTTOMRIGHT BOTTOM TOP)
  border=true BackgroundColor=GraphWalls:Color Opaque=true;
  entry "+ Censored";
  if (PVALUE < .0001) entry TESTNAME " p " eval (PUT(PVALUE, PVALUE6.4));
  else
    entry TESTNAME " p=" eval (PUT(PVALUE, PVALUE6.4));
  endif;
  entry " This is additional verbiage";
  entry " for illustrative purposes";
endlayout;

```

Additionally other layout code can be stuck in between other layouts. Note that the autoalign parameters indicate that the preferred location is TopRight, the second is Bottom, etc.

```

layout gridded / rows=1 autoalign=(TOPRIGHT BOTTOM TOP)
border=true BackgroundColor=GraphWalls:Color Opaque=true;
entry "This is a completely ";
entry " new layout added ";
entry " beneath another layout";
endlayout;

```

Another inset has been placed near the bottom of the code and above the last endlayout statement.

```

layout gridded / rows=1 autoalign=(BOTTOM LEFT TOP) border=true
BackgroundColor=GraphWalls:Color Opaque=true;
entry " This inset comes from ";
entry " Layout code directly above";
entry " the last endlayout statement";
endlayout;
endlayout;
endif;

```

The PLOTCESTORED route is taken if there is censored data and the PLOTTEST route is taken if TEST has been specified as a parameter of plots=survival(). So one of these layouts will be executed. Additional entry statements or even replacement statements can be put in a layout section.

Finally Run PROC LIFETEST with the Included Template to Produce FIGURE 2

Include the code from the template.

```

%let YY_label=Percentage Failing;
%let Gpath=U:\_SUG\Graphs;
%include "&Gpath.\Template_Failure_Pharma_SUG.sas";

```

Run PROC LIFETEST now with the modified template .

```

ods graphics on;

proc lifetest data=sashelp.BMT plots=survival(cb=hw failure
test atrisk(maxlen=13 ));
time T * Status(0);
strata Group / order=internal;
run;
ods rtf close;
ods graphics off;

```

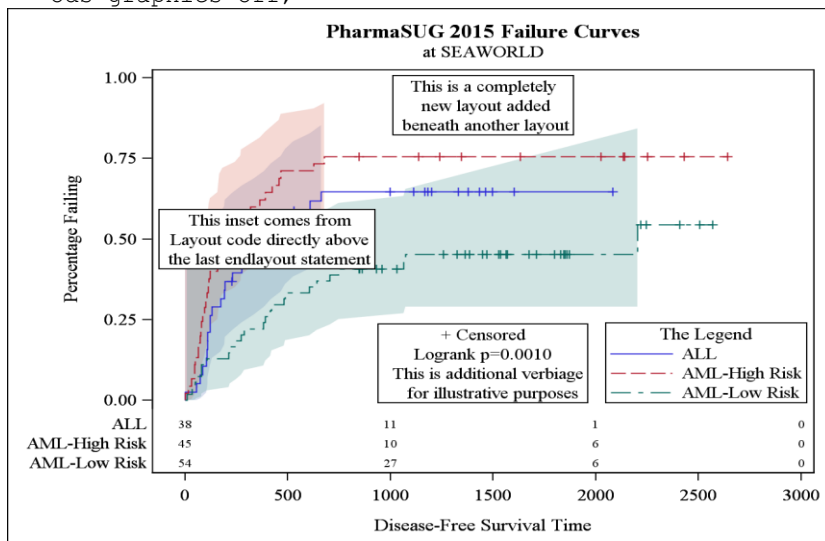


Figure 2. A Kaplan-Meier Failure Plot displaying the modifications

CONCLUSION

When the 'Failure' option is taken PROC LIFETEST does produce a basic general output figure. But it will also produce the source code for the failure template which we can store and then modify internally with specific code changes or through use of user-installed macro variables. The modified template can then be used by PROC LIFETEST to produce the desired outputs.

APPENDIX

The Modified PROC TEMPLATE Source Code

```
define statgraph Stat.Lifetest.Graphics.ProductLimitFailure;
  dynamic NStrata xName maxTime plotAtRisk plotCensored plotCL plotHW plotEP
    labelCL labelHW labelEP yMin xtickVals xtickValFitPol method StratumID
    classAtRisk plotTest GroupName Transparency SecondTitle TestName pValue
    _byline_ _bytitle_ _byfootnote_;
  BeginGraph;
    * < Below - The code for a single strata has been omitted > ;
    * < Thus the "else" refers to a multiple curve overlay > ;
  if (NSTRATA=1)
  else
    entrytitle "PharmaSUG 2015" " " "Failure Curves";
    if (EXISTS(SECONDTITLE))
      entrytitle "at SEAWORLD" / textattrs=GRAPHVALUETEXT;
    endif;
  layout overlay / xaxisopts=(shortlabel=XNAME offsetmin=.05
    linearopts=(viewmax=MAXTIME tickvaluelist=XTICKVALS
    tickvaluefitpolicy=XTICKVALFITPOL))
    yaxisopts=(label="&YY_Label" shortlabel="Failure"
    linearopts=(viewmin=0 viewmax=1 tickvaluelist=(0 .25 .5 .75 1.0)));

