PharmaSUG China 2019 - Paper HW-039

Build Popular Clinical Graphs using SAS

Hands on Tutorial

Sanjay Matange

ABSTRACT

Survival Plots, Forest Plot, Waterfall Charts and Swimmer Plots are some of the popular, frequently requested graphs in clinical research. These graphs are easy to build with SAS® using the SGPLOT procedure. Once you understand how SGPLOT works, you can develop a plan, prepare the data as per this plan and then use the right plot statements to create almost any graph.

This Hands-on workshop will take you step-by-step through the process needed to create these graphs. You will learn how to analyze the graph and make a plan. Then, put together the data set that has all the needed information. Finally, layer the right plot statements in the right order to build the graph. Once you master the process for these graphs, you can use the same process to build almost any other graph. This Hands on Workshop will teach you how to use SGPLOT procedure like a pro.

INTRODUCTION

The SGPLOT procedure allows you to combine multiple plot statements to create plots of varying complexity. The procedure supports over 30 different plot and supporting statements to build the graph one step at a time. These plots can be layered to build composites and can be combined with the AXISTABLE statement to add tabular rows or columns.

In this presentation, we will create the popular Survival Plot and Subgrouped Forest Plot. The Waterfall and Swimmer plots can be created following a similar process, but will not be discussed due to the time constraints for the tutorial.

Here we use the terms “Plots”, “Graphs” and “Charts” synonymously for the entire output from the procedure. The term “plot statement” is used for each of the statements supported in the procedure that can be combined together to create the output.

TASK 1 – SURVIVAL PLOT

The Survival Plot is automatically created by the LIFETEST Procedure. The program to create the survival plot for the BMT data is shown in Figure 1.1.

```
libname how 'c:\how\matange\data';
ods listing image_dpi=200 gpath='c:\how\matange\graph';
ods graphics / reset imagename='Survival_Lifetest';
proc lifetest data=how.BMT plots=survival(atrisk=0 to 2500 by 500);
   time T * Status(0);
   strata Group / test=logrank adjust=sidak;
run;
```

Figure 1.1 – LIFETEST Procedure program for BMT

The code in Figure 1.1 uses the LIFETEST procedure to process the data and create only the Survival plot as requested using the PLOTS option. The graph output to the LISTING destination is saved in the “c:\how\matange\graph” folder using DPI of 200.

The name of the graph is set to “Survival_Lifetest” using the IMAGENAME option in the ODS GRAPHICS statement. The graph is shown in Figure 1.2.

For the first task, we want to create a survival plot as shown in Figure 1.2 and customize its appearance using the SGPLOT procedure.

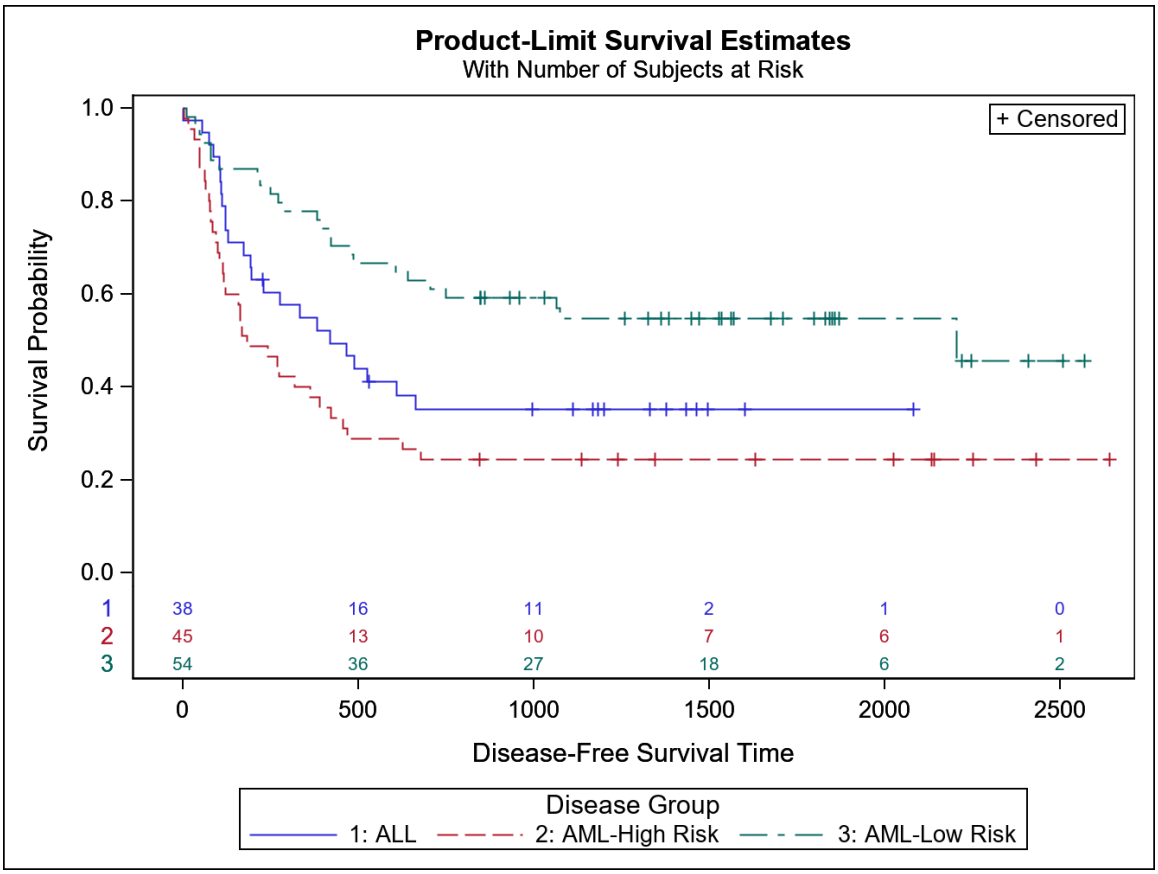


Figure 1.2 – Default Survival plot from the LIFETEST Procedure.

STEP – 1

To create a graph as shown in Figure 1.2, we need to get the data for the graph. We can do this by running the LIFETEST procedure and writing the data created for the graph to a SAS data set.

Open the file "C:\how\matange\pgm\Survival_Data.sas". Submit the program. This program sets a libref to the local data folder for the BMT data set.

```
libname how 'c:\how\matange\data';

/*--Create data for Survival Plot--*/
ods output Survivalplot=how.SurvivalPlotData;
ods graphics / reset imagename='Survival_Lifetest';
proc lifetest data=how.BMT plots=survival(atrisk=0 to 2500 by 500);
   time T * Status(0);
   strata Group / test=logrank adjust=sidak;
   run;

/*--Display survival data--*/
proc print data=survivalPlotData(obs=10);
run;
```

Figure 2.2 – Create data set from LIFETEST Procedure.

Note, the above program is run using the default HTML destination. The output is written to a HTML document using the default setting different from the graph shown in Figure 1.2. The data created by the procedure for building the Survival Plot is saved to the "**how.SurvivalPlotData**" data set using the ODS

OUTPUT statement. The program also prints the first 10 observations from this data set shown in Figure 1.3. We will use this data set to create a custom Survival Plot.

Obs	Time	Survival	AtRisk	Event	Censored	tAtRisk	Stratum	StratumNum
1	0	1.00000	38	0	.	.	1: ALL	1
2	0	.	38	.	.	0	1: ALL	1
3	1	0.97368	38	1	.	.	1: ALL	1
4	55	0.94737	37	1	.	.	1: ALL	1
5	74	0.92105	36	1	.	.	1: ALL	1
6	86	0.89474	35	1	.	.	1: ALL	1
7	104	0.86842	34	1	.	.	1: ALL	1
8	107	0.84211	33	1	.	.	1: ALL	1
9	109	0.81579	32	1	.	.	1: ALL	1
10	110	0.78947	31	1	.	.	1: ALL	1

Figure 1.3 – Data set for Survival Plot

We will build the graph shown in Figure 1.2 one step at a time. Note, this graph can be viewed as a layered graph with the following layers:

1. Display of the survival curves of survival by time by stratum using a STEP plot.
2. Display of a discrete legend of the stratum values below the x-axis.
3. Display of the censored observations of censored x time using a SCATTER plot.
4. Display of a discrete legend in upper right of the plot area for censored observations.
5. Display of the “Subjects at Risk” values at the bottom of the data area using an AXISTABLE plot.

