PharmaSUG China 2016 - 10

Create a Graph Template on the Fly

Kathy Chen, Pharmacyclics LLC, An Abbvie Company, Sunnyvale, CA, U.S.A

ABSTRACT

While Proc Template Graphics has enabled us to create customized graphs, the tendency is to have many static templates with minor differences. Instead, it makes more sense to create a template to work with your data in real time. Creating a Graphic Template on the fly based on actual data can greatly improve programming efficiency. This paper will show you how to dynamically create a graphic template.

INTRODUCTION

Dynamic data visualizations can be interpreted more easily than numbers in summary tables and thereby prove to be extremely valuable in clinical studies. We have seen more and more people from different functions start requesting graphs for many different things. While SAS keeps improving its graphic capability over the past 5-6 years, to be able to catch up with the flow and use these tools effectively is a challenge. Since so many people fall in love with PROC Template Graphics for its flexibility, some choose to use it for almost everything. So this paper will use Kaplan-Meier plot as an example to show how to create a graphic template dynamically.

PROC TEMPLATE GRAPHICS

The graphic template KMOS below is the template used to create multiple Kaplan-Meier plots upon request. It can not only plot the survival curve, it can also provide some critical statistical analysis results.

Here is how statgraph kmos is constructed:

```
Proc template;
    define statgraph kmos;      # define graph layout,   graphical content, footnote
          begingraph;
                --- define macro variables
                --- define color map
                layout lattice / …
                      layout overlay;
                            stepplot x=_TIME y=_SURVIAL/group=treatmentmarkers…
                            scatterplot x=_TIME2 y=_CENSORED/group=…
                      endlayout;
                      --- at risk plot here
                 endlayout;
          endgraph;
    end;
```

You can see it contains a color map if you want some treatment groups in a certain fixed color. It also allows you to have an at risk plot at the bottom of the graph. And if an inside statistical summary table is needed, the graph can have that too. For the three following graphs, if you use proc template graphs straight, you will need three graph templates. When the duration of the study is different, you will have to change the template again.