  if (PLOTBW=1)
    bandplot LimitUpper=eval (1-HW_LCL) LimitLower=eval (1-HW_UCL)
      x=TIME / group=STRATUM index=STRATUMNUM modelname="Failure"
      datatransparency=Transparency;
  endif;
  if (PLOTBW=1)
    bandplot LimitUpper=eval (1-EP_LCL) LimitLower=eval (1-EP_UCL)
      x=TIME / group=STRATUM index=STRATUMNUM modelname="Failure"
      datatransparency=Transparency;
  endif;
  if (PLOTCL=1)
    if (PLOTBW=1 OR PLOTBW=1)
      bandplot LimitUpper=eval (1-SDF_LCL) LimitLower=eval (1-SDF_UCL)
        x=TIME / group=STRATUM index=STRATUMNUM modelname="Failure"
        display=(outline);
    else
      bandplot LimitUpper=eval (1-SDF_UCL) LimitLower=eval (1-SDF_LCL)
        x=TIME / group=STRATUM index=STRATUMNUM modelname="Failure"
        datatransparency=Transparency;
    endif;
  endif;
  stepplot y=eval (1-SURVIVAL) x=TIME / group=STRATUM index=STRATUMNUM
    name="FailureZ" rolename=( _tip1=ATRISK _tip2=EVENT)
    tiplabel=(y="Failure Probability" _tip1="Number at Risk"
    _tip2="Observed Events") tip=(x y _tip1 _tip2);
  if (PLOTCESTORED)
    scatterplot y=eval (1-CENSORED) x=TIME / tiplabel=(y="Failure
    Probability") group=STRATUM index=STRATUMNUM
    markerattrs=(symbol=plus);
```

Customizing the Graph Templates for a Kaplan-Meier Failure Plot, continued

```

endif;
if (PLOTATRISK)
  innermargin / align=bottom;
  blockplot x=TATRISK block=ATRISK / class=CLASSATRISK
  repeatedvalues=true display=(label values) valuehalign=start
  valuefitpolicy=truncate labelposition=left
  labelattrs=GRAPHVALUETEXT valueattrs=GRAPHDATATEXT (size=7pt)
  includemissingclass=false;
endinnermargin;
endif;
DiscreteLegend "FailureZ" / title="The Legend" location=inside
  across=1 autoalign=(bottomright topright center);
* DiscreteLegend "Failure" / title=GROUPNAME location=outside;
if (PLOTCSOURED)
  if (PLOTTEST)
    layout gridded / rows=2 autoalign=(TOPLEFT BOTTOMRIGHT BOTTOM TOP)
    border=true BackgroundColor=GraphWalls:Color Opaque=true;
    entry "+ Censored";
    if (PVALUE < .0001)
      entry TESTNAME " p " eval (PUT(PVALUE, PVALUE6.4));
    else
      entry TESTNAME " p=" eval (PUT(PVALUE, PVALUE6.4));
    endif;
    entry " This is additional verbiage";
    entry " for illustrative purposes";
  endlayout;
  layout gridded / rows=1 autoalign=(TOPRIGHT BOTTOM TOP) border=true
  BackgroundColor=GraphWalls:Color Opaque=true;
  entry "This is a completely ";
  entry " new layout added ";
  entry " beneath another layout";
  endlayout;
else
  layout gridded / rows=1 autoalign=(TOPLEFT BOTTOMRIGHT TOP) border=true
  BackgroundColor=GraphWalls:Color Opaque=true;
  entry "+ Censored";
  endlayout;
endif;
else
  if (PLOTTEST)
    layout gridded / rows=1 autoalign=(TOPLEFT BOTTOMRIGHT BOTTOM TOP)
    border=true BackgroundColor=GraphWalls:Color Opaque=true;
    if (PVALUE < .0001)
      entry TESTNAME " p " eval (PUT(PVALUE, PVALUE6.4));
    else
      entry TESTNAME " p=" eval (PUT(PVALUE, PVALUE6.4));
    endif;
    endlayout;
  endif;
endif;
layout gridded / rows=1 autoalign=(BOTTOM LEFT TOP) border=true
  BackgroundColor=GraphWalls:Color Opaque=true;
  entry " This inset comes from ";
  entry " Layout code directly above";
  entry " the last endlayout statement";
  endlayout;
endlayout;
endif;
if (_BYTITLE_)
  entrytitle _BYLINE_ / textattrs=GRAPHVALUETEXT;
else
  if (_BYFOOTNOTE_)

```

```
        entryfootnote halign=left _BYLINE_ ;
    endif;
endif;
EndGraph;
end;
```

REFERENCES

¹ Warren F. Kuhfeld and Ying So . "Creating and Customizing the Kaplan-Meier Survival Plot in PROC LIFETEST in the SAS/STAT® 13.1 Release." *Proceedings of the PharmaSUG 2014 Conference*>. Cary,NC .

CONTACT INFORMATION

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

Name: N. Hugh Geary
Enterprise: Novella Clinical, a Quintiles company
Address: 580 N 4th Street
City, State ZIP: Columbus, OH 43215
Work Phone: 614 572-0250
E-mail: hgeary@novellaclinical.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.