STEP – 2

Now, let us start by creating a graph of the survival values by time and stratum. Start a New Program and **type in the code below and submit the code**. Or, open "c:\how\matange\pgm\Survival_Plot.sas".

```
libname how 'c:\how\matange\data';
ods html close;
ods listing image_dpi=200 gpath='c:\how\matange\graph';

/*--Create plot with survival curves--*/
ods graphics / reset imagename='Survival_Plot_1_5';
title 'Product-Limit Survival Estimates';
title2 h=0.8 'With Number of AML Subjects at Risk';
proc sgplot data=how.SurvivalPlotData;
  step x=time y=survival / group=stratum name='s';
run;
```

Figure 1.4 – Survival Curves by time.

The code above will create the graph shown in Figure 1.5. This graph displays the survival curves by time and stratum.

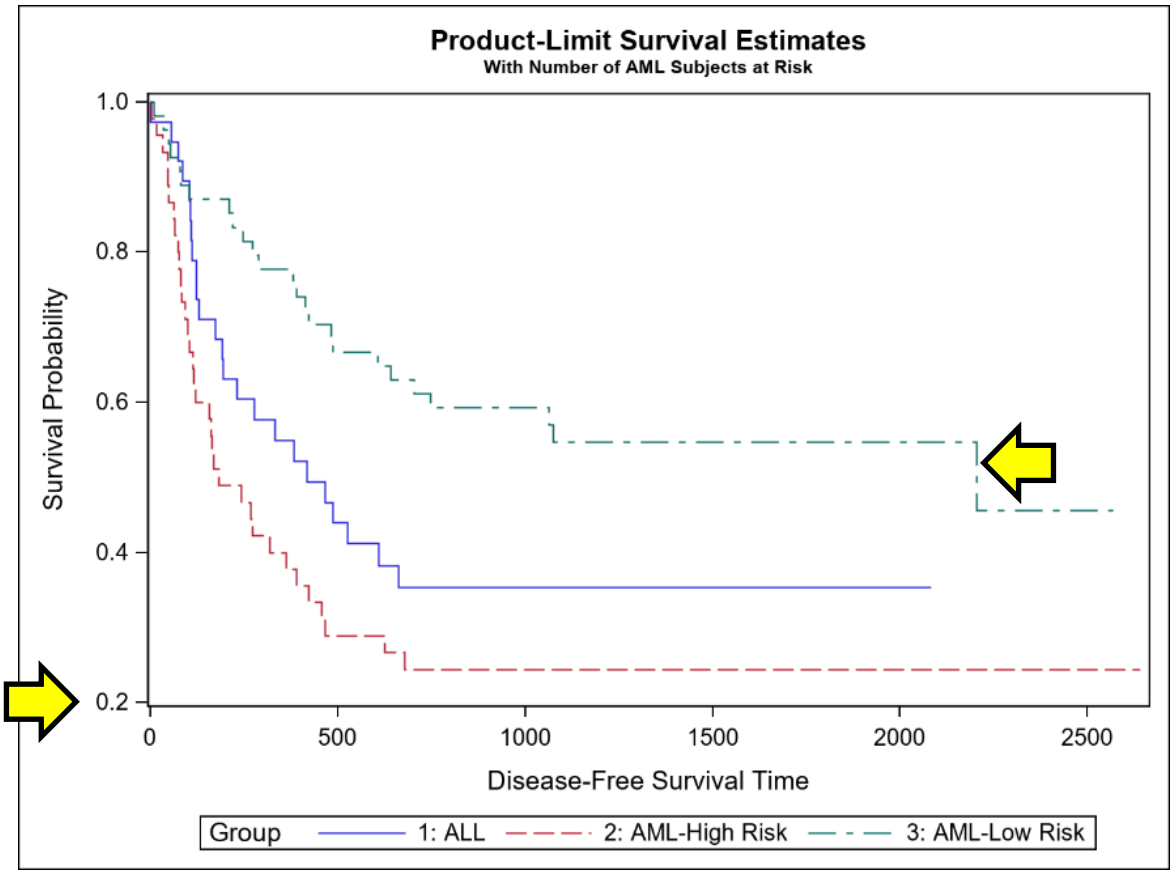


Figure 1.5 – Survival curves by time and stratum

Note the following items:

1. The y-axis min is determined by data range and is not zero.
2. The curve lines use colors and line patterns, which is not optimal for a step plot.
3. The legend uses a default title of "Group".

STEP – 3

Now, let us customize the step plot to set y-axis min=0, use solid plot lines and customize the legend. Start with the code in Figure 1.4 and add or edit the code in bold shown in Figure 1.6.

```
/*--Create plot with solid survival curves and Legend Title--*/
ods graphics / reset attrpriority=color imagename='Survival_Plot_1_7';
title 'Product-Limit Survival Estimates';
title2 h=0.8 'With Number of AML Subjects at Risk';
proc sgplot data=how.SurvivalPlotData;
  step x=time y=survival / group=stratum name='s';
  yaxis min=0;
  keylegend 's' / title='Disease Group: ' linelength=30;
run;
```

Figure 1.6 – Solid Survival Curves with legend title and shorter lines.

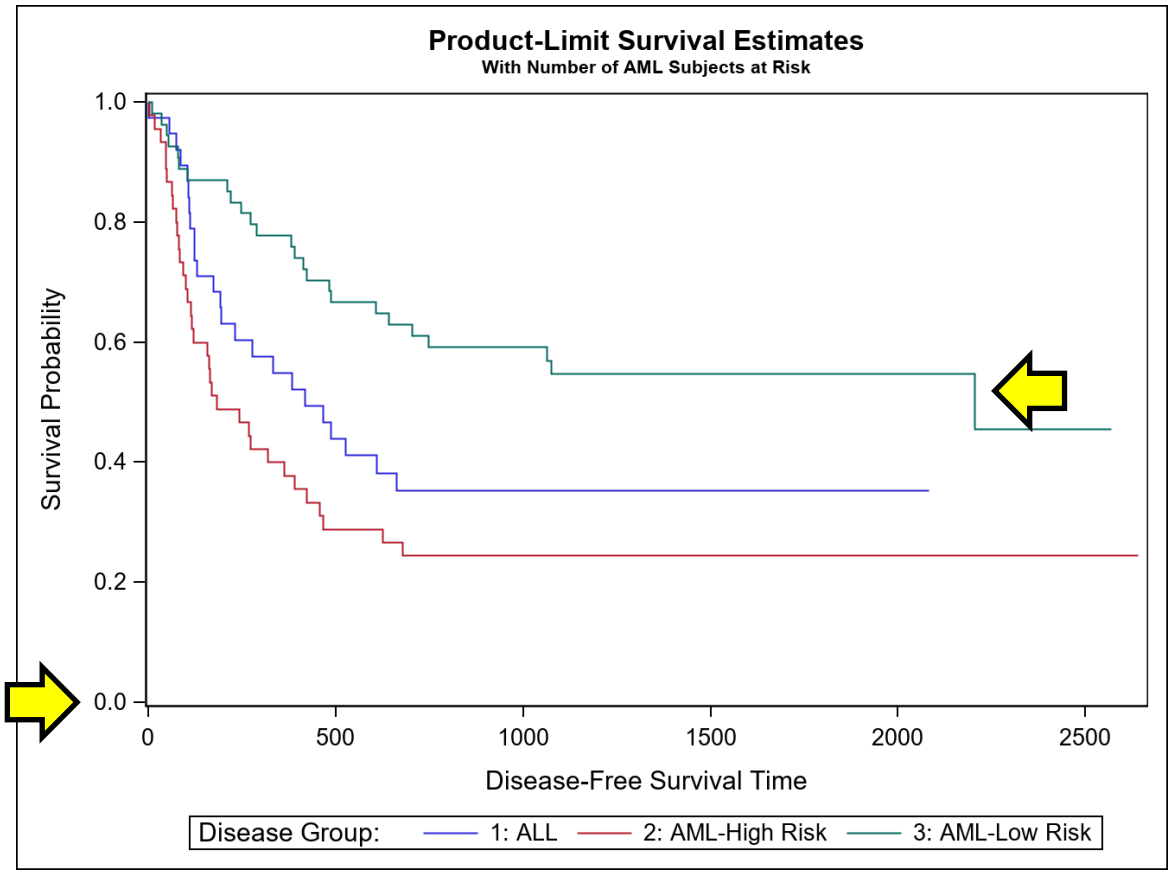


Figure 1.7 – Survival curves by time and stratum

Note the following features of this graph:

1. The y-axis min is now set to zero.
2. The curve lines use colors with solid line patterns.
3. The legend title is changed.
4. The lines indicating the values in the legend are shortened.