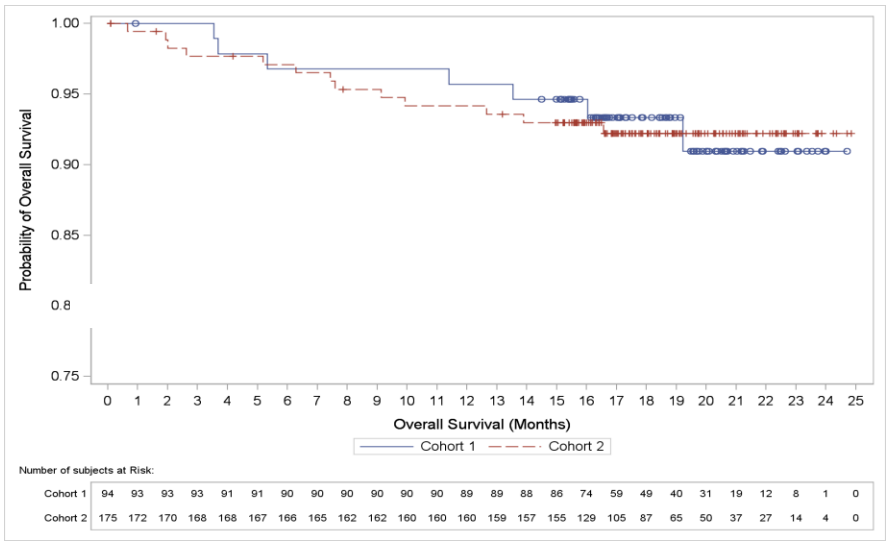


Figure 1. This is a Kaplan-Meier Overall Survival plot with two treatment groups

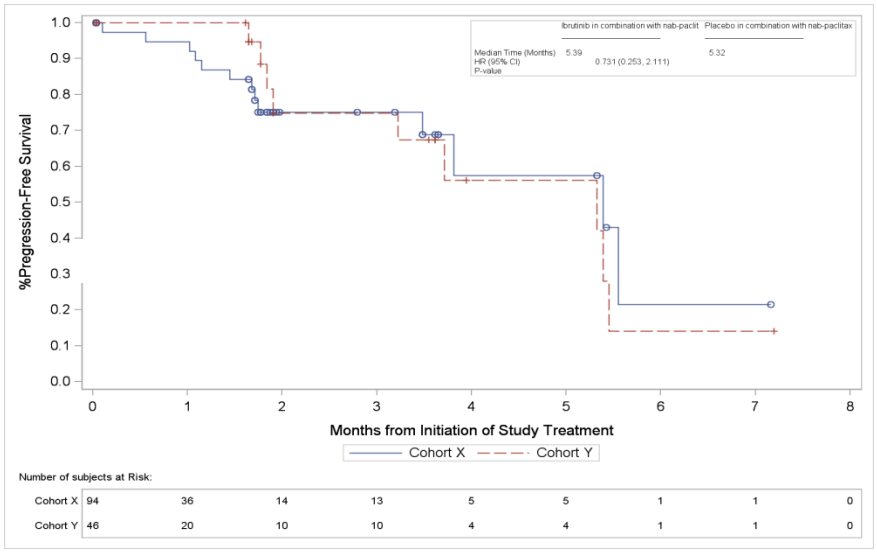


Figure 2. This is a Progression-Free Survival plot with two treatment groups and an inset summary table

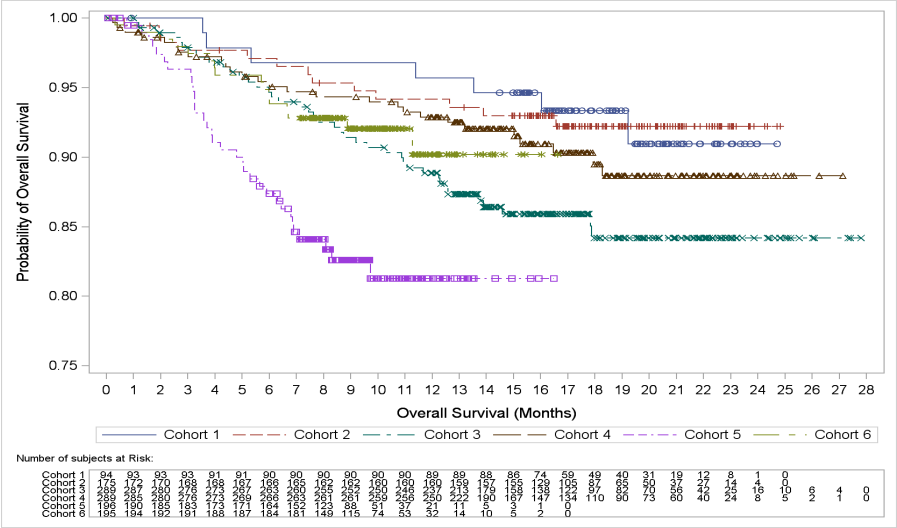


Figure 3. This is an Overall Survival plot with 6 cohorts

CREATE LOCAL AND GLOBAL MACRO VARIABLES

In order to be able to use one macro/graph template to create multiple graphs dynamically, we need to use macro variables of proc template to support it. Proc template language has its own way of creating macro variables and DYNAMIC, MVAR and NMVAR are used to create global, character and numeric macro variables. The code below is an example how to create a series of character macro variables dynamically if you want to display a summary table for each treatment group. Dynamic is the key word for creating global macro variables.

```
mvar  %do i=1 %to &ncols;
# used for embedded summary table

  trtn&i n&i dr&i kmm&i


%end;;


dynamic _TIME _SURVIVAL _STRATUM _TIME2 _CENSORED _STRATUM2 _TATRISK _ATRISK
_STRATUM3;
```

