

Kaplan-Meier Survival Plotting Macro %NEWSURV

Jeffrey Meyers, Mayo Clinic, Rochester, Minnesota

1.0 ABSTRACT

The research areas of pharmaceuticals and oncology clinical trials greatly depend on time-to-event endpoints such as overall survival and progression-free survival. One of the best graphical displays of these analyses is the Kaplan-Meier curve, which can be simple to generate with the LIFETEST procedure but difficult to customize. Journal articles generally prefer that statistics such as median time-to-event, number of patients, and time-point event-free rate estimates be displayed within the graphic itself, and this was previously difficult to do without an external program such as Microsoft Excel. The macro NEWSURV takes advantage of the Graph Template Language (GTL) that was added with the SG graphics engine to create this level of customizability without the need for backend manipulation. Taking this one step further, the macro was improved to be able to generate a lattice of multiple unique Kaplan-Meier curves for side by side comparisons or condensing figures for publications. The following is a paper describing the functionality of the macro and a description of how the key elements of the macro work.

2.0 INTRODUCTION

The research areas of pharmaceuticals and oncology clinical trials greatly depend on time-to-event endpoints such as overall survival and progression-free survival. The standard graphical display of these analyses is the Kaplan-Meier curve, which can be simple to generate with the LIFETEST procedure but difficult to customize. My office has made use of Microsoft Excel to produce their customized Kaplan-Meier curves, but the process to create and customize these plots is tedious and time-consuming. Generally statistics such as median time-to-event, number of patients, number of events, and other statistics are manually added to these plots for publications with text boxes. This means the entire process of exporting data to Excel, making necessary modifications to the plot colors, thicknesses, titles, and labels, and adding text boxes for statistics must be repeated for each data update. However with the SAS[®] Graph Template Language (GTL) it has become possible to create highly customized plots without the need to manually modify on the back end. Taking this customizability and incorporating it into a macro has led to a highly efficient and powerful tool for displaying Kaplan-Meier analyses. SAS keywords will be shown in all capital letters, and macro parameters will be shown in italic capital letters. The terms used in the paper are described in further detail in appendix one, and all parameters for the macro are listed in the documentation section of the macro code which is uploaded as an attachment to this paper or is available upon request (see contact information).

2.1 SAMPLE DATASET USED IN EXAMPLES

All of the examples in this paper use the SASHELP.BMT data set, which is only available in SAS 9.3 or later. The data set contains survival data on bone marrow transplant patients and has three variables: GROUP, T, and STATUS. The GROUP variable is a discrete categorical variable containing three different disease groups: ALL, AML-High Risk, and AML-Low Risk. The T variable is the time from transplant, and the STATUS variable is the survival status (0=Alive, 1=Dead).

Table 1 shows the first several rows of the SASHELP.BMT data set.

Group	T	Status
ALL	2081	0
ALL	1602	0
ALL	1496	0
ALL	1462	0
ALL	1433	0
ALL	1377	0
ALL	1330	0
ALL	996	0
ALL	226	0
ALL	1199	0
ALL	1111	0
ALL	530	0
ALL	1182	0
ALL	1167	0

Table 1. The GROUP variable is a discrete categorical character variable with three levels: ALL, AML-High Risk, and AML-Low Risk. The T variable is a numeric variable representing time from transplant. The STATUS variable is a numeric variable representing survival status where 0 means alive and 1 means dead.

3.0 MACRO OVERVIEW

The macro, *NEWSURV*, is designed to easily display Kaplan-Meier curves while also displaying common statistics associated with time-to-event analyses such as the median time-to-event, the hazard ratio from a Cox proportional hazards model, time-point event-free rates (i.e. Kaplan-Meier estimates), and number of patients-at-risk. The macro also produces a summary table outside of the plot image using the *REPORT* procedure from the same macro call that can contain the patient counts and events, median time-to-event, time-point event-free rates, p-value, and the hazard ratio from a Cox proportional hazards model. The macro is written in a way that the plot can be as simple or as customized as the user intends. There are only four required inputs for the macro: *DATA*, *TIME*, *CENS*, and *CEN_VL*. These components refer to the data set name, time variable name, censor or event variable name, and the value representing a censor within the censor or event variable respectively. Outside of these required parameters, nearly every line, text, and layout is customizable with optional macro parameters. These outputs can be sent to a PDF, RTF, HTML or be included within Output Delivery System (ODS) tags outside of the macro call, and the ability to enable scalable vector graphics is also an option in the macro (SAS 9.3 or later only).

3.1 PLOT OVERVIEW

The plots can be called with or without a *CLASS* input. Plot components that are able to be customized include line colors, thicknesses, and patterns; text fonts and sizes; and axes. The statistics displayed as a summary table within the plot can be turned on or off individually. The reference group for the hazard ratio can be defined by the user, the p-value displayed in the plot is one of four options, and multiple time-point event-free rate estimates can be requested. Patients-at-risk numbers can be displayed in three different ways with or without color matching the plot lines. One of the strongest features of this macro is that it can also generate images of a lattice of plots instead of one single plot. Any number of models (i.e. overall survival, progression-free survival, subgroup analyses, and association analyses with or without adjusting covariates) can be input into one macro call and they can be displayed in any rectangular array of plots. Each model is customized separately in the macro call.

Figure 1 is an example plot image showing a number of the customization options available in *NEWSURV*

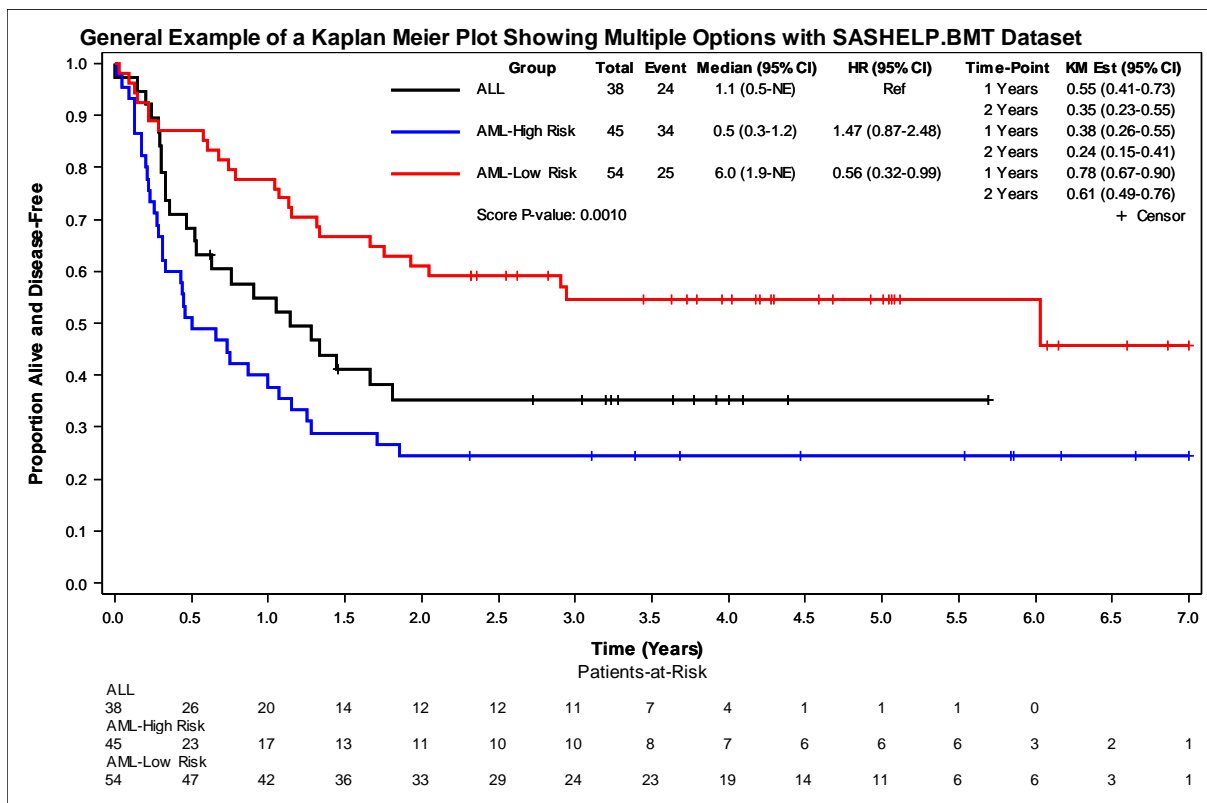


Figure 1. An example showing *CLASS* option, a patients-at-risk table, axis options, and a plot summary table.

```
%newsurv(data=sashelp.bmt,time=T,cens=status,cen_vl=0,
class=group,classref=ALL,classdesc=Group,classvalalign=left,
risklabellocation=above,risklocation=bottom,risklist=0 to 7 by 0.5,
timelist=1 to 2 by 1,timedx=Years,
xmin=0,xmax=7,xincrement=0.5,xdivisor=365.25,ytype=ppt,
xlabel=Time (Years),ylabel=Proportion Alive and Disease-Free,rows=1,
```

```

plot=1,summary=0,
width=9in,height=6in,symbolsize=6pt,linesize=2pt,color=BLACK BLUE RED,
title=General Example of a Kaplan Meier Plot Showing Multiple Options with
SASHELP.BMT Dataset,
gpath=~ /ibm/,plotname=figure1,plottype=emf);

```

3.2 STATISTICAL REPORT OVERVIEW

A report outside of the plot can display the number of patients, number of events, median time-to-event, hazard ratios, event-free rate estimates, and p-values. Each column of the report can be turned on or off, can have the width modified, and all fonts and font sizes of the values are modifiable. A title can be added to the start of the report, and footnotes can be added to the end of the report. Separate macro calls are able to output to the same statistical report table.