STEP – 4

Now, let us customize graph by adding overlaid markers for the censored observations, a legend for the censored observations. Starting with the code in Figure 1.6, add the code shown in bold below in Figure 1.8.

```
/*--Create plot with solid survival curves and Legend Title--*/
ods graphics / reset attrpriority=color imagename='Survival_Plot_1_9';
title 'Product-Limit Survival Estimates';
title2 h=0.8 'With Number of AML Subjects at Risk';
proc sgplot data=how.SurvivalPlotData;
  legenditem type=marker name='c' / markerattrs=(symbol=plus color=black)
             label='Censored';
  step x=time y=survival / group=stratum name='s';
  scatter x=time y=censored / markerattrs=(symbol=plus) group=stratum;
  yaxis min=0;
  keylegend 's' / title='Disease Group: ' linelength=30;
  keylegend 'c' / location=inside position=topright;
run;
```

Figure 1.8 – Solid Survival Curves with legend title and shorter lines.

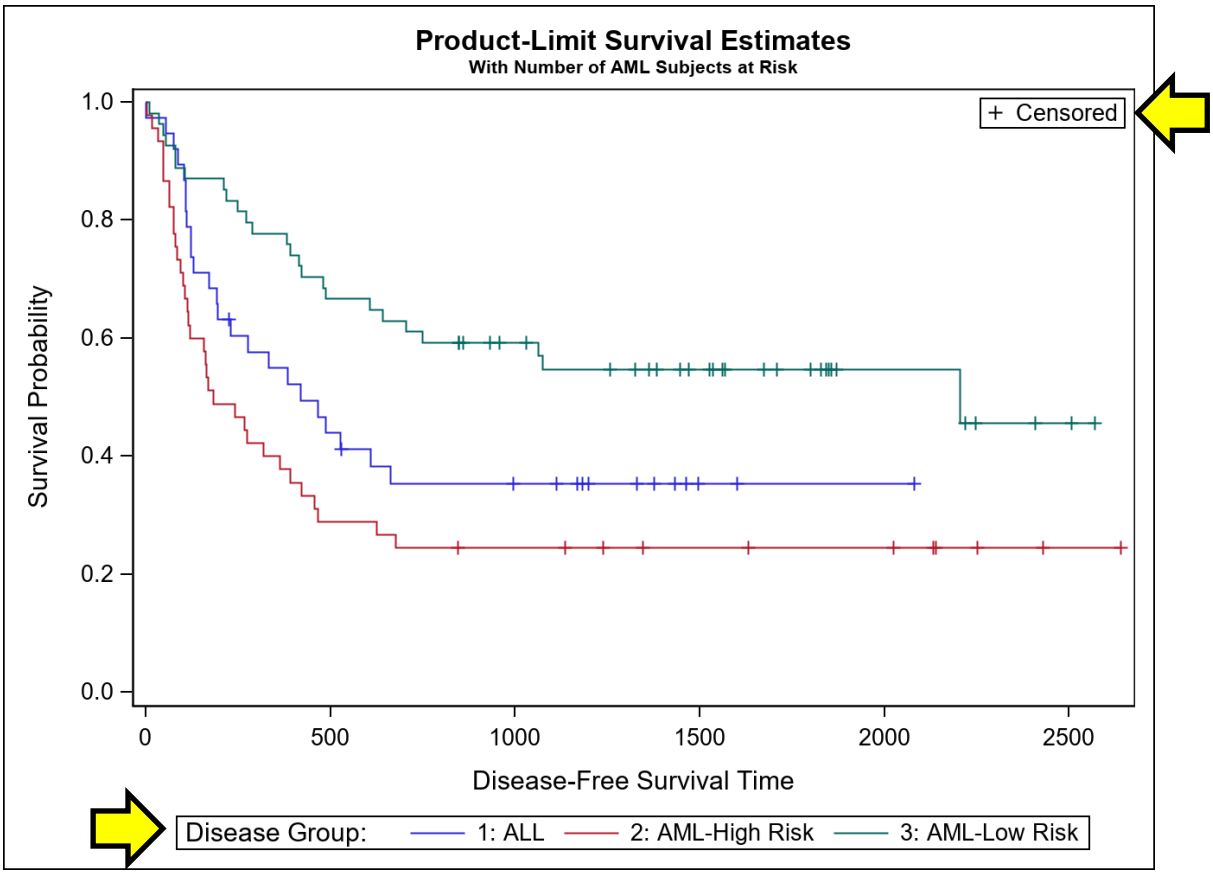


Figure 1.9 – Survival curves by time and stratum

Note the following items:

1. The censored observations are added over the survival curves.
2. A legend item is defined, with a marker shape of "+" and label of "Censored".
3. A legend is added to the plot area at the top right using the legend item.

STEP – 5

In this step we will add the display of the Subjects at Risk to the bottom of the plot data area. Starting with the code in Figure 1.8, edit or add the code shown in bold below in Figure 1.10.

```
/*--Survival plot with subjects at risk--*/
ods graphics / reset attrpriority=color imagename='Survival_Plot_1_11';
title 'Product-Limit Survival Estimates';
title2 h=0.8 'With Number of AML Subjects at Risk';
proc sgplot data=how.SurvivalPlotData;
  legenditem type=marker name='c' / markerattrs=(symbol=plus color=black)
             label='Censored';
  step x=time y=survival / group=stratum name='s';
  scatter x=time y=censored / markerattrs=(symbol=plus) GROUP=stratum;
  xaxistable atrisk / x=tatrisk class=stratum colorgroup=stratum
             location=inside valueattrs=(size=9) title="Subjects at Risk";
  yaxis min=0;
  keylegend 's' / title='Disease Group:' linelength=25 valueattrs=(size=8);
  keylegend 'c' / location=inside position=topright;
run;
```

Figure 1.10 – Solid Survival Curves with legend title and shorter lines.

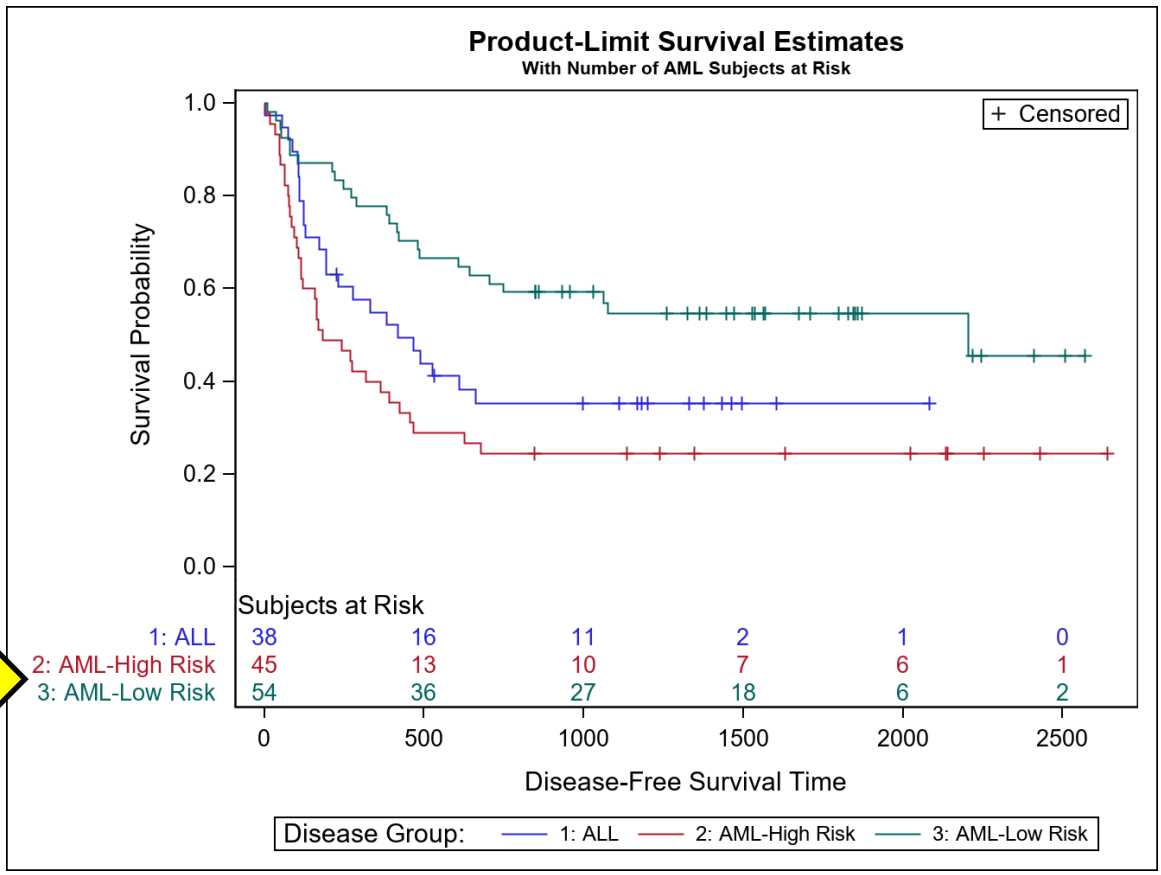


Figure 1.11 – Survival curves by time and stratum

Note the following items:

1. The table of Subjects at Risk is displayed just above the x-axis with a title.
2. Legend settings are adjusted a bit to keep the legend title on the left.