CREATE COLOR MAP

If you want to have a consistent look for each graph with one fixed color for each treatment group, a color map is needed. Otherwise the default color will be picked by the system automatically.

```
discreteattrmap name='treatment'/ignorecase=true;
value "CONTROL"/lineattrs=graphdata1(color=CX000000 pattern=shortdash);
    # Used for graphic lines
value "IBRUTINIB"/lineattrs=graphdata2 (color=CX000000 pattern=solid);
enddiscreteattrmap;
```

CREATE SURVIVAL PLOT

You can see from below how the macro variables are used in graph configuration.

```
xaxisopts=(label=('Overall Survival (Months)') griddisplay=off linearopts=(
tickvaluesequence=(start=0.0 end=30.0 increment=1.0)))

yaxisopts=(label=('Probability of Overall Survival')  linearopts=(
viewmin=&vmin. tickvaluesequence=(start=&vmin. end=1.0 increment=&incrmt.)));
```

```
stepplot x=_TIME y=_SURVIVAL / group=treatmentmarkers  name='step'
connectorder=xaxis;

scatterplot x=_TIME2 y=_CENSORED / group=_STRATUM2 name='censored'
markerattrs=(symbol=PLUS ) legendlabel="Censored";


discretelegend 'censored'/ location=inside autoalign=(bottomleft topright);

discretelegend 'step' / location=outside;
```

CREATE EMBEDDED SUMMARY TABLE

If you want a summary table to list out the events for each group, you can setup an inside table this way.

```
%if &intbl ne %then %do;

      layout gridded /columns=%eval(&ncols.+1) border=true autoalign=(&loc.
topright topleft) columngutter=0 ;

       entry textattrs=(size=6pt) halign=left "                    ";

       %do i=1 %to &ncols;

             entry textattrs=(size=6pt) halign=left trtn&i     ;

       %end;;

       entry textattrs=(size=6pt) halign=left "             ";

       %do i=1 %to &ncols;

             entry textattrs=(size=6pt) halign=left "_____________________";

       %end;;

        entry textattrs=(size=6pt) halign=left "            ";

       %do i=1 %to &ncols;

             entry textattrs=(size=6pt) halign=left "       ";

       %end;;
```

CREATE SUBJECT AT RISK TABLE

An at risk table is commonly used together with a survival plot, so the following code can help populate the at risk table.

```
layout overlay / /*walldisplay=none*/ xaxisopts=(type=linear display=none
linearopts=( tickvaluesequence=( start=0.0 end=30.0 increment=1.0)));


blockplot x=_TATRISK block=_ATRISK / repeatedvalues=true class=_STRATUM3
name='Number of subjects at Risk' display=(VALUES LABEL ) filltype=alternate
valuehalign=start valueattrs=GRAPHDATATEXT(size=7)
labelattrs=GRAPHVALUETEXT(size=7);


entry halign=left 'Number of subjects at Risk:' /valign=top location=outside
textattrs=GraphText(size=7 style=normal weight=normal);

endlayout;
```

RENDER THE GRAPH

Once graph template is defined, sgrender procedure is used to bind the data to it and create the graphs planned.

```
proc sgrender data=work.survg  template=kmos;
```

```
        dynamic _TIME="TIME" _SURVIVAL="SURVIVAL" _STRATUM="STRATUM"
                _TIME2="TIME" _CENSORED="CENSORED" _STRATUM2="STRATUM"
                _TATRISK="TATRISK" _ATRISK="ATRISK" _STRATUM3="STRATUM" ;
    run;
```

DRIVER PROGRAM

Once the graph template is built, we can start using it. However in order to make graph creation more efficient, I create a macro **kmplot** which contains a few macros, one for each function needed, so I can create multiple K-M plots with one program by different macro calls dynamically.