Table is an example of a statistical report table:

General Statistical Report Table Example

	Event/Total	Median (95% CI) [†]	Hazard Ratio (95% CI) [‡]	Survival Estimates (95% CI) [†]	P-value
SASHELP.BMT Data set with no Class Variable					
All Patients	83/137	1.3 (1.0-2.9)		1 Years: 58.3 (50.6-67.2) 2 Years: 42.0 (34.5-51.2)	

This is the footnote for this model

SASHELP.BMT Data set with Class Variable

Group	Event/Total	Median (95% CI)	Hazard Ratio (95% CI)	Survival Estimates (95% CI)	P-value
ALL	24/38	1.1 (0.5-NE)	Ref	1 Years: 54.9 (41.1-73.4) 2 Years: 35.3 (22.7-54.8)	0.0010 [§]
AML-High Risk	34/45	0.5 (0.3-1.2)	1.47 (0.87-2.48)	1 Years: 37.8 (26.0-55.0) 2 Years: 24.4 (14.6-40.9)	
AML-Low Risk	25/54	6.0 (1.9-NE)	0.56 (0.32-0.99)	1 Years: 77.8 (67.4-89.7) 2 Years: 61.1 (49.4-75.6)	

This is the footnote for this model

[†]Kaplan-Meier method; [‡]Cox model; [§]Score test;

This also allows an overall footnote

Table 2. Statistical report table output from a multiple model macro call using the SASHELP.BMT data set.

```

%newsurv(data=sashelp.bmt,nmodels=2,time=T,cens=status,cen_vl=0,
class=|group,classref=ALL,classdesc=All Patients|Group,
timelist=1 to 2 by 1, timedx=Years,xdivisor=365.25,
summary=1,plot=0,destination=rtf,
title=SASHELP.BMT Dataset with no Class Variable|
SASHELP.BMT Dataset with Class Variable,
footnote=This is the footnote for this model,
outdoc=~ /ibm/MWSUG_second_table.rtf,
tabletitle=General Statistical Report Table Example,
tablefootnote=This also allows an overall footnote);

```

4.0 COMPUTED STATISTICS

4.1 DATA SET TRANSFORMATIONS

A temporary data set is generated from the input data set for each model that is run by the macro. These temporary data sets are then able to be transformed or manipulated without affecting the input data set. There are four types of data set transformations the macro can do: CLASS variable transformation, time variable transformation, where clause, and landmarking. These temporary data sets are then deleted at the conclusion of the macro.

4.1.1 CLASS Variable Transformation

In order to make the programming for the macro simpler, the *CLASS* variable is recoded to a character variable whether it was originally a numeric or a character variable. This is done by saving the variable's format into a macro variable in a `_NULL_` data set with a `CALL SYMPUT` function combined with the `VFORMAT` function and then doing a `PUT` function with the variable and its format. If there is no format associated with the *CLASS* variable then the default for numeric variable is the `BEST12` format and the default for a character variable is the length of the variable.

4.1.2 Time Variable Transformation

The inputted time variable is normally displayed exactly as it is in the input dataset. A macro parameter *XDIVISOR* allows the time variable to be transformed into other time units. For example, if the time variable is currently in days then it can be transformed into years by setting *XDIVISOR* to be 365.25. This is done before any of the analysis procedures run in order to allow the user to specify other parameters such as *TIMELIST* and *RISKLIST* in the transformed units instead of the original units.

4.1.3 Where Clause

The *WHERE* macro parameter allows the user to set a unique where clause for each model they wish to run within the macro. The where clause is applied when the temporary data set is created. This is useful for subgroup analysis. An example could be comparing treatment effect within males versus treatment effect within females. Both models would be run with the same *CLASS* variable, but with separate *WHERE* values.

```
%newsurv(data=example,time=T,cens=status,cen_v1=0,
class=treatment,classref=A,xdivisor=365.25,
summary=0,destination=rtf,
title=Treatment Effect within Males|
Treatment Effect within Females,
where=gender='Male'|gender='Female',rows=2,nmodels=2);
```

4.1.4 Landmarking

The time variable can be landmarked at either a pre-specified time or by a numeric variable using the same macro parameter *LANDMARK* in order to perform landmarking analyses. If *LANDMARK* is set to a numeric value, then any times that are less than or equal to this value are set to missing along with their event variables. Other times have their values subtracted by the landmarked value. If *LANDMARK* is set to a variable, then this same comparison is made by comparing the *TIME* variable against the *LANDMARK* variable.

4.2 STATISTICS AVAILABLE FOR EVERY MODEL

The statistics for the plot summary table are generated from one `PROC LIFETEST` and up to two `PHREG` procedures (one when a *CLASS* variable is included, one when a multivariate model is requested).

4.2.1 Patient Numbers

The number of patients and the number of events are pulled from `PROC LIFETEST` using the `ODS OUTPUT` table `CENSOREDSSUMMARY`. This data set contains the number of patients, the number of censors, and the number of events for the overall population and by *CLASS* variable levels.

4.2.2 Survival Estimates

The Kaplan-Meier survival estimates and times are taken from `PROC LIFETEST` and saved into a data set using the `PLOT` option with `ODS GRAPHICS` set to `ON` and output with the `ODS OUTPUT` table `SURVIVALPLOT`. This data set contains the survival times, survival function estimates, censor indicators, and *CLASS* levels.

4.2.3 Median Time-to-Event

The median time-to-event values are pulled from a `PROC LIFETEST` using the `ODS OUTPUT` table `QUARTILES`. This dataset contains the median time-to-events along with the first and third quartile time-to-events, and for each of these quartiles a 95% confidence interval is included. The macro parameter *CONFTYPE* determines which method to use for the confidence intervals with the default being `LOG`.

4.2.4 Event-Free Rate Estimates

The Kaplan-Meier event-free rate estimates are pulled from a `PROC LIFETEST` using the `OUTSURV` option in conjunction with the `TIMELIST` option and `REDUCEOUT` option. This reduces the output data set to just the requested time-points, and includes the event-free rates along with 95% confidence intervals. The macro parameter *CONFTYPE* determines which method to use for the confidence intervals with the default being `LOG`.

4.2.5 Censor Values

The values are created by taking the survival estimates from the patient survival estimates table where the values are marked as a censor.

4.2.6 Patients-at-Risk Numbers

The patients-at-risk list is pulled from PROC LIFETEST using the ATRISK option within the PLOT option with ODS GRAPHICS set to ON and output with the ODS OUTPUT table SURVIVALPLOT. This data set contains the number of patients at risk at each event-free rate requested in the macro call.

4.3 STATISTICS CALCULATED ONLY WHEN CLASS VARIABLE SPECIFIED

4.3.1 Cox Proportional Hazard Ratios

4.3.1.1 Univariate

The Cox proportional hazard ratio is pulled from the PHREG procedure using the ODS OUTPUT table PARAMETERESTIMATES. The *CLASS* input variable is included in the CLASS statement within the procedure, and the reference value is customizable. The RISKLIMITS option in the model statement is used to generate the 95% confidence interval. The method for handling ties is customizable, and a stratification variable can also be added to the model.

4.3.1.2 Multivariate

A second set of Cox proportional hazard ratios can be pulled from the second PHREG procedure by using the *CLASSCOV* or *CONTCOV* macro parameters to create a multivariate model. Discrete adjusting variables are specified with the *CLASSCOV* parameter and are included in the CLASS statement and MODEL statement. Continuous adjusting covariates are specified with the *CONTCOV* macro parameter and are only included in the MODEL statement. None of the hazard ratios for adjusting variables are kept in the outputted data set and they are not displayed in the plot summary table. The method for handling ties is customizable, and a stratification variable can also be added to the model. The hazard ratios from the multivariate model are displayed separately from the hazard ratios from the univariate model.

4.3.2 P-values

4.3.2.1 Univariate

There are five different p-values that the macro can pull from the univariate model. The first is a type III score p-value from a PROC PHREG using the ODS OUTPUT table TYPE3. This is also where the second and third options, likelihood-ratio and Wald, are pulled from. The fourth option, log-rank, is pulled from a PROC LIFETEST using the ODS OUTPUT table HOMTESTS. HOMTESTS is also where the fifth option, Wilcoxon, is pulled from.

4.3.1.2 Multivariate

There are three different p-values that the macro can pull from the multivariate model. The first is a type III score p-value from a PROC PHREG using the ODS OUTPUT table TYPE3. This is also where the second and third options, likelihood-ratio and Wald, are pulled from. The Type 3 p-values are only pulled for the *CLASS* macro parameter.

5.0 DESIGNING THE PLOT

5.1 CREATING THE PLOT DATA SET

One of the greatest benefits of the Graph Template Language and the SGRENDER procedure is that the variables in the supplied data set do not need to have any relation to each other in order to be used in plot statements. The template created with the Graph Template Language can refer to any number of variables in the supplied data set for any number of different plots. An example used in this macro is to plot a step plot for the Kaplan-Meier curves and a scatterplot for the censor values. The variables for the step plot are different than and will have many more observations than the variables for the scatterplot, but as long as these variables are grouped into the same data set there will not be any issues when referring to these variables with the Graph Template Language.

Even when plotting within the same plot window, any number of plot statements can be called assuming they all use the same type of axis in the plot window. These plot statements are rendered in the order that they are coded; for example, if the same variables are used for two different SCATTERPLOT statements. They are listed with the same symbol and symbol size, but the first is set to black and the second is set to blue. Because the blue scatterplot is rendered second, the black scatterplot will not be visible because the blue scatterplot completely overwrote it. This exact method is used in this macro to generate the legend for the censor markers. The greatest way the macro takes advantage of being able to use multiple plot statements is that each *CLASS* variable level is plotted separately. This is for the following reasons:

- This avoids needing to change the template for grouped data plot attributes or to add an attribute map because each plot statement has its own line attributes.
- Each plot statement has its own NAME. This sets up the statistics table addressed in a later section by giving each CLASS variable level its own legend name option.
- There is no difference in the appearance of the plot.

In order to plot each CLASS variable level separately, the plot data set does not include one set of variables with a CLASS variable variable, time variable, and survival estimate variable, but instead creates a set of time and survival estimate variables for each CLASS variable level. Each of these sets is then used in their own plot statements. The same is done for censor values and patients-at-risk numbers. The final data set looks like a messy mixture of variables, but each variable serves a specific purpose for specific plot statement calls.

Table 3 shows the first half of the completed plot dataset from the macro call used in figure 1

cl1_1	t1_1	c1_1	s1_1	cl2_1	t2_1	c2_1	s2_1	cl3_1	t3_1	c3_1	s3_1
ALL	0.0000		1.0000	AML-High Risk	0.0000		1.0000	AML-Low Risk	0.0000		1.0000
ALL	0.0027		0.9737	AML-High Risk	0.0055		0.9778	AML-Low Risk	0.0274		0.9815
ALL	0.1506		0.9474	AML-High Risk	0.0438		0.9556	AML-Low Risk	0.0958		0.9630
ALL	0.2026		0.9211	AML-High Risk	0.0876		0.9333	AML-Low Risk	0.1314		0.9444
ALL	0.2355		0.8947	AML-High Risk	0.1287		0.8889	AML-Low Risk	0.1451		0.9259
ALL	0.2847		0.8684	AML-High Risk	0.1314		0.8667	AML-Low Risk	0.2163		0.9074
ALL	0.2930		0.8421	AML-High Risk	0.1725		0.8444	AML-Low Risk	0.2190		0.8889
ALL	0.2984		0.8158	AML-High Risk	0.1752		0.8222	AML-Low Risk	0.2875		0.8704
ALL	0.3012		0.7895	AML-High Risk	0.2026		0.8000	AML-Low Risk	0.5777		0.8519
ALL	0.3340		0.7368	AML-High Risk	0.2081		0.7778	AML-Low Risk	0.5996		0.8333
ALL	0.3532		0.7105	AML-High Risk	0.2190		0.7556	AML-Low Risk	0.6790		0.8148
ALL	0.4709		0.6842	AML-High Risk	0.2300		0.7333	AML-Low Risk	0.7447		0.7963
ALL	0.5257		0.6579	AML-High Risk	0.2546		0.7111	AML-Low Risk	0.7885		0.7778
ALL	0.5311		0.6316	AML-High Risk	0.2738		0.6889	AML-Low Risk	1.0431		0.7593
ALL	0.6188	0.6316	0.6316	AML-High Risk	0.2875		0.6667	AML-Low Risk	1.0678		0.7407

Table 3. The cl#_#, t#_#, s#_#, and c#_# variables are the CLASS variable label, time, survival estimate, and censor value variables respectively. The # before the underscore represents the CLASS variable value order, and the # after the underscore represents the model number in the case of a lattice.

Table 4 shows the second half of the completed plot dataset from the macro call used in figure 1

time1_1	tAtRisk	time2_1	time3_1	partitle_1	atrisk1_1	atrisk2_1	atrisk3_1
0	0	0	0	Patients-at-Risk	38	45	54
0.5	0.5	0.5	0.5		26	23	47
1	1	1	1		20	17	42
1.5	1.5	1.5	1.5		14	13	36
2	2	2	2		12	11	33
2.5	2.5	2.5	2.5		12	10	29
3	3	3	3		11	10	24
3.5	3.5	3.5	3.5		7	8	23
4	4	4	4		4	7	19
4.5	4.5	4.5	4.5		1	6	14
5	5	5	5		1	6	11
5.5	5.5	5.5	5.5		1	6	6
6	6	6	6		0	3	6
	6.5	6.5	6.5			2	3
	7	7	7			1	1

Table 4. The time#_# and atrisk#_# variables are the patients-at-risk table time and patients-at-risk numbers respectively. The # before the underscore represents the CLASS variable level order, and the # after the underscore represents the model number in the case of a lattice. The partitle_# are the patients-at-risk header text used in blockplot statement in the GTL. The # after the underscore represents the model number in the case of a lattice.

5.2 SETTING UP THE PLOT WINDOW

5.2.1 Creating the Lattice

The lattice layout is a very versatile layout even when only including one cell as it opens up a large number of extra spaces that ENTRY statements can be placed.

Figure 2 shows the extra spaces around the plot window that can be utilized when using LAYOUT LATTICE

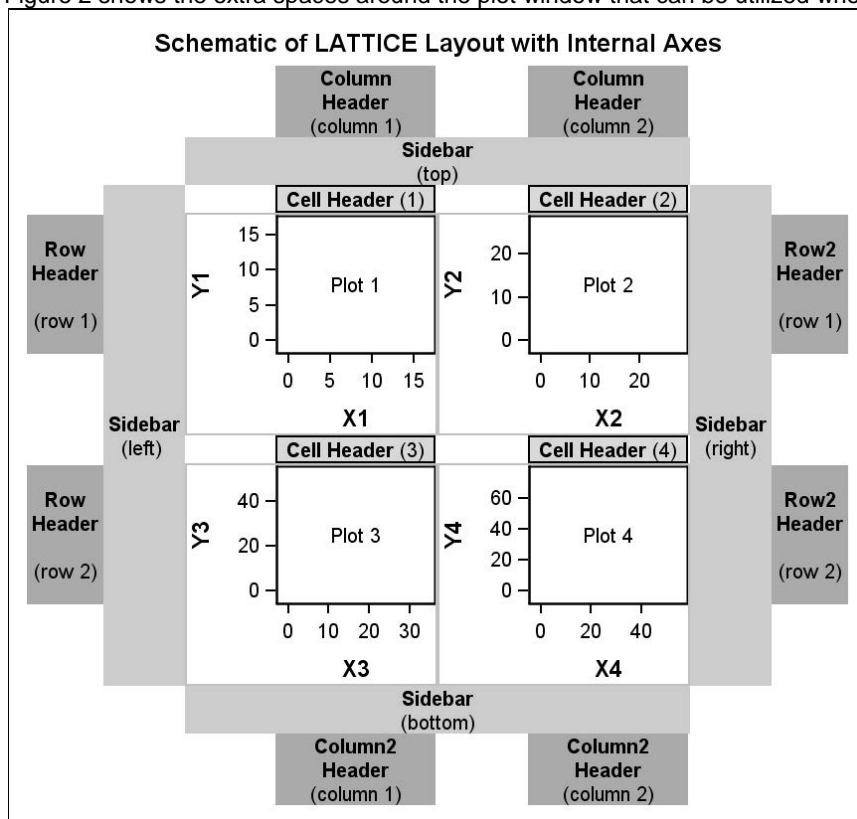


Figure 2. The LAYOUT LATTICE opens up cell headers, row and variable headers, and sidebars. Taken from SAS 9.3 Graph Template Language Reference, Third Edition¹

One of the most flexible properties of the LATTICE layout is that you can nest a LATTICE layout within another LATTICE layout. This allows a great amount of customization of ENTRY text statements. Within the template created by the macro, there is an outer lattice that has one cell for each model called. Within each of these cells is another lattice that is a maximum of one column and multiple rows. The top cell of this inner lattice is for the Kaplan-Meier curves. The bottom cells are created when the patients-at-risk table is located below the x-axis. For the purpose of this paper, the outermost lattice will be referred to as the outer lattice block, and the lattice block within each of the outer lattice block's cells will be referred to as the inner lattice block.

Figure 3 shows the layout of an inner lattice block when RISKLOCATION is set to BOTTOM

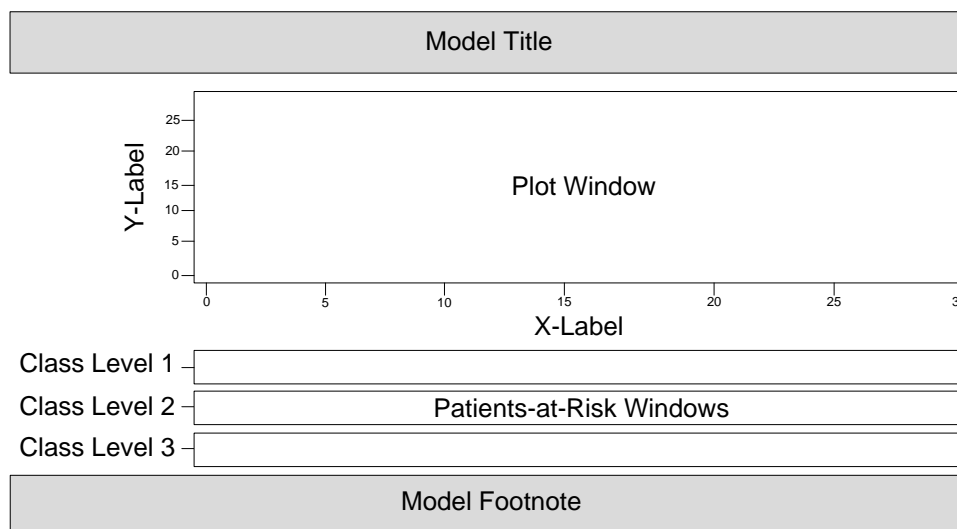


Figure 3. The inner lattice block has one column and multiple rows, with the first cell containing the Kaplan-Meier curves and the remaining cells containing the patients-at-risk table. The text in grey are SIDEBAR blocks

5.2.2 Axes and Labels

The y-axis can be set to be proportional (0.0-1.0) or a percentage (0-100), and can also be reversed with a macro option to plot 1-Survival instead of Survival. The start, end and increment values for both the x and y axes are customizable. If the x-axis is not pre-specified then the macro defaults to a minimum of zero, a maximum of the greatest time in the data set rounded up to the next value divisible by 5, and the increment is set to split the range into five pieces. If the y-axis is not pre-specified then the macro defaults to setting the minimum to 0, the maximum to either 100 or 1.0 depending on the *YTYPE* parameter. The size, font and weight of the axis tick values are customizable.

The x and y labels can be set with macro parameters (*XLABEL* and *YLABEL*). If the x label is not set with the macro parameter, then the time variable label is used. If the y label is not set with the macro then the default is either "Proportion Without Event" or "Percent Without Event" depending on the *YTYPE* parameter.

5.2.3 Titles and Footnotes

Each plot within the lattice can have its own title and its own footnote, and the overall plot image can have an overall title and an overall footnote. The title for each individual plot is added with an *ENTRY* statement within a *SIDEBAR* block with the *ALIGN* option set to top within the inner lattice block. The title can be aligned left, center or right. The footnote for each individual plot is added with an *ENTRY* statement within a *SIDEBAR* block with the *ALIGN* option set to bottom within the inner lattice block.

5.3 PLOTTING THE KAPLAN-MEIER CURVES

The plots are created with the *STEPLOT* statement. The color and pattern of the lines are customizable on a line-by-line basis, but are set automatically if the macro parameters *COLOR* and *PATTERN* are not used. The plot thicknesses are also adjustable, but not on a line-by-line basis.

5.4 PLOTTING THE PATIENTS-AT-RISK NUMBERS

The macro allows for three separate locations to display the patients at risk. They can be below the graph following the x-axis, within the graph but below 0 on the y-axis, or they can be in the summary table at times matching the requested event-free rate estimates. For the two options that follow the x-axis, the start, stop, and increments of these values are customizable.

5.4.1 *RISKLOCATION* set to *TIMELIST*

If the *RISKLOCATION* option is set to *TIMELIST*, then the time-points for patients-at-risk numbers are modified to match the time-points listed in the *TIMELIST* macro parameter and are listed with *ENTRY* statements within the plot statistical summary table (see figure 7 part c). These will be described more in section 5.6.

5.4.2 *RISKLOCATION* Set to *BOTTOM*

The patients-at-risk table below the x-axis is created with several *BLOCKPLOT* statements. The space for these block plots is created by adding rows to the inner lattice block (see figure 3). There is one block plot created per *CLASS* variable level. The x-axis is the same as the Kaplan-Meier plot, the x values are set based on the requested time-points, and the *BLOCK* values are the patients-at-risk values. This creates one block per patients-at-risk value across the x-axis, starting at each of the requested time-points. The *VALUEHALIGN* option is set to the *START* value, which places the *BLOCK* value centered on top of the starting values.

The labels for *CLASS* variable levels are created in one of two ways depending on the *RISKLABELLOCATION* option. When the option is set to *LEFT*, the y-axis labels are turned on for the block plot, which shows the *CLASS* variable level (see figures 4 and 5). When the option is set to *ABOVE*, additional *BLOCKPLOTS* are created with a variable that contains the *CLASS* variable level (see figure 6). The label can be horizontally aligned to the left, center or right. The color of the patient-at-risk numbers can be set to match the color of the plot lines with the macro parameter *RISKLABELCOLOR* (see figure 9 part b). A header is placed above all of the patients-at-risk numbers by placing another block plot above the other block plots (see figure 4). The block value is set to the *PARTITLE_#* variable in the plot dataset (see Display 3).

Figure 4 is a copy of figure 3 with a depiction of the block plots

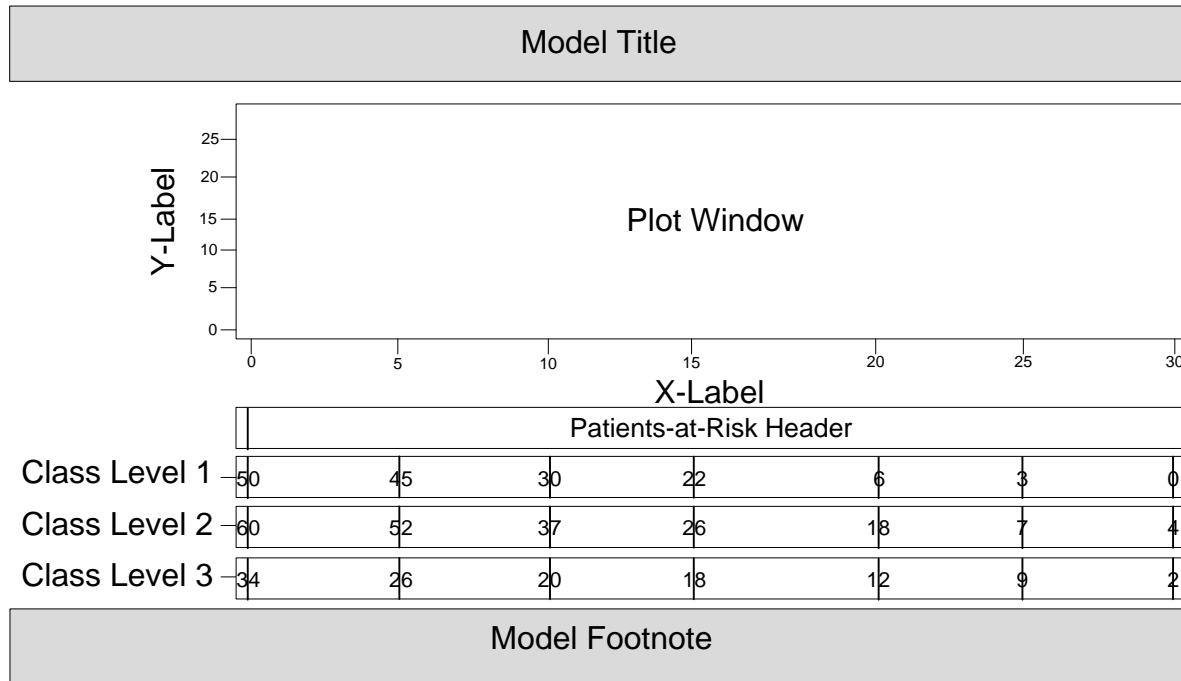


Figure 4. The four block plots below the “Plot Window” cell demonstrate how block plots can be used to create formatted tables of data. The first block plot only has one block starting at zero, and has the block value “Patients-at-Risk Header.” The next three block plots have blocks starting at times 0 to 30 by 5, with block values equaling the number of patients at risk at each of those times. By using the VALUEALIGN=START option, the values are placed on the start of each block instead of within the block.

Figure 5 shows an example of a patients-at-risk table when *RISKLOCATION* parameter set to *BOTTOM*.

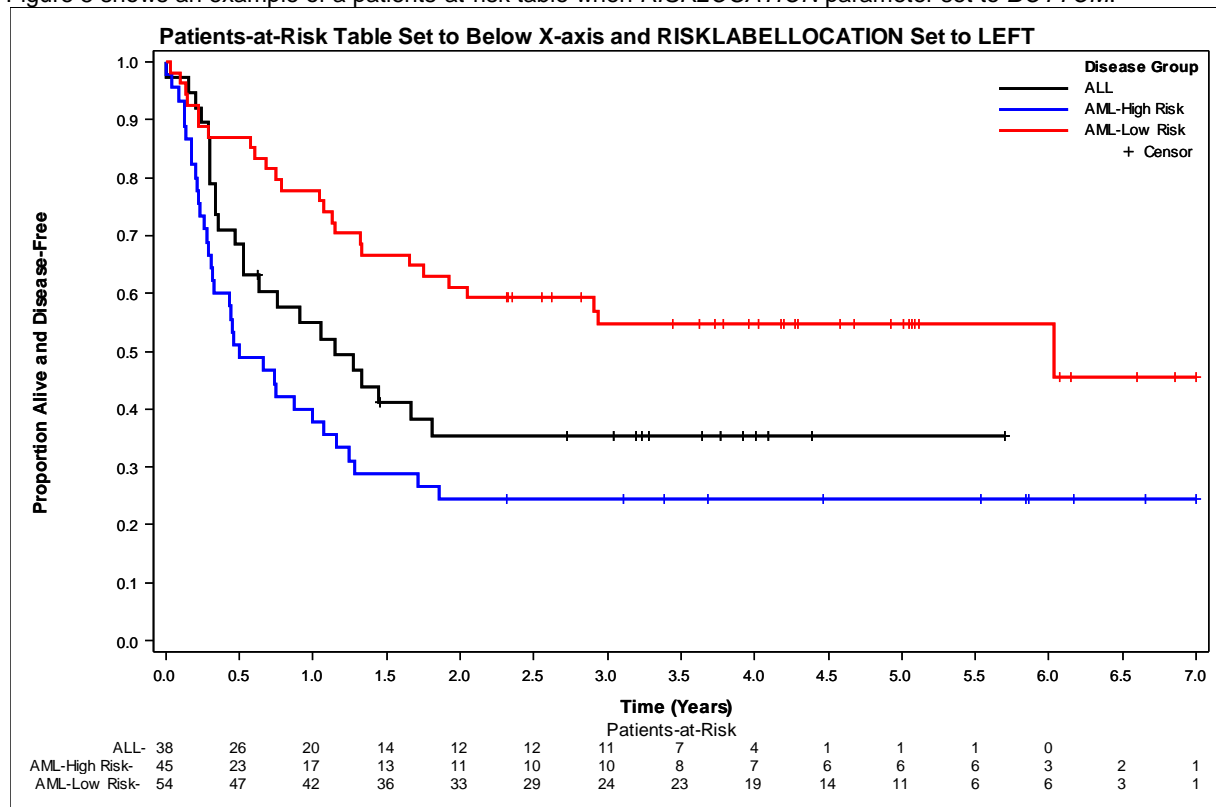


Figure 5. Patients-at-risk table displays one row per *CLASS* variable level in the same order as the in plot summary table.

```

%newsurv(data=sashelp.bmt,time=T,cens=status,cen_vl=0,
class=group,classref=ALL,classvalalign=left,xdivisor=365.25,ytype=ppt,
risklocation=bottom,risklist=0 to 7 by 0.5,
plot=1,summary=0,xmin=0,xmax=7,xincrement=0.5,display=legend,
width=9in,height=6in,symbolsize=6pt,linesize=2pt,color=BLACK BLUE RED,
xlabel=Time (Years),ylabel=Proportion Alive and Disease-Free,gpath=~ /ibm/,
title=Patients-at-Risk Table Set to Below X-axis and RISKLABELLOCATION Set to
LEFT,plotname=figure5,plottype=emf);