STEP – 6

In this step we will move the Subjects at Risk to the bottom of the graph, below the Legend. To do this, start with code in Figure 1.10 and edit the code shown in bold in Figure 1.12.

```
/*--Survival plot with subjects at risk--*/
ods graphics / reset attrpriority=color imagename='Survival_Plot_1_13';
title 'Product-Limit Survival Estimates';
title2 h=0.8 'With Number of AML Subjects at Risk';
proc sgplot data=how.SurvivalPlotData;
  legenditem type=marker name='c' / markerattrs=(symbol=plus color=black)
              label='Censored';
  step x=time y=survival / group=stratum name='s';
  scatter x=time y=censored / markerattrs=(symbol=plus) GROUP=stratum;
  xaxistable atrisk / x=tatrisk class=stratum colorgroup=stratum
              location=inside valueattrs=(size=9) title="Subjects at Risk";
  yaxis min=0;
  keylegend 's' / title='Disease Group:' linelength=25 valueattrs=(size=8);
  keylegend 'c' / location=inside position=topright;
run;
```

Figure 1.12 – Solid Survival Curves with legend title and shorter lines.

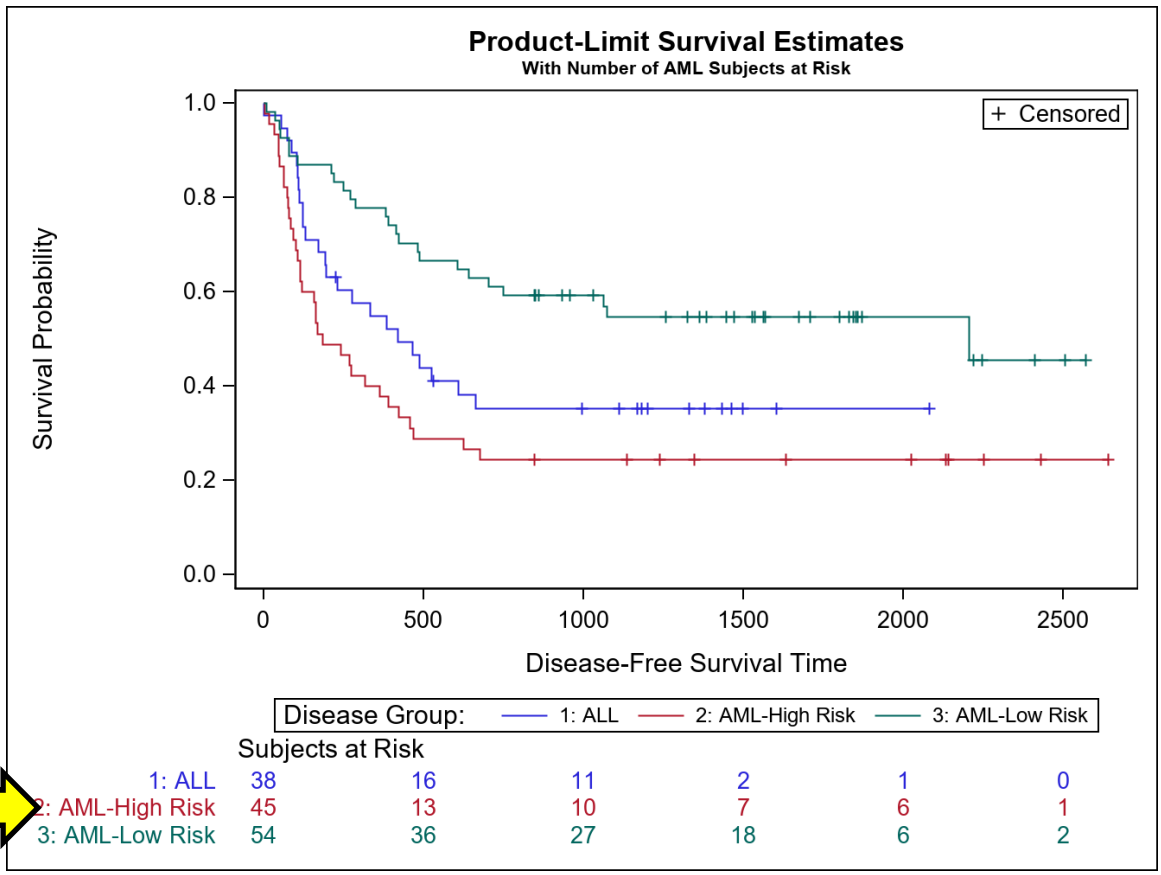


Figure 1.13 – Survival curves by time and stratum

Note the following items:

1. The table of Subjects at Risk is displayed below the Legend, in the traditional location.
2. Note, the traditional location may be due to software features, and may not be ideal.

STEP – 7

In this step we will move the Subjects at Risk to the bottom of the graph, below the Legend. To do this, start with code in Figure 1.12 and edit the code shown in bold in Figure 1.14.

```
/*--Survival plot with Journal Style--*/
ods listing style=journal2;
ods graphics / reset attrpriority=color imagename='Survival_Plot_1_15';
title 'Product-Limit Survival Estimates';
title2  h=0.8 'With Number of AML Subjects at Risk';
proc sgplot data=how.SurvivalPlotData;
  legenditem type=marker name='c' / <options>;
  step x=time y=survival / group=stratum curvelabel splitchar='-';
  scatter x=time y=censored / markerattrs=(symbol=plus) GROUP=stratum;
  xaxistable atrisk / x=tatrisk class=stratum colorgroup=stratum
            valueattrs=(size=9) title="Subjects at Risk";
  yaxis min=0;
  keylegend 's' / title='Disease Group:' linelength=25 valueattrs=(size=8);
  keylegend 'c' / location=inside position=topright;
run;
```

Figure 1.14 – Black and white Survival plot with solid lines for step plot.

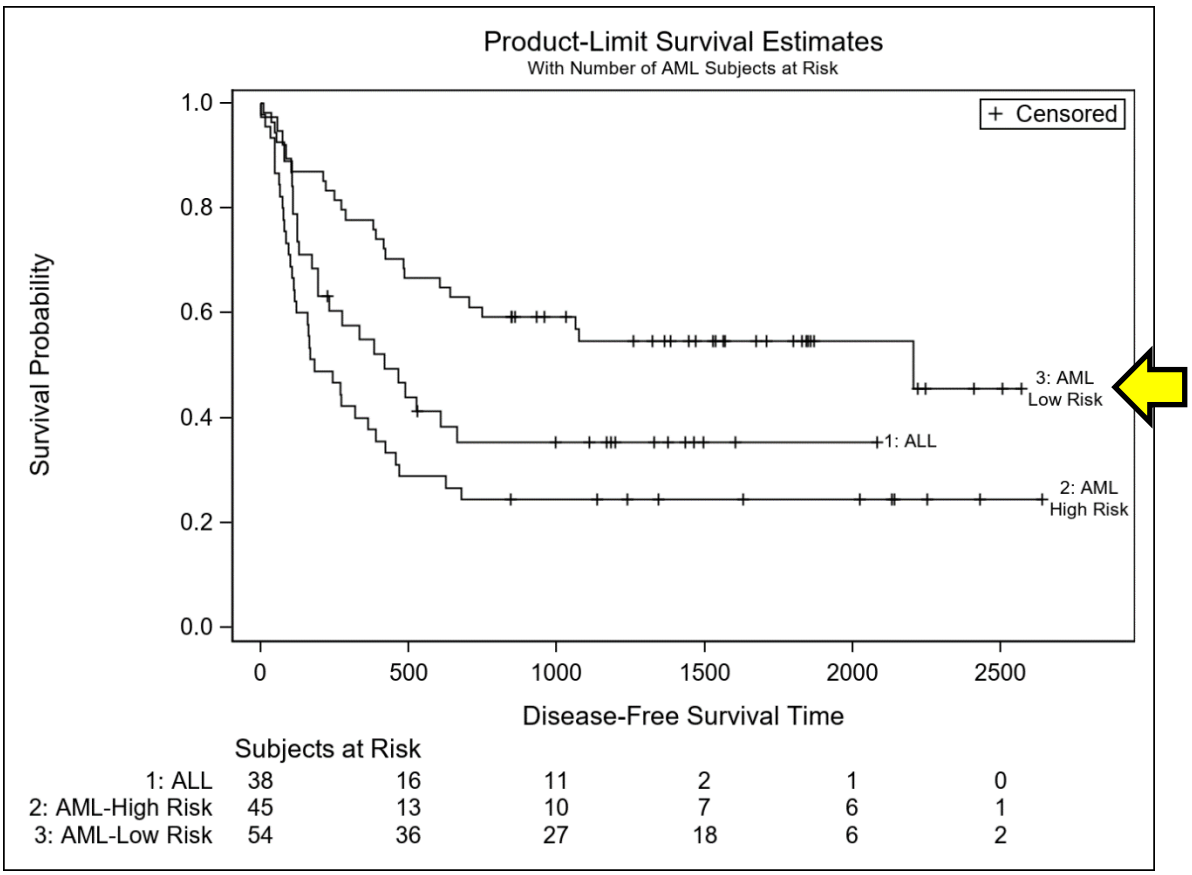


Figure 1.15 – Black and white Survival plot with solid lines for step plot.

Note the following items:

1. We use STYLE=Journal2 to render the graph in black and white with solid lines.
2. The survival curves are labeled with splitting, and the Group legend is removed.

TASK 2 – FOREST PLOT

The Forest Plot with subgroups is a very popular graph in the Health and Life Sciences domain. For this task, we will build the graph shown in figure 2.2 using the data set shown in Figure 2.1.

STEP – 1

View the subgrouped forest plot data set "**how.forest**" by submitting the code below.

```
libname how 'C:\how\matange\data';
proc print data=how.forest(firstobs=5 obs=20);
  var obsid Subgroup id countpct mean low high pcigroup
      group pvalue indentwt text1 text2 ref;
run;
```

Obs	ObsId	Subgroup	Id	countpct	Mean	Low	High	PCIGroup	Group	PValue	indentWt	text1	text2	ref
8	8	Race or ethnic group	1		.	.	.	.	.	0.52	0			8
9	9	Nonwhite	2	428(20)	1.05	0.60	1.8	18.8	17.8	.	1			9
10	10	White	2	1738(80)	1.20	0.85	1.6	16.7	15.0	.	1			10
11	11	From MI to Randomization	1		.	.	.	.	.	0.81	0			.
12	12	<= 7 days	2	963(44)	1.20	0.80	1.5	18.9	18.6	.	1			.
13	13	> 7 days	2	1203(56)	1.15	0.75	1.5	15.9	12.9	.	1			.
14	14	Diabetes	1		.	.	.	.	.	0.41	0			14
15	15	Yes	2	446(21)	1.40	0.90	2.0	29.3	23.3	.	1			15
16	16	No	2	720(79)	1.10	0.80	1.5	14.4	13.5	.	1			16
17	17		2		.	.	.	.	.	.	.	<--PCI Better		.
18	17		2		.	.	.	.	.	.	.		Therapy Better-->	.

Figure 2.1 – Data for Subgrouped Forest Plot

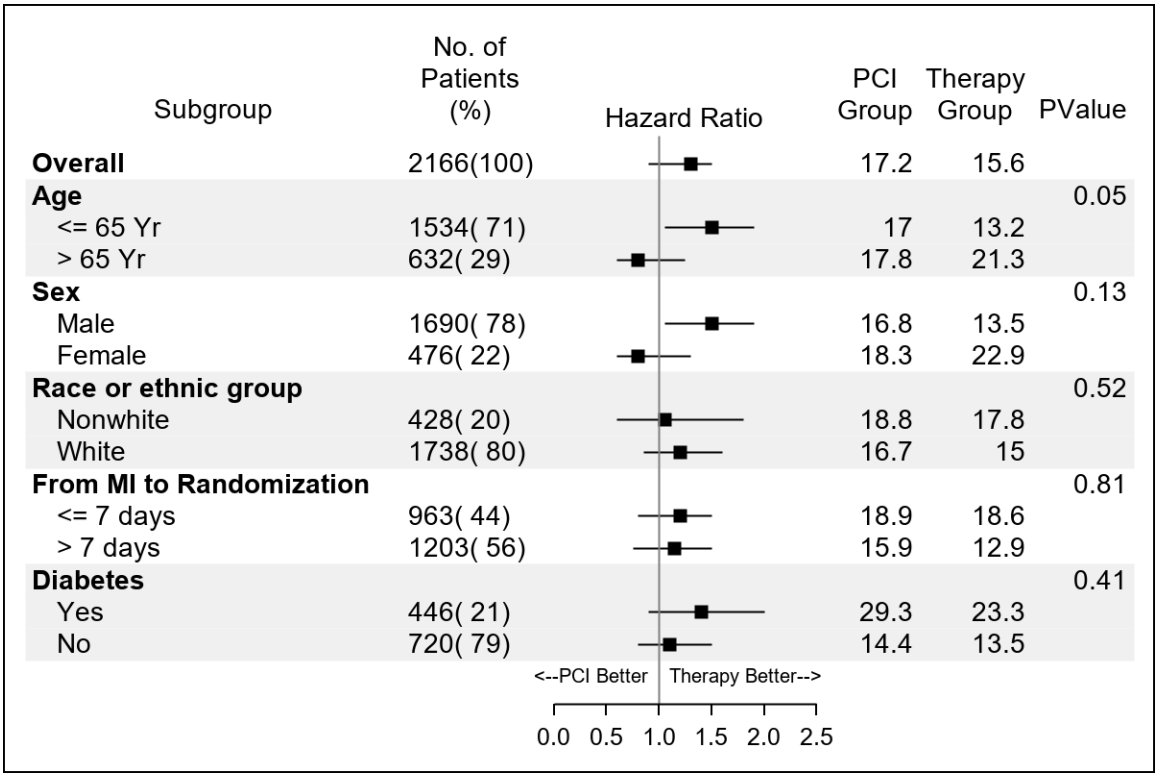


Figure 2.2 –Subgrouped Forest Plot with bands

A close study of the graph in Figure 2.2 shows the following:

1. A plot of mean, low and high hazard ratio values by subgroups in the middle of the graph.
2. Display of a column of categories by subgroups on the left, the values are indented.
3. Display of a column of the number of patients with percent values to the right of the subgroups.
4. Display of the statistics by subgroups on the right.
5. A reference line at x=1, with indication of which treatment was better.
6. Faint bands grouping the subgroup together to aid the eye across the graph.

We will create this graph a step at a time, adding each element enumerated above.

STEP – 2

Create the graph including display of the Hazard Plot in the middle. Open a new program and enter the code shown in Figure 2.3. Or, open the file "**c:\how\matange\Forest_plot**" and paste "Step 2" code.

```
ods html close;
ods listing image_dpi=200 gpath='c:\how\matange\graph';ods graphics / reset
width=6in height=3in imagename='Forest_Plot_2_4';
proc sgplot data=how.forest nowall noborder nocycleattrs noautolegend;
  highlow y=obsid low=low high=high;
  scatter y=obsid x=mean / markerattrs=(symbol=squarefilled);
  scatter y=obsid x=mean / markerattrs=(size=0) x2axis;
  refline 1 / axis=x;
  text x=xl y=obsid text=text1 / position=left contributeoffsets=none;
  text x=xl y=obsid text=text2 / position=right contributeoffsets=none;
  xaxis display=(nolabel) values=(0.0 0.5 1.0 1.5 2.0 2.5);
  x2axis label='Hazard Ratio' display=(noline noticks novalues) ;
  yaxis reverse display=none type=discrete;
run;
```

Figure 2.3 – Subgrouped Forest Plot with Hazard Ratio plot.

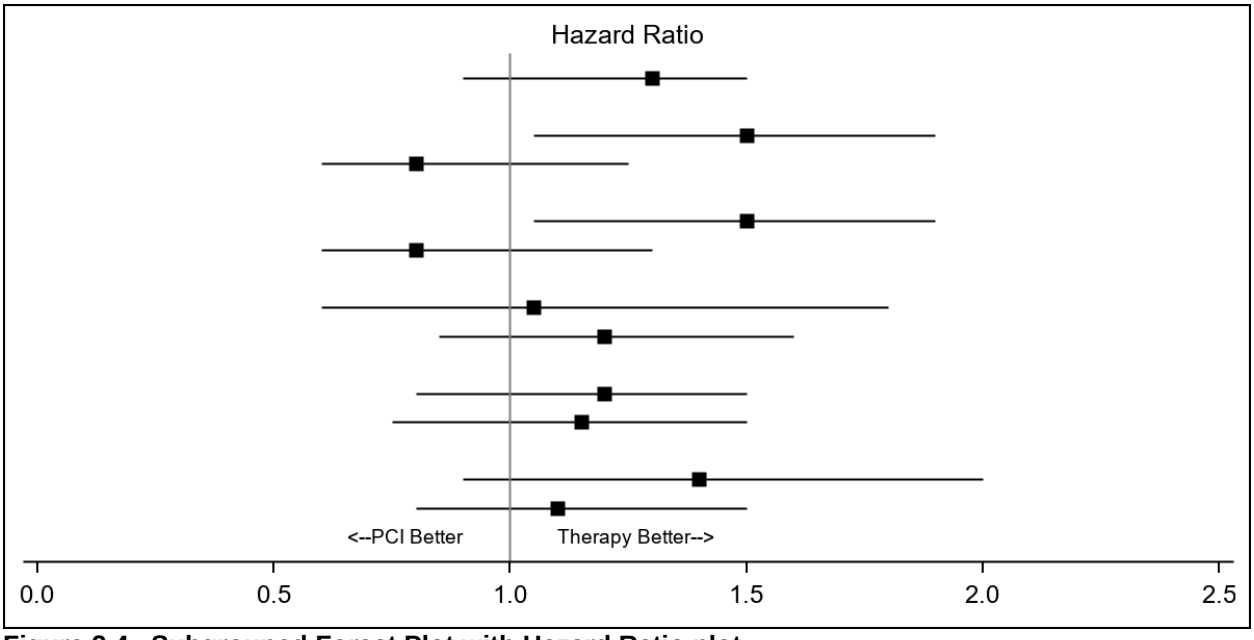


Figure 2.4 –Subgrouped Forest Plot with Hazard Ratio plot.

STEP – 3

Add the category values to the left side of the graph. Starting with the code in Figure 2.3, add the code shown in bold in Figure 2.5. Or, copy Step – 3 from file.

```
ods graphics / reset width=6in height=4in imagename='Forest_Plot_2_6';
proc sgplot data=how.forest nowall noborder nocycleattrs noautolegend;
  highlow y=obsid low=low high=high;
  scatter y=obsid x=mean / markerattrs=(symbol=squarefilled);
  scatter y=obsid x=mean / markerattrs=(size=0) x2axis;
  refline 1 / axis=x;
  text x=xl y=obsid text=text1 / position=left contributeoffsets=none;
  text x=xl y=obsid text=text2 / position=right contributeoffsets=none;
  yaxistable subgroup / location=inside position=left indentweight=indentWt
            valueattrs=(size=10) labelattrs=(size=10);
  xaxis display=(nolabel) values=(0.0 0.5 1.0 1.5 2.0 2.5);
  x2axis label='Hazard Ratio' display=(noline noticks novalues) ;
  yaxis reverse display=none type=discrete;
run;
```

Figure 2.5 – Add the Subgroup column on the left.

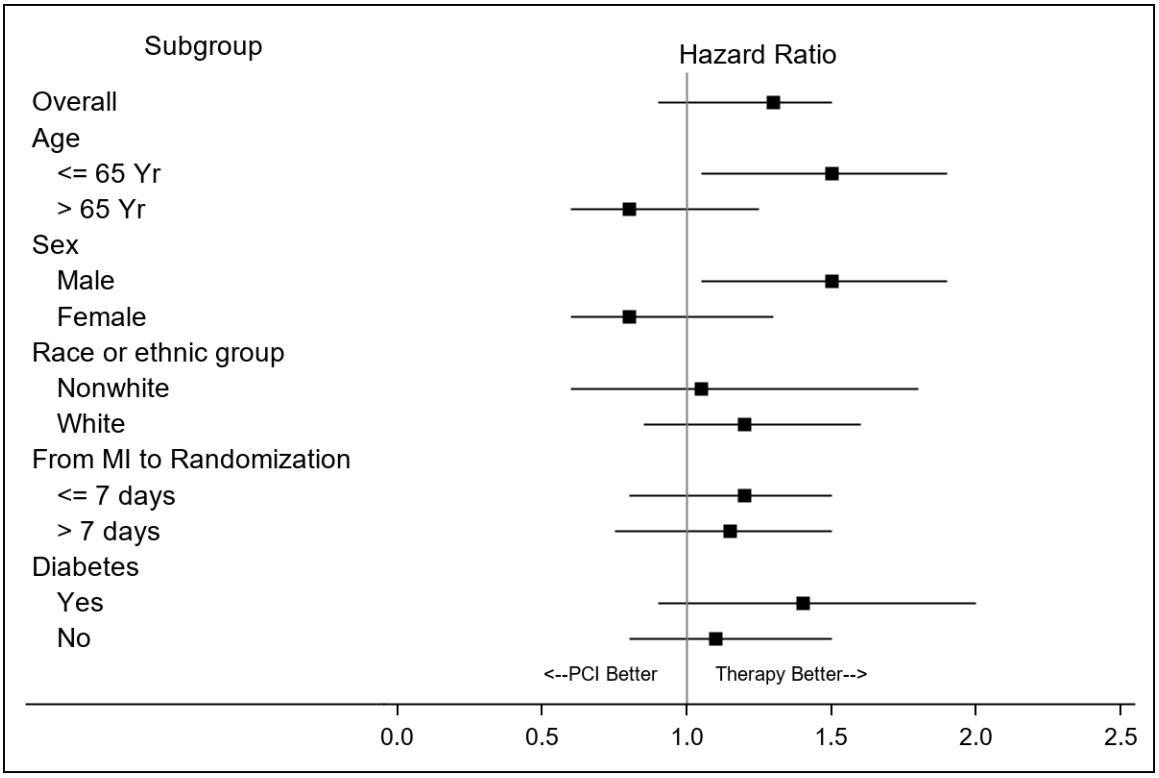


Figure 2.6 –Forest Plot with Hazard Ratio plot and subgroups on left.

Note the following features of this graph.

1. The category values are displayed on the left with subgroup labels.
2. The values are indented setting the option INDENTWEIGHT=indentWt.
3. The value and label attributes are set.
4. “PCI Better” and “Therapy better” are annotated on the x-axis using the TEXT plot.
5. Graph values and the table entries are automatically aligned using common Y=obsid.

STEP – 4

In the graph shown in Figure 2.4, we want to display the subgroup labels using a bold font and the values with a normal font. To do this, we need to define a Attributes Map data set shown in figure 2.8. The program to create the Attributes Map data set is shown in Figure 2.7.

Type in the code shown in Figure 2.7, or copy Step – 4 from file.

```
/*--Define Attrmap--*/
data attrmap;
  length textweight $10;
  id='text'; value='1'; textcolor='Black'; textweight='bold'; output;
  id='text'; value='2'; textcolor='Black'; textweight='normal'; output;
run;

proc print;
run;
```

Figure 2.7 – Add the Subgroup column on the left.

Obs	textweight	id	value	textcolor
1	bold	text	1	Black
2	normal	text	2	Black

Figure 2.8 –Attributes map data set.

Note the following features of this graph.

1. We have defined two rows for attributes using Id=text, and values of “1” and “2”.
2. Value 1 uses black color with bold font.
3. Value 2 uses black color with normal font.
4. Any font attribute can be set such as color, style, weight. Size cannot be changed as this can cause the misalignment of the rows due to varying text height.
5. We will use the DATTRMAP option in the proc statement to use this attributes map data set.
6. We will use the TEXTGROUPID and TEXTGROUP to associate the values.

STEP – 5

Adjust the text attributes of the category values on the left using the discrete attributes map defined in Step - 4. Starting with the code in Figure 2.5, add or edit the code shown in bold in Figure 2.9. Or, copy Step – 5 from file.

```
ods graphics / reset width=6in height=4in imagename='Forest_Plot_2_10';
proc sgplot data=how.forest nowall noborder nocycleattrs noautolegend
     dattrmap=attrmap;
  highlow y=obsid low=low high=high;
  scatter y=obsid x=mean / markerattrs=(symbol=squarefilled);
  scatter y=obsid x=mean / markerattrs=(size=0) x2axis;
  refline 1 / axis=x;
  text x=xl y=obsid text=text1 / position=left contributeoffsets=none;
  text x=xl y=obsid text=text2 / position=right contributeoffsets=none;
  yaxistable subgroup / location=inside textgroupid=text textgroup=id
        position=left indentweight=indentWt
        valueattrs=(size=10) labelattrs=(size=10);
  xaxis display=(nolabel) values=(0.0 0.5 1.0 1.5 2.0 2.5);
  x2axis label='Hazard Ratio' display=(noline noticks novalues) ;
  yaxis reverse display=none type=discrete;
run;
```

Figure 2.9 – Add the Subgroup column on the left.

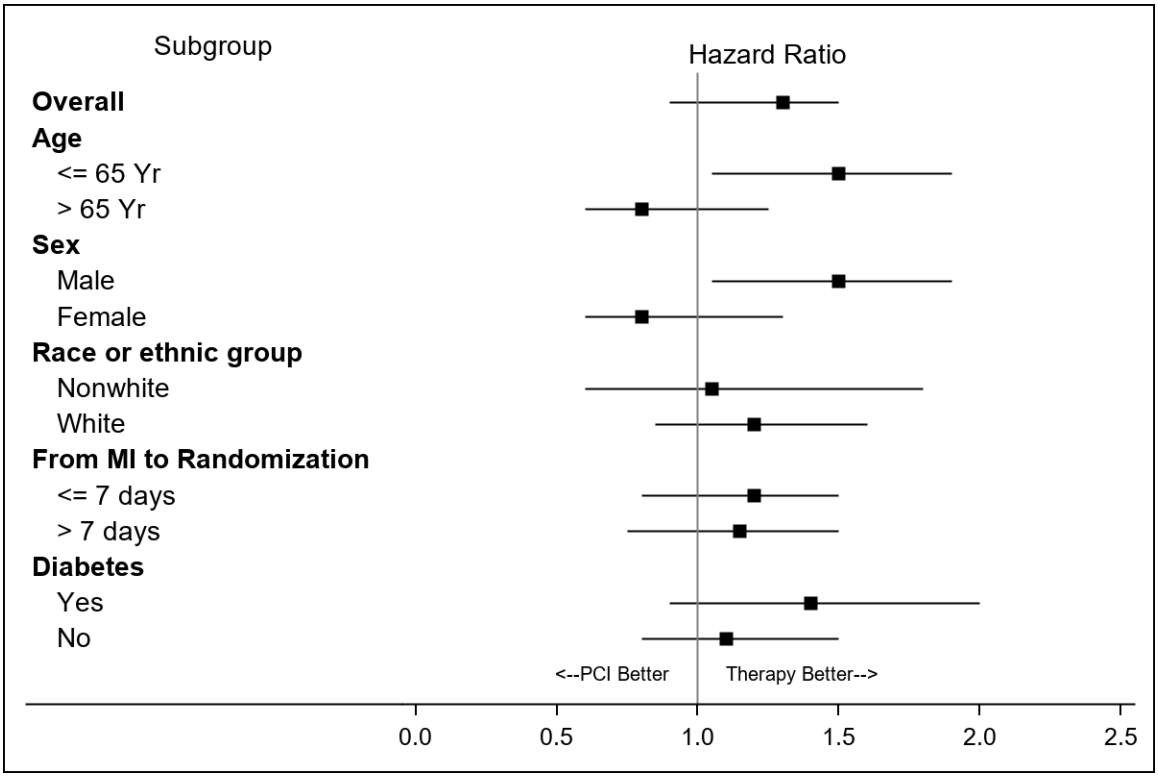


Figure 2.10 – Forest Plot with subgroups on left.

Note the following features of this graph.

1. The attributes map data set defined in Step – 4 is used in here with the DATTRMAP option.
2. The rows with ID=1 are displayed using a bold font as defined in the attributes map.
3. The rows with ID=2 are displayed using a normal font as defined in the attributes map.

STEP – 6

Add the column for number of patients with percent values to the right of the Subgroup column. Starting with the code in Figure 2.9, add or edit the code shown in bold in Figure 2.11. Or, copy Step – 6 from file.

```
ods graphics / reset width=6in height=4in imagename='Forest_Plot_2_12';
proc sgplot data=how.forest nowall noborder nocycleattrs noautolegend
dattrmap=attrmap;
  highlow y=obsid low=low high=high;
  scatter y=obsid x=mean / markerattrs=(symbol=squarefilled);
  scatter y=obsid x=mean / markerattrs=(size=0) x2axis;
  refline 1 / axis=x;
  text x=xl y=obsid text=text1 / position=left contributeoffsets=none;
  text x=xl y=obsid text=text2 / position=right contributeoffsets=none;
  yaxistable subgroup  / location=inside textgroup=id textgroupid=text
            position=left indentweight=indentWt
            valueattrs=(size=10) labelattrs=(size=10);
  yaxistable countpct / location=inside position=left
            valueattrs=(size=10) labelattrs=(size=10);
  xaxis display=(nolabel) values=(0.0 0.5 1.0 1.5 2.0 2.5);
  x2axis label='Hazard Ratio' display=(noline noticks novalues) ;
  yaxis reverse display=none type=discrete;
run;
```

Figure 2.11 – Add the column for number of patients and percentages.

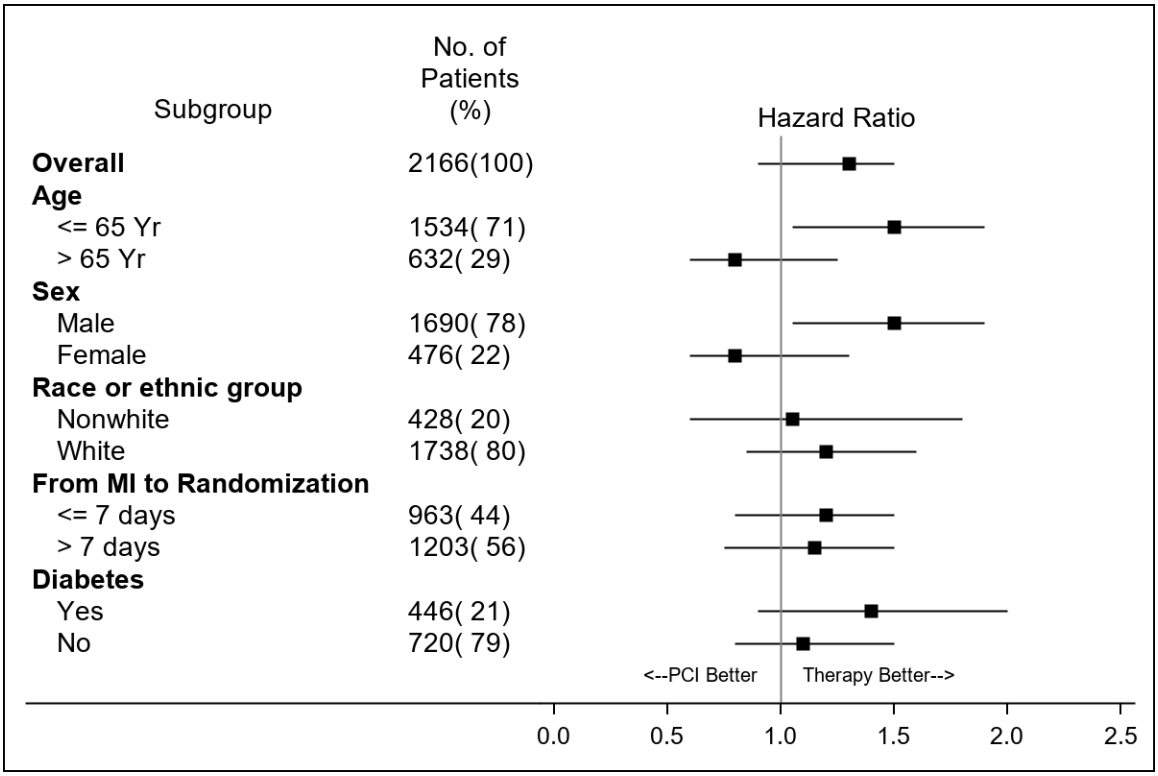


Figure 2.12 – Forest Plot with column for number of patients and percentages.

Note the following features of this graph.

1. The column showing the number of patients and percent in the group is displayed.

STEP – 7

Add the columns for various statistics on the right. Starting with the code in Figure 2.11, add or edit the code shown in bold in Figure 2.13. Or, copy Step – 7 from file.

```
ods graphics / reset width=6in height=4in imagename='Forest_Plot_2_14';
proc sgplot data=how.forest nowall noborder nocycleattrs noautolegend
dattrmap=attrmap;
  styleattrs axisextent=data;
  highlow y=obsid low=low high=high;
  scatter y=obsid x=mean / markerattrs=(symbol=squarefilled);
  scatter y=obsid x=mean / markerattrs=(size=0) x2axis;
  refline 1 / axis=x;
  text x=xl y=obsid text=text1 / position=left contributeoffsets=none;
  text x=xl y=obsid text=text2 / position=right contributeoffsets=none;
  yaxistable subgroup  / location=inside textgroup=id textgroupid=text
            position=left indentweight=indentWt <options>;
  yaxistable countpct / location=inside position=left <options>;
  yaxistable PCIGroup group pvalue / location=inside position=right
            nomissingchar valueattrs=(size=10) labelattrs=(size=10);
  xaxis display=(nolabel) values=(0.0 0.5 1.0 1.5 2.0 2.5);
  x2axis label='Hazard Ratio' display=(noline noticks novalues) ;
  yaxis reverse display=none type=discrete;
run;
```

Figure 2.13 – Add the statistics columns on the right.