MACROS USED

1. get_surv() to obtain median survival estimate from proc lifetest
2. get_risk() to get hazard ratio and CI from proc phreg
3. get_pval() to get log-rank p-value from proc lifetest
4. get_survg() to get the survival plot data from proc lifetest.
5. get_kmos() to create the final K-M plot

DATASET FOR INSIDE TABLE

inside table - f_km_os_xxxx

Obs	ROWLBL	COL1	COL2	VARNAME	LEVEL	_DEN1
1	Death rate			EVNTDESN	0	95
2	Death	6 (6.3%)	11 (6.2%)	EVNTDESN	1	95
3				EVNTDESN	999	95
4	K-M estimate (Months)			AVAL	0	.
5	Median	n/a	n/a	AVAL	1	.
6				AVAL	999	.
7	Stratified Hazard Ratio (95% CI)[1]			AVALH	1	.
8				AVALH	999	.
9	Stratified Log-Rank test p-value			AVALP	999	.

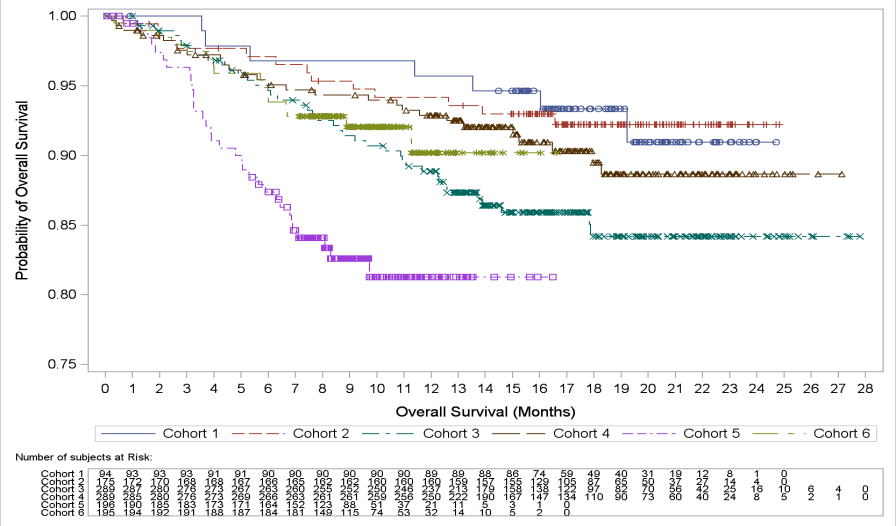
Obs	_DEN2	_NUM1	_NUM2	COL3	PARAMETER	CLASSVAL0	DF	ESTIMATE	STDERR
1	178	6	11		.	.	.		
2	178	6	11		.	.	.		
3	178	6	11		.	.	.		
4	.	.	.		.	.	.		
5	.	.	.		.	.	.		
6	.	.	.		.	.	.		
7	.	.	.	1.161 (0.420, 3.210)	TRTN	Ibrutinib	1	0.14907	0.51893
8	.	.	.		TRTN	Ibrutinib	1	0.14907	0.51893
9	.	.	.	0.7737		.	.	.	

Obs	CHISQ	PROBCHISQ	HAZARDRATIO	HRLOWERCL	HRUPPERCL	LABEL	PAGE
1	.	.	.	.	.		1
2	.	.	.	.	.		1
3	.	.	.	.	.		1
4	.	.	.	.	.		1
5	.	.	.	.	.		1
6	.	.	.	.	.		1
7	0.0825	0.7739	1.161	0.420	3.210	TRTN Ibrutinib	1
8	0.0825	0.7739	1.161	0.420	3.210	TRTN Ibrutinib	1
9	.	.	.	.	.		1

MACRO CALL

Now we can create the graphs with **kmplot** macro dynamically. For the graph with six cohorts, I use **gcols** to subset the cohorts needed. Since I don't need a summary table inside of this graph, I used **NO** for parameter **intbl** in the macro call.