```

5.4.3 RISKLOCATION Set to INSIDE

The patients at risk table above the x-axis within the Kaplan-Meier plot is created by adding an INNERMARGIN block to the graph space, and then plotting one block plot per CLASS level within the INNERMARGIN block (see figure 6). The INNERMARGIN block is very useful because it will automatically set aside enough space to fit the contents it contains. The reason a block plot is used to plot the patients-at-risk numbers instead of a scatterplot with the MARKERCHARACTER option is that in SAS 9.2 only block plots were allowed within an INNERMARGIN block. The actual block plots are created using the same methods as in section 5.4.2, and an optional REFERENCELINE is created at the minimum y-axis value to split the patients-at-risk values from the rest of the Kaplan-Meier plot.

Figure 6 is a depiction of the block plots used within an INNERMARGIN block

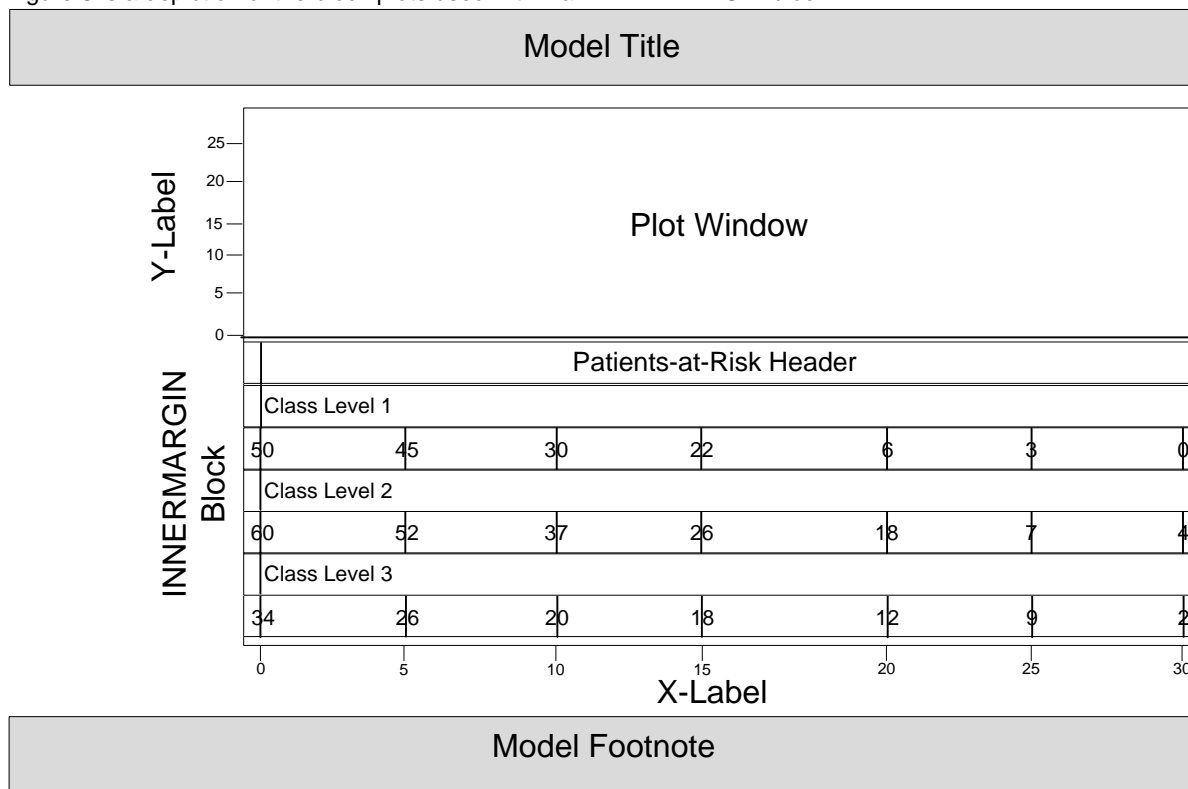


Figure 6. When RISKLOCATION=INSIDE the INNERMARGIN block is added to the plot window below the minimum y-axis value. Block plots are added to this space similarly to section 5.4.2, but this example also shows what the option RISKLABELLOCATION=ABOVE does. Instead of showing the CLASS labels to the left of the block plots, additional block-plots are added to contain the CLASS level values. This is useful when the labels are very long.

Figure 7 shows an example of a patients-at-risk table when *RISKLOCATION* parameter set to *INSIDE*.

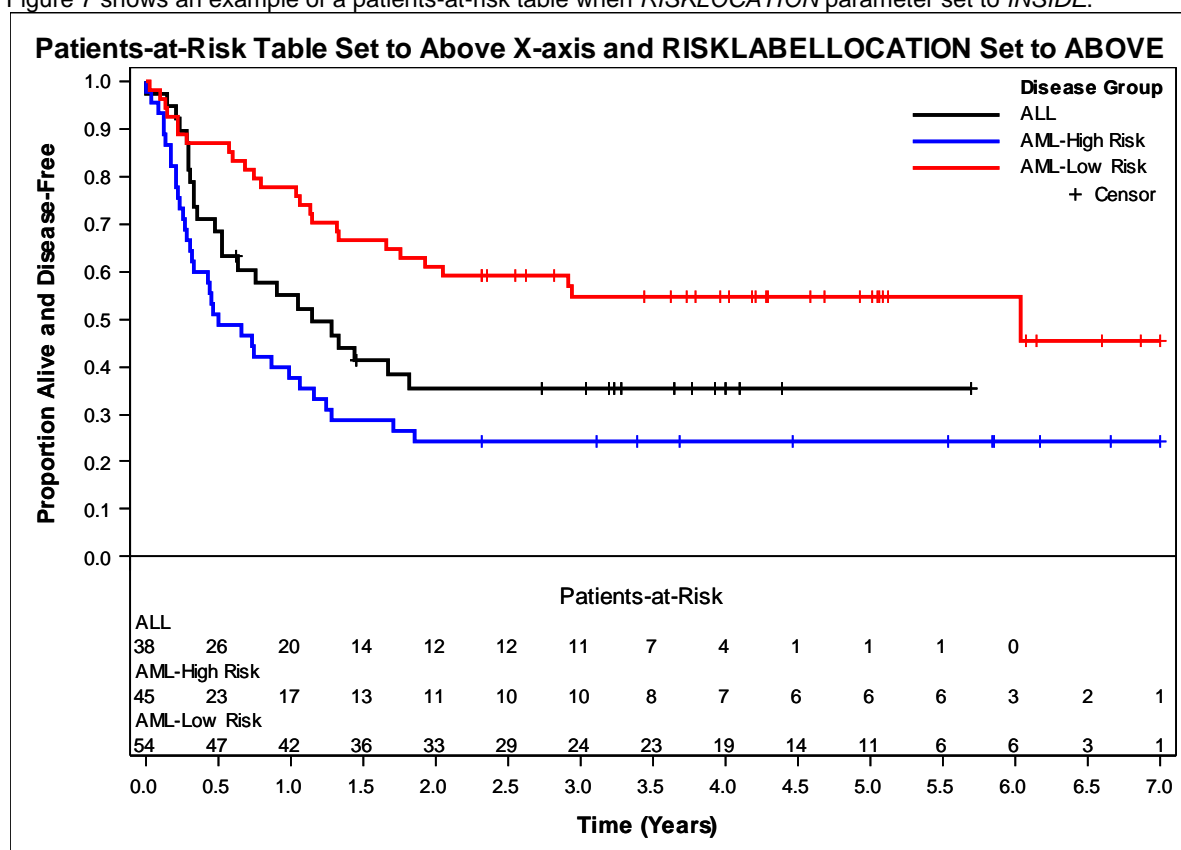


Figure 7. Patients-at-risk table displays one row per *CLASS* variable level in the same order as the in plot summary table.

```
%newsurv(data=sashelp.bmt,time=T,cens=status,cen_vl=0,
class=group,classref=ALL,classvalalign=left,xdivisor=365.25,ytype=ppt,
lsize=10pt,risklocation=inside,risklist=0 to 7 by 0.5,risklabellocation=above,
plot=1,summary=0,xmin=0, xmax=7,xincrement=0.5,display=legend,
width=7in,height=5in,symbolsize=6pt,linesize=2pt,color=BLACK BLUE RED,
xlabel=Time (Years),ylabel=Proportion Alive and Disease-Free,
title=Patients-at-Risk Table Set to Above X-axis and RISKLABELLOCATION Set to
ABOVE,plotname=figure7,gpath=~\ibm/,plottype=emf);
```

5.5 PLOTTING THE CENSOR INDICATORS

The censor values are plotted with the SCATTERPLOT statements. There is one SCATTERPLOT statement for each *CLASS* variable level. The color is taken from the color of the corresponding STEPLOT statement, and the size of the symbol is customizable. There is one additional SCATTERPLOT created with a black color that is run on the first *CLASS* variable level. This is run before any of the others so that the following SCATTERPLOT renders over the values. The purpose of this extra SCATTERPLOT statement is to create a black scatterplot symbol for a legend statement in the summary statistics table.

5.6 SUMMARY STATISTICS TABLE

The following statistics are optionally displayed within the plot: number of patients, number of events, median time-to-event, event-free rate estimates, patients-at-risk numbers, Cox proportional hazards ratio and a p-value. These values are saved into macro variables using the SQL procedure queries with the INTO statement to be used in ENTRY statements.

The actual construction of the summary table within the plot is made with LAYOUT GRIDDED statements, ENTRY statements, and DISCRETELEGEND statements. Layout GRIDDED is a way to create a rectangular grid that can be filled with plots, text, and other objects. One of the features of layout GRIDDED is that it can be nested within itself, allowing for objects within the nest of GRIDDED layouts to be laid out in nearly any way imaginable. Layout GRIDDED also automatically sizes itself to the content within it, with the height coming from the tallest object in the row and the width coming from the widest object in the column. The way that this macro uses layout GRIDDED is to first start with one that creates one column and two rows. The first row is referred to as the class-level gridded block

and it houses all of the class-level statistics such as the number of patients, median time-to-event, and hazard ratios. The second row is referred to as the model-level gridded block and it houses the model-level statistics such as the p-value and table comments.

5.6.1 Class-Level Gridded Block

The class-level gridded block nests another layout GRIDDED that has one column for each requested statistic, except for the legend and the event-free rate estimates which have up to two columns, and has rows equal to the number of values of the *CLASS* variable (which is one if there is no *CLASS* variable) multiplied by the number of event-free rates requested (substituting a one if none are requested) plus one row for the column headers. The reason for the algorithm for the number or rows is described in section 5.6.1.3. Pre-specifying the number of columns and rows at the same time allows for them to be in alignment.

5.6.1.1 Legend

The legend portion of the summary statistics table takes up two columns in the class-level gridded block. The first is the legend symbols and the second is the legend values. The legend is created with the DISCRETELEGEND statement referring to the name option given in the STEPLOT statement. The legend values are turned off so that they can be added in another column with an ENTRY statement. The legend values are then added with an ENTRY statement in the second column. This is to allow the header for the legend values to be aligned above the legend values.

5.6.1.2 Number of Patients/Events, Median Time-to-Event, Hazard Ratios, and Event-free rate Estimates

The rest of the class-level statistics are created in the remaining columns using ENTRY statements aligned to the center.

5.6.1.3 Creating Space between Legend Symbols When Multiple Event-free rate Estimates are called

One of the issues with displaying multiple event-free rate estimates is how to show them. The options are to add multiple horizontal columns, to add vertical rows, or to limit how many event-free rate estimates are displayed. The issue with adding multiple columns horizontally is there is almost never enough space to do so. The issue with only allowing one to be shown is the macro loses versatility. The largest issue with adding rows vertically is how to correctly space the legend to align with the rest of the class-level statistics. This macro created a solution by adding multiple rows vertically and by limiting each DISCRETELEGEND statement to only one *CLASS* variable level. There is a set of rows equal to the number of event-free rates for each level of the *Class* variable. The first row of each set contains the *CLASS* level statistics to create horizontal alignment, and the remaining rows are left blank except for the event-free rate columns (see figure 8).

5.6.2 Model-Level Gridded Block

The model-level gridded block has another LAYOUT GRIDDED block with a maximum of 3 columns and one row. There is one column added for each of the following options if they are called: *LEGEND* in the *DISPLAY* macro parameter, *TABLECOMMENTS* or *PVAL* in the *DISPLAY* macro parameter, and *CENSORMARKERS* when equal to one. The p-value and table comments share a column, so only a maximum of one column is added if either of those are requested.

5.6.2.1 Legend Column

If the legend is displayed in the class-level gridded block, then the step plot that was created earlier with a white line and assigned the name "spacer" is used in a DISCRETELEGEND statement. This is done with the sole purpose of creating the exact space in the left side of the row that the legend in the class-level gridded block creates. The intention for this is that the p-value and/or table comments would be aligned with the class-level gridded block starting with the actual text instead of the legend symbols.




5.6.2.2 P-value and Table Comments Column

A GRIDDED layout is created with one column and as many rows as the number of table comments plus one row for the p-value if it was requested in the *DISPLAY* macro parameter. The p-value is always displayed first. The number of table comments is determined by splitting the macro parameter *TABLECOMMENTS* into segments with the "" delimiter. Each of these items is then created with an ENTRY statement.

5.6.2.3 Censor Markers Column

A legend for the censor markers is created using the black scatterplot statement described earlier in the Plotting the Censor Indicators Section in a DISCRETELEGEND statement. The legend is always aligned to the right side of the model-level gridded block. If the SAS version is 9.3 or later, then the option AUTOITEMSIZE is used to keep the symbol size the same as the text in the legend.

Figure 8 shows the summary statistics table if all LAYOUT GRIDDED blocks had their borders turned on.

Class-Level Gridded Block					
Legend	Class Value	Total	Events	Time-Point	KM Estimate
	ALL	38	24	1 Years	0.55 (0.41-0.73)
				2 Years	0.35 (0.23-0.55)
	AML-High Risk	45	34	1 Years	0.38 (0.26-0.55)
				2 Years	0.24 (0.15-0.41)
	AML-Low Risk	54	25	1 Years	0.78 (0.67-0.90)
				2 Years	0.61 (0.49-0.76)

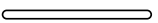
Model-Level Gridded Block						
	<table border="1"> <tr> <td>Score P-value: 0.0010</td> </tr> <tr> <td>Table Comment 1</td> </tr> <tr> <td>Table Comment 2</td> </tr> <tr> <td>Table Comment 3</td> </tr> </table>	Score P-value: 0.0010	Table Comment 1	Table Comment 2	Table Comment 3	+ Censor
Score P-value: 0.0010						
Table Comment 1						
Table Comment 2						
Table Comment 3						

Figure 8. An example of the nested LAYOUT GRIDDED statements when two time-points are requested for event-free rates. This shows what the plot summary statistics table would look like if the borders around the layouts were turned on.

```
%newsurv(data=sashelp.bmt,time=T,cens=status, cen_vl=0,
class=group,classref=ALL, classvalalign=left timelist=1 2,
xdivisor=365.25, xincrement=0.5, xlabel=Time (Years), ylabel=Proportion Alive and Disease-Free,
display=legend total event timelist pval tablecomments,
tablecomments=Table Comment 1`Table Comment 2`Table Comment 3`Table Comment 4,
plot=1,summary=0,xmin=0, xmax=7,
width=7in,height=5in,symbolsize=6pt,linesize=2pt,ytype=ppt,color=BLACK BLUE RED,
plotname=figure8,gpath=-/ibm/,plottype=emf);
```

6.0 DESIGNING THE LATTICE

The *NEWSURV* macro has the ability to generate multiple Kaplan-Meier plots into one graphic file within a rectangular lattice. Each plot is unique and is customized individually. Macro parameters are set for multiple plots by using the ‘|’ delimiter. This delimiter was chosen because it does not get used in labels and titles very often. The macro then uses a macro do loop to process a number of models equal to *NMODELS*. All of these models are saved into one data set for plotting (see section 5.1) with the variables having an underscore and number to indicate which model they came from. Within the GTL template the LATTICE layout is used to set up the plot window into multiple panes. A macro do loop is used within the LATTICE layout to process the different requested plots.

Figure 9 shows an example the customizability of each plot within a lattice.

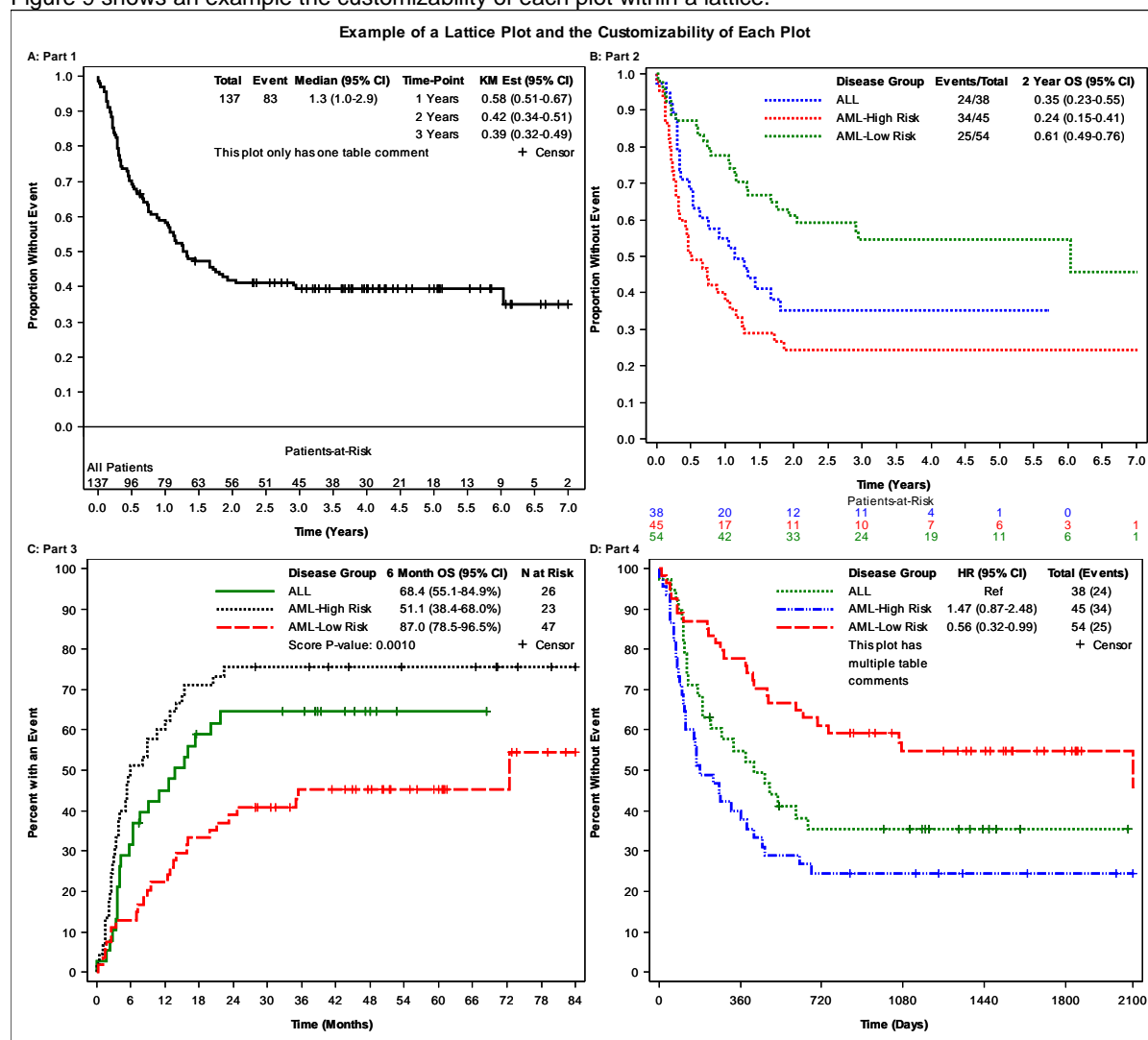


Figure 9. Each cell of the lattice uses the same variables, but has different options enabled to show the versatility of the NEWSURV macro.

```
%newsurv(data=sashelp.bmt,
  time=T,cens=status,cen_v1=0,class=|group|group|group,
  classdesc=All Patients|||,classvalalign=left,
  risklabellocation=above|||,riskcolor=0|1|0|0,classref=ALL,
  timelist=1 to 3 by 1|2|6|,timedx=Years|Years|Months|Days,
  display=total event median timelist tablecomments|
  legend total event timelist|
  legend timelist pval|
  legend hr tablecomments,censormarkers=1|0|1|1,
  tablecomments=This plot only has one table comment|||
  This plot has`multiple table`comments,
  ptabsize=7pt,tsize=7pt,ovtsize=10pt,xtickvalsize=7pt,
  ytickvalsize=7pt,lszize=7pt,parsize=7pt,
  riskklocation=inside|bottom|timelist|,risklist=0 to 7 by 0.5|0 to 7 by 1|3 4|,
  xminoffset=0.03|0.02|0.02|0.02,xmin=0,xmax=7|7|84|2100,xincrement=0.5|0.5|6|180,
  xlabel=Time (Years)|Time (Years)|Time (Months)|Time (Days),
  ylabel=||Percent with an Event|,ytype=ppt|ppt|pct|pct,
  xdivisor=365.25|365.25|30.44|1,sreverse=0|0|1|0,
  plot=1, summary=0,
  width=10in,height=9in,symbolsize=6pt,linesize=2pt,
  color=BLACK BLUE RED|BLUE RED GREEN|GREEN BLACK RED|GREEN BLUE RED,
  pattern=SOLID SHORTDASH LONGDASH|SHORTDASH|SOLID SHORTDASH LONGDASH|
  SHORTDASH MEDIUMDASHDOTDOT LONGDASH,
```

```

rows=2,columns=2,nmodels=4,order=rowmajor,
title=A: Part 1|B: Part 2|C: Part 3|D: Part 4,
titlealign=left,
ovtitle=Example of a Lattice Plot and the Customizability of Each Plot,
plotname=figure9,gpath=~ /ibm/);

```

7.0 STATISTICAL REPORT TABLE

In addition to the plot graphic, an external report table can also be created with the macro that can include the number of patients and events, median time-to-event, hazard ratios, event-free rate estimates, and p-values as columns. Each of these columns can be turned on and off separately from the summary table in the plot. The *TITLE* and *FOOTNOTE* for each model is displayed before and after the statistics for each model, and a title that displays at the top of the table (*TABLETITLE*) and a footnote that displays at the bottom of the table (*TABLEFOOTNOTE*) can be added. Footnotes with symbol markers are automatically added for Cox proportional hazard ratios, Kaplan-Meier estimates, and p-values. The report table data set can be saved out from the macro, and if a lattice of plots is created by the macro, each model is put into the same data set. The statistical report table is generated using the *REPORT* procedure to allow customizability and for the ease of style modifiers.

7.1 PUTTING THE REPORT TABLE DATA SET TOGETHER

The statistical report table data set is created using PROC SQL after the analysis has been run. A blank temporary data set is created for each model called with variables for each of the statistics along with a few variables used for sorting and headers using the CREATE TABLE statement with no attached query. The statistics are then inserted into the data set using INSERT INTO statements along with a SET statement for each row that assigns values to each variable with a sub-query to the analysis tables created earlier in PROC LIFETEST and PROC PHREG.

7.1.1 Outputting the Statistical Report Data Set

Once the temporary data set is completed, it is then inserted into a designated data set that houses all of the model's output. The macro has the ability to output this data set to a specified name with the *OUT* macro parameter.

7.1.2 Combining the *OUT* Macro Parameter with the *NEWTABLE* Macro Parameter

If the data set name passed to the *OUT* macro parameter matches an already existing data set then the macro will do one of two actions depending on the macro parameter *NEWTABLE*. If *NEWTABLE* is set to 1, then the macro will overwrite the data set and save the statistical report data set out. If *NEWTABLE* is set to 0, then the macro will attempt to append the new statistical report data set into the *OUT* data set. This allows separate calls to the macro to output into the same final analysis dataset, and the final call to the macro can output the fully combined statistical report table.

```

%newsurv(data=sashelp.bmt,time=T,cens=status,cen_vl=0,
classdesc=All Patients,timelist=1 to 2 by 1, timedx=Years,xdivisor=365.25,
summary=0,plot=0,
title=SASHELP.BMT Dataset with no Class Variable,
footnote=This is the footnote for this macro call 1,
out=table2);
%newsurv(data=sashelp.bmt,time=T,cens=status,cen_vl=0,
class=group,classref=ALL,
timelist=1 to 2 by 1, timedx=Years,xdivisor=365.25,
summary=1,plot=0,destination=rtf,
title=SASHELP.BMT Dataset with Class Variable,
footnote=This is the footnote for macro call 2,
outdoc=~ /ibm/pharmaSUG_fifth_table.rtf,
tabletitle=Example of Outputting Two Macro Calls to One Table,
out=table2,newtable=0,
tablefootnote=The statistics in this table are identicle to Table 1);

```

Table is an example of a statistical report table with analysis gathered from two macro calls:

Example of Outputting Two Macro Calls to One Table

	Event/Total	Median (95% CI) [†]	Hazard Ratio (95% CI) [‡]	Survival Estimates (95% CI) [†]	P-value
SASHELP.BMT Data set with no Class Variable					
All Patients	83/137	1.3 (1.0-2.9)		1 Years: 58.3 (50.6-67.2) 2 Years: 42.0 (34.5-51.2)	
This is the footnote for this macro call 1					
SASHELP.BMT Data set with Class Variable					
Disease Group					0.0010 [§]
ALL	24/38	1.1 (0.5-NE)	Ref	1 Years: 54.9 (41.1-73.4) 2 Years: 35.3 (22.7-54.8)	
AML-High Risk	34/45	0.5 (0.3-1.2)	1.47 (0.87-2.48)	1 Years: 37.8 (26.0-55.0) 2 Years: 24.4 (14.6-40.9)	
AML-Low Risk	25/54	6.0 (1.9-NE)	0.56 (0.32-0.99)	1 Years: 77.8 (67.4-89.7) 2 Years: 61.1 (49.4-75.6)	
This is the footnote for macro call 2					

[†]Kaplan-Meier method; [‡]Cox model; [§]Score test;

The statistics in this table are identical to Table 1

Table 5. Statistical report table with data from two macro calls. The statistics listed in this table are the same as in table 2, but the statistics in table 2 were gathered with one macro call. This table shows that the end result looks the same with either method; however the ability to append analysis into the same outputted data set allows analysis to be gathered from multiple input data sets as well.

8.0 CONCLUSION

This primary goal of this macro is to allow the fast and efficient creation of Kaplan-Meier curves for time-to-event analysis without the need for back-end manipulation. In addition, this macro gives an example of the ability to use the gridded and lattice layouts in combination with ENTRY statements within the plot window in neatly constructed tables. This allows an even greater flexibility in the use of the Graph Template Language to display graphical analysis and means that fewer tables and figures could be required in publications. The future of SAS graphics is very bright and the power of the Graph Template Language are only limited to the programmer's imagination.

9.0 RECOMMENDED READING

- SAS 9.3 Graph Template Language Reference, Third Edition
- SAS 9 PROC TEMPLATE Styles Tip Sheet
- SAS/STAT® 9.2 User's Guide, Second Edition. Specifically the LIFETEST Procedure and the PHREG procedure.

10.0 REFERENCES

¹Copyright 2014, SAS Institute Inc., Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute Inc., Cary, NC.

11.0 CONTACT INFORMATION

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

Name: Jeffrey Meyers
 Enterprise: Mayo Clinic
 Address: 200 First Street SW
 City, State ZIP: Rochester, MN 55905
 Work Phone: 507-266-2711
 E-mail: Meyers.jeffrey@mayo.edu / jpmeyers.spa@gmail.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.

12.0 APPENDIX I: TABLE OF REFERENCE TERMS

Table 3 SAS keywords used within this paper:

Keyword	Definition
CALL SYMPUT	SAS function that takes a value and assigns it to a macro variable.
PUT	SAS function that takes a variable and returns it as a character string per an inputted format.
VFORMAT	SAS function that takes a variable and returns its format as a character string. Returns default values depending on variable type if no format is assigned.
NULL	SAS statement used in a DATA step to run DATA step functions without creating a data set.
ODS OUTPUT	SAS statement used to output analysis tables from SAS procedures. Individual or multiple tables can be called from one ODS OUPUTPUT statement.
ODS GRAPHICS	SAS statement used to modify options when using the SG graphics engine.
LIFETEST	SAS procedure designed for Kaplan-Meier analyses.
CENSORED SUMMARY	SAS ODS table called from procedures such as PROC LIFETEST that contain a summary of the number of patients and number of censors.
QUARTILES	SAS ODS table called from procedures such as PROC LIFETEST that contain the computed quartiles of the survival estimates.
SURVIVAL PLOT	SAS ODS table called from PROC LIFETEST that contains the times, survival estimates, and other statistics generated from the plot created in PROC LIFETEST.
HOMTESTS	SAS ODS table called from PROC LIFETEST that contains the p-values for the covariate tests.
OUTSURV	SAS procedure option within PROC LIFETEST that designates a data set to output survival estimates at pre-designated times or at each time an event or censor occurs.
CONFTYPE	SAS procedure option within PROC LIFETEST that determines the method of calculating the confidence bounds for survival estimates.
TIMELIST	SAS procedure option within PROC LIFETEST that determines specific times to compute the survival function.
REDUCEOUT	SAS procedure option within PROC LIFETEST that reduces the observations within the OUTSURV dataset to only include the time-points called with the TIMELIST option.
PLOT	SAS procedure option within PROC LIFETEST that generates a plot.
ATRISK	SAS procedure option within PROC LIFETEST that designates patients-at-risk to be displayed when used with the PLOT option.
PHREG	SAS procedure designed for Cox proportional hazard regression analyses.
CLASS	SAS statement within PROC PHREG that initializes a covariate as being discrete and allows a reference level to be selected.
MODEL	SAS statement within PROC PHREG that defines the Cox proportional hazards regression model.
RISKLIMITS	SAS procedure option within PROC PHREG that turns on the confidence intervals for the hazard ratios.
PARAMETER ESTIMATES	SAS ODS table called from PROC PHREG that contains the Cox proportional hazards regression model estimates, p-values, and confidence intervals.
TYPE3	SAS ODS table called from PROC PHREG that contains the p-values for the type-III chi-square tests.
TEMPLATE	
Graph Template Language (GTL)	SAS language within the TEMPLATE procedure used to design graphics.

SCATTERPLOT	SAS statement within PROC TEMPLATE that generates a scatterplot.
NAME	SAS procedure option within PROC TEMPLATE that designates a text name for a plot statement. This allows the plot statement to be referred to later in a legend statement.
MARKERCHARACTER	SAS procedure option within PROC TEMPLATE that allows a variable of values to replace the scatterplot markers in the plot window.
ENTRY	SAS statement within PROC TEMPLATE that generates a text string at a specified location.
LAYOUT LATTICE	SAS statement within PROC TEMPLATE that generates a rectangular array of cells for multiple plot windows. Width and height of cells can be determined on the row and column level.
SIDEBAR	SAS statement within PROC TEMPLATE that creates a space within a lattice layout that can contain text and/or plot statements. The space can be created on the top, bottom, left, or right side of the lattice depending on the ALIGN option.
ALIGN	SAS procedure option within PROC TEMPLATE that determines where the space is created for a SIDEBAR block. Can be top, bottom, left, or right.
STEPLOT	SAS statement within PROC TEMPLATE that generates a step plot.
NAME	SAS procedure option within PROC TEMPLATE that designates a text name for a plot statement. This allows the plot statement to be referred to later in a legend statement.
DRAWTEXT	SAS statement within PROC TEMPLATE that generates text at a specified location based on multiple types of coordinate systems. Only available in SAS 9.3 or later.
REFERENCELINE	SAS statement within PROC TEMPLATE that generates a line across the entire plot window at a specified point on either the x axis (makes a vertical line) or y axis (makes a horizontal line).
LAYOUT GRIDDED	SAS statement within PROC TEMPLATE that generates a rectangular array of cells for multiple plot windows. Width and height of cells are automatically determined by the content of the cells
DISCRETELEGEND	SAS statement within PROC TEMPLATE that generates a legend for discrete values.
AUTOITEMSIZE	SAS procedure option within PROC TEMPLATE that allows legend items to automatically scale with the text font size. This is only available in SAS 9.3 or later.
REPORT	SAS procedure designed for displaying tabular output.
COMPUTE	SAS statement within PROC REPORT that creates a block that allows DATA step features.
BEFORE	SAS procedure option within PROC REPORT that is used with the COMPUTE statement to create a block before (above) each value of a group variable.
AFTER	SAS procedure option within PROC REPORT that is used with the COMPUTE statement to create a block after (below) each value of a group variable.
NOPRINT	SAS procedure option within PROC REPORT that indicates a column will not be printed.
SQL	SAS procedure designed for manipulating and printing data.
INTO	SAS statement within PROC SQL that takes the values pulled from a query and saves them into SAS macro variables.
CREATE TABLE	SAS statement within PROC SQL that takes the values pulled in a query and saves them into a data set.
INSERT INTO	SAS statement within PROC SQL that takes a query and appends the rows into a current data set.
SET	SAS statement within PROC SQL that is used with INSERT INTO to insert one row of values where each column is assigned individually.
SGRENDER	SAS procedure designed for generating graphics based on a data set and a template created from the TEMPLATE procedure.

Table 3. List of SAS keywords mentioned in paper and their definition.

Table 4 macro parameters used within this paper:

Macro Parameter	Definition
NEWSURV	Name of the macro.
DATA	Macro parameter that determines the input data set.

TIME	Macro parameter that determines the time variable.
CENS	Macro parameter that determines the censor or event variable.
CEN_VL	Macro parameter that determines the value of the censor or event variable that indicates a censor.
CLASS	Macro parameter that determines the covariate to be analyzed in the statistical models.
TITLE	Macro parameter that determines the title printed in the plot as well as the subheader within the report table.
CLASSCOV	Macro parameter that determines the categorical covariates to be adjusted for in the multivariate statistical models.
CONTCOV	Macro parameter that determines the continuous covariates to be adjusted for in the multivariate statistical models
FOOTNOTE	Macro parameter that determines the footnote printed in the plot as well as the subfooter within the report table.
TABLETITLE	Macro parameter that determines the title printed at the top of the report table.
TABLEFOOTNOTE	Macro parameter that determines the footnote printed at the bottom of the report table.
XDIVISOR	Macro parameter that transforms the <i>TIME</i> variable into different units.
TIMELIST	Macro parameter that sets up time-points for the event-free rate estimates.
RISKLIST	Macro parameter that sets up the time-points for the patients-at-risk tables.
WHERE	Macro parameter that subsets the input data set by the conditions entered into the parameter.
LANDMARK	Macro parameter that resets the starting time for the Kaplan-Meier curves to a time after the original starting time. Can be either a numeric value or a numeric variable.
RISKLOCATION	Macro parameter that designates where the patients-at-risk table will be shown in the plot window. Options are <i>BOTTOM</i> , <i>INSIDE</i> , and <i>TIMELIST</i> .
BOTTOM	Macro option for parameter <i>RISKLOCATION</i> that sets the patients-at-risk table to be shown below the x-axis.
INSIDE	Macro option for parameter <i>RISKLOCATION</i> that sets the patients-at-risk table to be shown above the x-axis within the plot-window
TIMELIST	Macro option for parameter <i>RISKLOCATION</i> that sets the patients-at-risk values to be shown next to the event-free rates in the plot summary table.
YTYPE	Macro parameter that determines the scale of the y-axis to either be a proportion or a percentage.
XLABEL	Macro parameter that determines the label for the x-axis.
YLABEL	Macro parameter that determines the label for the y-axis.
RISKLABELLOCATION	Macro parameter that determines where the labels for the patients-at-risk table are shown. Options are <i>LEFT</i> , <i>ABOVE</i> , and missing (none).
LEFT	Macro option for parameter <i>RISKLABELLOCATION</i> that sets the patients-at-risk labels to the left of the patients-at-risk table.
ABOVE	Macro option for parameter <i>RISKLABELLOCATION</i> that sets the patients-at-risk labels to be above the patients-at-risk table. This is only available in SAS 9.3 or later.
RISKLABELCOLOR	Macro parameter that determines if the numbers for the different <i>CLASS</i> levels of the patients-at-risk table are colored to match the plot lines.
DISPLAY	Macro parameter that determines which statistics appear in the plot summary table.
LEGEND	Macro option for parameter <i>DISPLAY</i> that determines if the legend is displayed in the plot.
PVAL	Macro option for parameter <i>DISPLAY</i> that determines if the model p-value is displayed in

	the plot.
TABLECOMMENTS	Macro option for parameter <i>DISPLAY</i> that determines if the user-specified comments for the plot are displayed.
CENSORMARKERS	Macro parameter that determines if censor marker symbols are displayed in the plot.
TABLECOMMENTS	Macro parameter that allows the user to specify comments to be listed within the plot statistical summary.
NMODELS	Determines how many models are processed while the macro runs.

Table 4. List of macro parameters mentioned in paper and their definition.