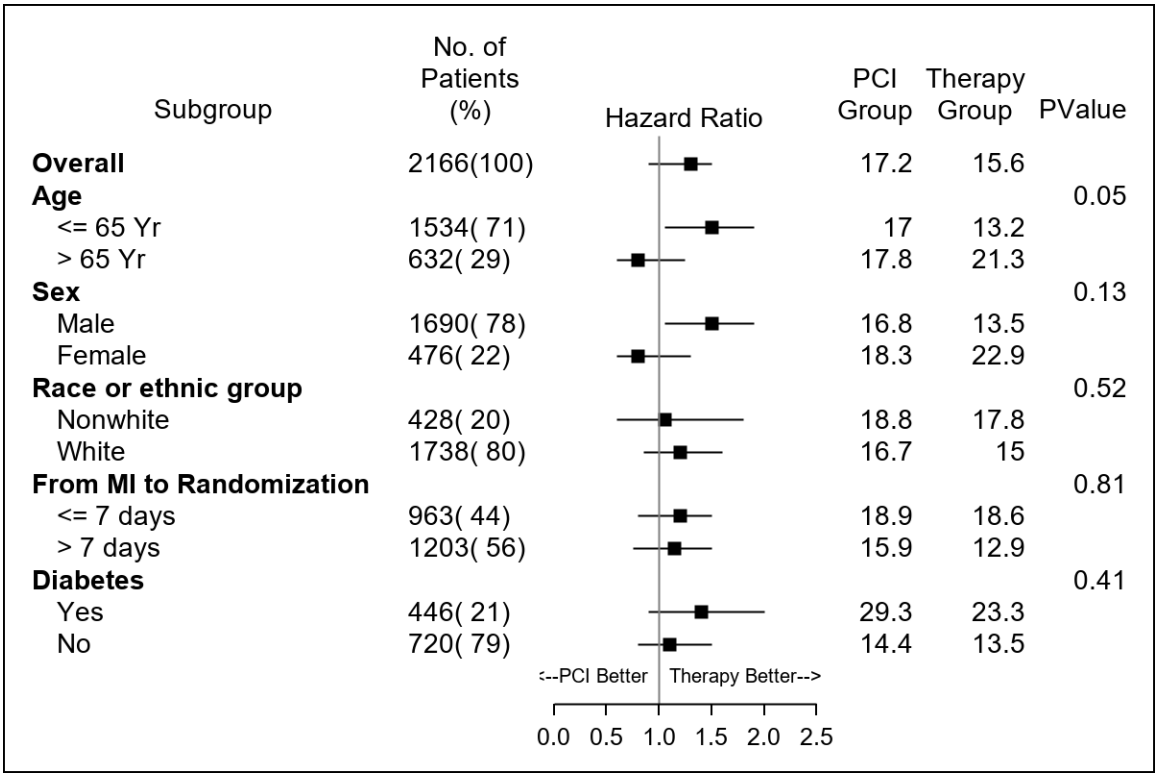


Figure 2.14 – Forest Plot with statistics columns on the right.

Note the following features of this graph.

1. The column showing the number of patients and percent in the group is displayed.
2. Note, the x-axis line now extends only as far as the data, and not all the way across the tables.

STEP – 8

Add horizontal bands to help understand the graph better. Starting with the code in Figure 2.13, add or edit the code shown in bold in Figure 2.15. Or, copy Step – 8 from file.

```
ods graphics / reset width=6in height=4in imagename='Forest_Plot_2_16';
proc sgplot data=how.forest nowall noborder nocycleattrs noautolegend
dattrmap=attrmap;
  styleattrs axisextent=data;
  refline ref / discretethickness=1 lineattrs=(color=cxf0f0f0);
  highlow y=obsid low=low high=high;
  scatter y=obsid x=mean / markerattrs=(symbol=squarefilled);
  scatter y=obsid x=mean / markerattrs=(size=0) x2axis;
  refline 1 / axis=x;
  text x=xl y=obsid text=text1 / position=left contributeoffsets=none;
  text x=xl y=obsid text=text2 / position=right contributeoffsets=none;
  yaxistable subgroup  / location=inside textgroup=id <options>;
  yaxistable countpct / location=inside position=left <options>;
  yaxistable PCIGroup group pvalue / location=inside <options>;
  xaxis display=(nolabel) values=(0.0 0.5 1.0 1.5 2.0 2.5);
  x2axis label='Hazard Ratio' display=(noline noticks novalues) ;
  yaxis reverse display=none type=discrete colorbands=odd
          colorbandsattrs=(transparency=1);
run;
```

Figure 2.15 – Add the statistics columns on the right.

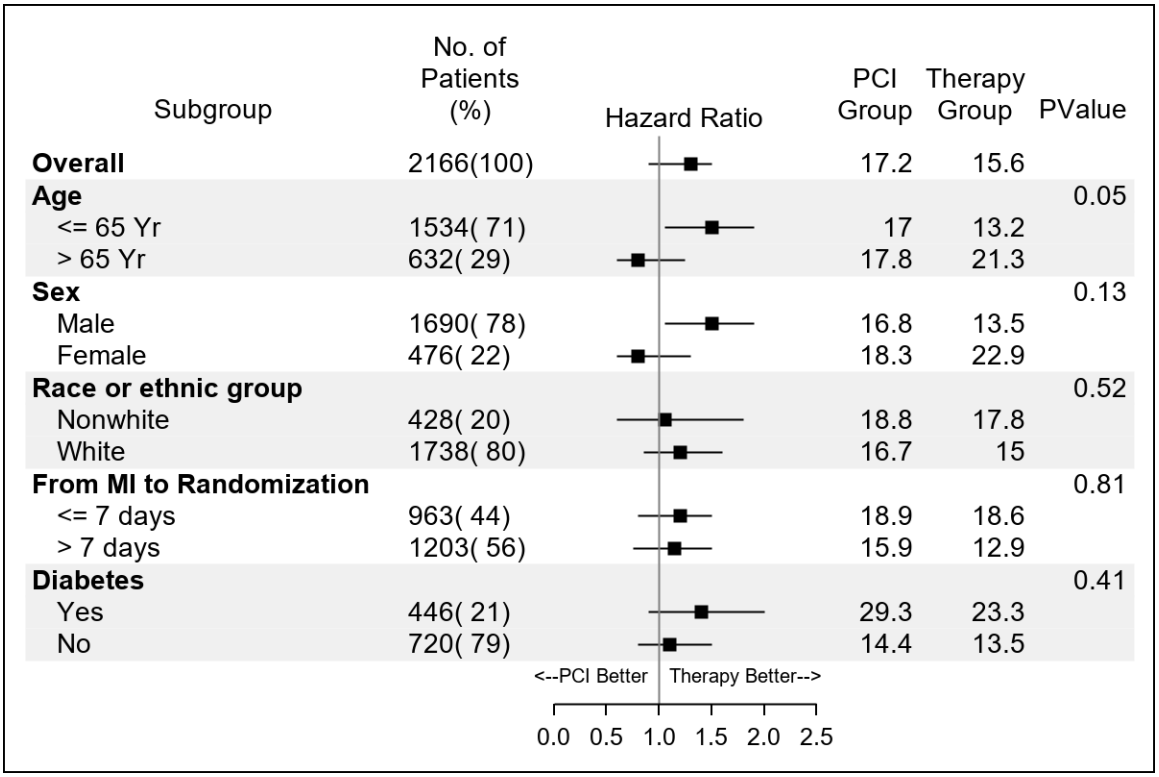


Figure 2.16 – Forest Plot with statistics columns on the right.

Note the following features of this graph.

1. Light gray bands are displayed for alternate groups to help understand the graph.
2. Not the setting of COLORBANDS option to allow the refline bands to extend all the way.

CONCLUSION

The SGPLOT procedure provides us the features necessary to create frequently requested clinical graphs such as the Survival Plot or the Subgrouped Forest Plot. These can be easily created by using the layering feature of the procedure.

Analyze the graph you want to create and break the process into steps as shown in Task 1 and Task 2. Then, layer the appropriate plot statements with the available options to create the graph.

SAS 9,.4 includes the xAxisTable and the yAxisTable statements that make it easy to add rows or columns of textual information to the graph, aligned with the x or y axis.

REFERENCES

Matange, Sanjay 2018. "Advanced Graphs using Axis Tables" *SAS Global Forum, April 2018*.

RECOMMENDED READING

Graphically Speaking Blog Articles: https://blogs.sas.com/content/graphicallyspeaking. Sanjay Matange.

CONTACT INFORMATION

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

Author Name	Sanjay Matange
Address	109 Flying Leaf Ct.
City / Postcode	Cary, NC 27513
Email:	SanjayMatange@gmail.com
Web:	https://make-a-graph.com/

Brand and product names are trademarks of their respective companies.