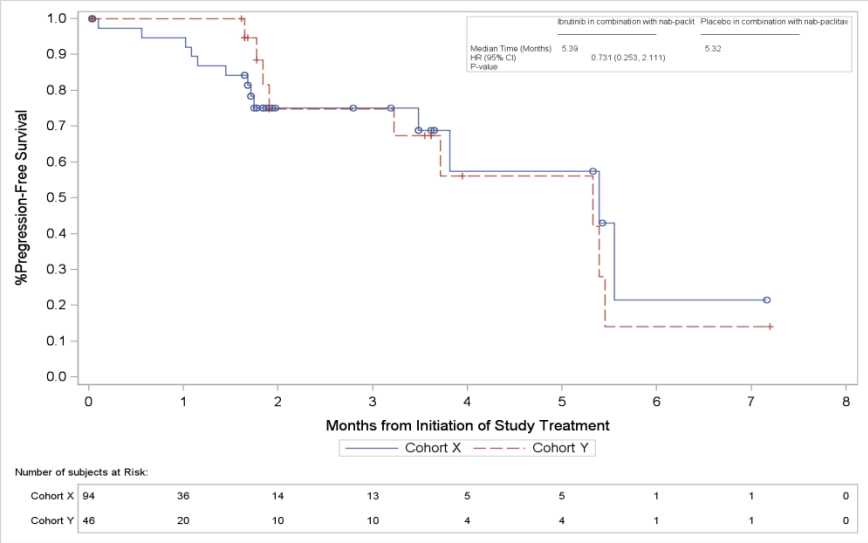
```
%let gcols= 1 2 3 4 5 6;
%kmplot(param="OS",
          prg=f_km_os,
          pop=ittfl,
        strata=,
          vmin=0.75,
        incrmt=0.05,
         intbl=NO,
        rngvar=0 to 30 by 1,
      timepts=0 3 6 9 12 15 18 21 24 27);
```



This is the graph produced by the macro call above.

If you need a graph with two treatment arms plus a summary table included inside, all you need to do is to set the **gcols** to the ones you want to report and use **YES** for parameter **intbl** in the macro call. If these two lines are very close to each other, you can change the starting point of Y axis by set vmin to a different value etc.

```
%let gcols= 3 4;
%kmplot(param="OS",
          prg=f_km_os,
          pop=ittfl,
        strata=,
          vmin=0.75,
        incrmt=0.05,
         intbl=YES,
        rngvar=0 to 30 by 1,
      timepts=0 3 6 9 12 15 18 21 24 27);
```



This is the graph produced by the macro call above.

CONCLUSION

When a new tool becomes available, to keep learning new things about it is important but to be able to understand what it is capable of and use it efficiently is even more important. We should keep our curiosity alive, work together so we can make programming support more efficient.

REFERENCES

Creating and Customizing the Kaplan-Meier Survival Plot in PROC LIFETEST<Warren F. Kuhfeld and Ying So>. <SAS Institute Inc.> https://support.sas.com/resources/papers/proceedings13/427-2013.pdf

ACKNOWLEDGMENTS

Pete Pilliod (Pete Anderson) who inspires me to learn more about utility macros.

Statistical Programming Colleagues at Pharmacyclics.

RECOMMENDED READING

Customizing the Graph Templates for a Kaplan-Meier Failure Plot < Hugh Geary>. < Novella Clinical, a Quintiles company> http://www.pharmasug.org/proceedings/2015/QT/PharmaSUG-2015-QT47.pdf

CONTACT INFORMATION

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

Name: Kathy Chen
Enterprise: Pharmacyclics LLC, An Abbvie Company
Address: 999 E Arques Ave,
City, State ZIP: Sunnyvale, CA 94085
Work Phone: 408-990-7333
Fax:
E-mail: qchen@pcyc.com
Web:
Twitter: