Paper 280-2012

Innovative uses of ODS and GTL

Qinghua (Kathy) Chen, Genentech, a member of Roche family, South San Francisco, CA

ABSTRACT

Graphical representation of treatment effect is often easier to understand and is therefore frequently used in conjunction with summary reports. The Output Delivery System (ODS) Graphics Template Language (GTL) when used with a data step or PROC SGRENDER procedure allows you to design your own layout so you can easily incorporate statistical graphics into summary reports. ODS Graphics have facilitated the automatic creation of statistical graphics and that makes SAS® a more viable tool for rapid production of graphics. This paper will describe how to use ODS Graphics and GTL to effectively report study results with dynamic visualization of data.

Topics include:

- Customizing summary reports by incorporating graphics using GTL
- Making graphics less restricted to input data by using GTL EVAL function
- Creating statistical graphs from SAS procedures with ODS graphs
- Editing graphs

INTRODUCTION

Dynamic data visualizations can usually be interpreted much more quickly than numbers in summary tables and thereby prove to be extremely valuable. For many types of analyses such as survival analysis, there are often two sets of output: a summary table of statistics and a separate graph. In an effort to improve readability, the concept of presenting a graph and a summary report together was thought to be an ideal solution although it was not readily achievable. For example, the creation of a forest-plot was only possible using an annotated data set in previous versions of SAS software. Unfortunately, while the end product was visually appealing, it was not very flexible in terms of different layouts and the learning curve was quite steep.

Beginning with version 9.2, SAS® provides innovative new ways of creating graphics. The ODS graphics language enables us to create high quality graphs and control the location, name of the output, graph dimensions and output types easily. GTL, the extension of PROC TEMPLATE, utilizes STATGRAPH templates to define our own graphics layout and style. Now, incorporating a graph into a SAS summary report is much easier than it previously was using an annotated data set. When ODS Graphics is enabled, more than 80 statistical procedures that support ODS Graphics can create appropriate graphs, either by default or by selecting different procedure options to generate specific graphs. Therefore it increases our programming efficiency.

CUSTOMIZING SUMMARY REPORTS BY INCORPORATING GRAPHICS USING GTL

There are four new SAS/GRAPH® procedures for statistical graphics, SGPLOT, SGPANEL, SGSCATTER and SGREDER, and each of them has their own purpose. Unlike the first three, the SGRENDER procedure allows you to create graphs based on templates you create. Since it uses GTL, it is the most flexible procedure. The ODS graphics statement controls the physical aspects of the graph, such as image size, name of the output, etc., and it affects all the graphs that are rendered in a SAS session until overridden by another ODS graphics statement. The example below shows you how SGRENDER is used to execute a template and direct a .pdf output to a predefined "pdf" directory with foreststyle style defined in PROC TEMPLATE.

STATGRAPH and STYLE are two of many templates that PROC TEMPLATE supports. The STATGRAPH template defines the layout and detail of the output to be produced. It includes statements for graph layout (lattice or overlay,) plot type (scatter plots or histograms,) text elements (titles, footnotes and insets) etc. The STYLE template defines the format information for the visual SAS output. GTL is a compiling language, so it will accept runtime variable substitution via dynamics or macro variables. For STATGRAPH such variables should be declared within the template definition before the BEGINGRAPH block. Each of the DYNAMIC, MVAR, and NMVAR statements can define multiple variables and values for dynamics that resolve to column names or strings should be quoted. The current symbol table (local or global) will be used for macro variable values at runtime.

There are three basic steps to create a graph using GTL.

1. Use PROC TEMPLATE to define a STATGRAPH template
2. Define STYLE template
3. Create the graph by running the SGRENDER procedure with appropriate data

EXAMPLE 1: **CREATING A FOREST-PLOT IN A SUMMARY REPORT.**

1. Use PROC TEMPLATE to define a STATGRAPH template

Define STATGRAPH ForestPlot for the structure of graph to be produced.

```
define statgraph ForestPlot ;
```

Define macro variables SI that will resolve to numbers for bubble size in the graph.

```
nmvar %do i=1 %to &nobs;
           s&i
      %end;;
```

Define the dimension of the graph.

```
begingraph / designwidth=1000px designheight=600px;
```

Define title.

```
entrytitle "Table 1" /  pad=(bottom=5px);
```

Define output layout: The following statement creates a grid of graph cells with 6 columns and each column width will be proportional to the values specified in columnweights. We can use external AXES for the graph with lattice layout.

```
layout lattice / columns=6   columngutter=0
            columnweights=(.29 .31 .10 .10 .10 .10);
```

Define text column in the graph. The y2axisopts is used to define second y-axis in the graph for Baseline Risk Factors, so the values can be left justified in display. You may have to adjust the offsetmin and offsetmax in order to make the summary report well aligned to the graph on the right as SAS will align Y2 to Y axis by default, but not on the offset. In this example I have used 0.09 for both offsets. CONSTANT is a dummy variable set equal to one and is used to overlay text on the same graph.

```
layout overlay / walldisplay=none border=false
                 y2axisopts=(reverse=true type=discrete display=(tickvalues)
                 offsetmin=0.09 offsetmax=0.09)
               xaxisopts=(display=none offsetmin=0 offsetmax=0);
             entry halign=left "Category Subgroup"  / textattrs=GraphLabelText
             location=outside valign=top;
             scatterplot y=&keyvar x=constant  / yaxis=y2 markerattrs=(size=0);
endlayout;
```

Define the column with graph.

```
layout overlay / walldisplay=(fill)
              yaxisopts=( display=none reverse=true  offsetmin=0.1    offsetmax=0.1
            linearopts=(integer=true))
            xaxisopts=(type=log offsetmin=0 offsetmax=0
                            label="Hazard Ratio"
                       logopts=(base=10 tickintervalstyle=logexpand
                       minorticks=true viewmin=0.02 viewmax=50));
       entry "Favors Treatment                Favors Control" / location=outside
      valign=top textattrs=GraphLabelText;
      seriesplot y=keyvargp x=value / group=keyvargp lineattrs=(pattern=solid);
      scatterplot x=hazardratio y=&keyvar / markerattrs=(color=black symbol=plus
      size=1pct);
```

AXES can be of different type, discrete, time, linear or log. From the example above, you can see I used linear for y axis, but use log for the x axis in order to have a balanced look in the graph.

.

The following statement defines where a vertical reference line is drawn in the graph.

```
            referenceline x=1  / lineattrs=(pattern=shortdash);
```

Define footnote.

```
      entryfootnote halign=left "CI = Confidence interval." /pad=(bottom=0px);
```

2. Define STYLE template

```
      proc template;
            define Style foreststyle;
                  parent = styles.analysis;
                  style GraphFonts from GraphFonts
                  "Fonts used in graph styles" /
                        'GraphTitleFont' = (", ",8pt)
                        'GraphLabelFont' = (", ",8pt)
                        'GraphValueFont' = (", ",7pt)
                        'GraphDataFont' = (", ",7pt)
                        'GraphFootnoteFont' = (", ",8pt);
            end;
      run;
```

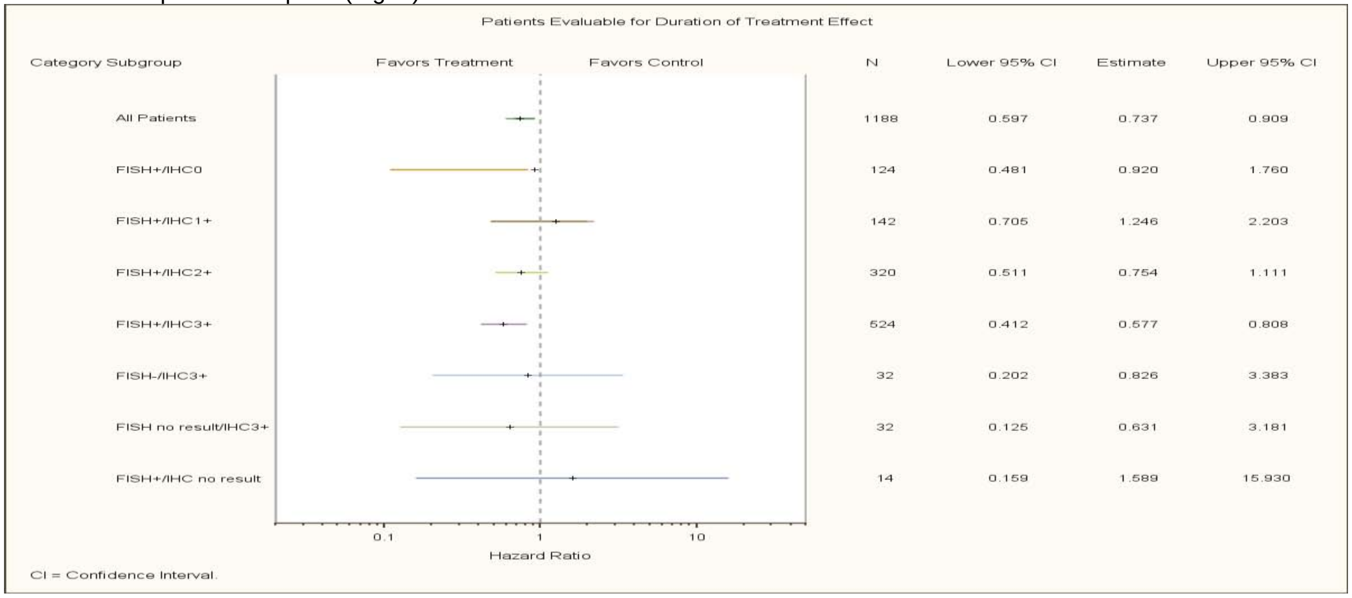
3. The SGRENDER procedure is bound to the ODS object at run time and directs the graphic output to the ODS destination.

```
      ods select none;
      ods listing close;
      ods graphics / width=10.5in height=8.5in imagename="&jobname" ;
      ods pdf style=foreststyle file="&jobname..pdf";

      ods select sgrender;
      proc sgrender data=fortable template=ForestPlot;
      run;

      ods graphics off;
      ods pdf close;
```

Here is the output of example 1 (Fig. 1).



Category Subgroup	N	Lower 95% CI	Estimate	Upper 95% CI
All Patients	1188	0.597	0.737	0.909
FISH+/IHC0	124	0.481	0.920	1.760
FISH+/IHC1+	142	0.705	1.246	2.203
FISH+/IHC2+	320	0.511	0.754	1.111
FISH+/IHC3+	524	0.412	0.577	0.808
FISH-/IHC3+	32	0.202	0.826	3.383
FISH no result/IHC3+	32	0.125	0.631	3.181
FISH+/IHC no result	14	0.159	1.589	15.930

Fig. 1

You can see the last three columns report the same information as the graph. Instead of reading the Odds Ratio and 95% CI from the text column, one can very easily spot the location of the odds ratio relative to the reference line 1 on the dynamic data visualization portion which enables you to quickly assess which sub-group showed treatment effect.

Sometimes, in addition to showing the odds ratio and 95% confidence interval, you may also want to display the treatment response/effect in difference groups. If the result is significant, you may want to emphasize the value of the odds ratio or sample size, then you can add more columns and change the symbol in the graph in order to create a comprehensive summary report.

```
layout lattice / columns=10   columngutter=0
                 columnweights=(.14 .04 .04 .05 .14  .04 .05 .14 .15 .21);

scatterplot y=keyvargp x=hazardratio / markerattrs=(color=black symbol=circle
size=sizen);
```

Before V9.3, SAS does not allow the use of variable values for graph attributes, so the scatterplot statement needs to be replaced with code below in order to create bubbles of varying size.

```
%do i=1 %to &nobs;
      scatterplot x=c&i y=v&i / markerattrs=(color=black symbol=circle size=s&i);
%end;;
```

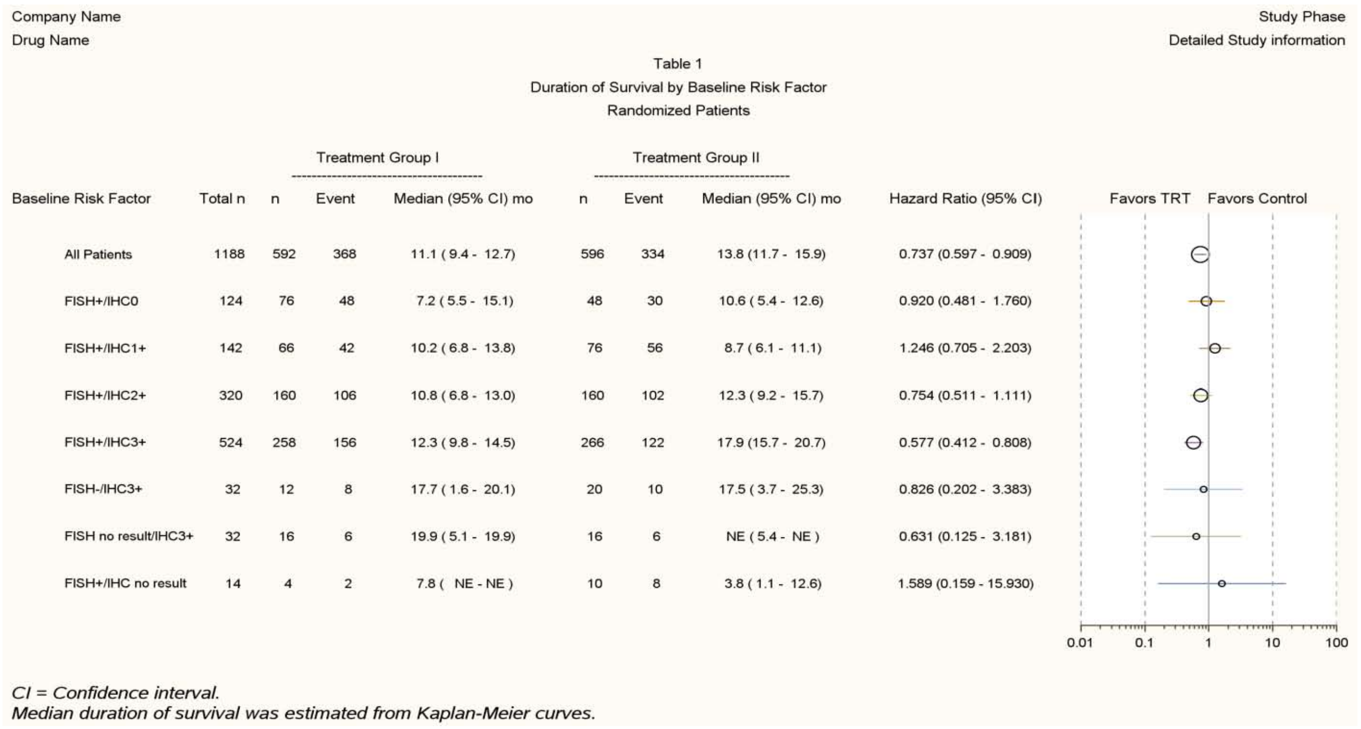

Company Name
Drug Name
Study Phase
Detailed Study information

Table 1
Duration of Survival by Baseline Risk Factor
Randomized Patients

		Treatment Group I			Treatment Group II			
Baseline Risk Factor	Total n	n	Event	Median (95% CI) mo	n	Event	Median (95% CI) mo	Hazard Ratio (95% CI)
All Patients	1188	592	368	11.1 (9.4 - 12.7)	596	334	13.8 (11.7 - 15.9)	0.737 (0.597 - 0.909)
FISH+/IHC0	124	76	48	7.2 (5.5 - 15.1)	48	30	10.6 (5.4 - 12.6)	0.920 (0.481 - 1.760)
FISH+/IHC1+	142	66	42	10.2 (6.8 - 13.8)	76	56	8.7 (6.1 - 11.1)	1.246 (0.705 - 2.203)
FISH+/IHC2+	320	160	106	10.8 (6.8 - 13.0)	160	102	12.3 (9.2 - 15.7)	0.754 (0.511 - 1.111)
FISH+/IHC3+	524	258	156	12.3 (9.8 - 14.5)	266	122	17.9 (15.7 - 20.7)	0.577 (0.412 - 0.808)
FISH-/IHC3+	32	12	8	17.7 (1.6 - 20.1)	20	10	17.5 (3.7 - 25.3)	0.826 (0.202 - 3.383)
FISH no result/IHC3+	32	16	6	19.9 (5.1 - 19.9)	16	6	NE (5.4 - NE)	0.631 (0.125 - 3.181)
FISH+/IHC no result	14	4	2	7.8 (NE - NE)	10	8	3.8 (1.1 - 12.6)	1.589 (0.159 - 15.930)

CI = Confidence interval.
Median duration of survival was estimated from Kaplan-Meier curves.

Fig. 2

The treatment benefit is often summarized in tabular and graphic formats. The example above shows how presenting both together can improve readability. You can see that one can very easily spot the location of the bubble relative to the reference line 1 on the dynamic data visualization portion which enables you to quickly assess which sub-groups showed treatment effect. You can have the death event or patient response, median survival time, etc. embedded in one display.

If, instead of having odds ratio as your statistical summary, you have MEAN, STD, Q1 and Q3 or those types of descriptive statistics, a box plot could be more appropriate. Therefore, by changing “seriesplot” to “boxplotparm” in the template (see code below,) you can generate a table with a box plot in it.

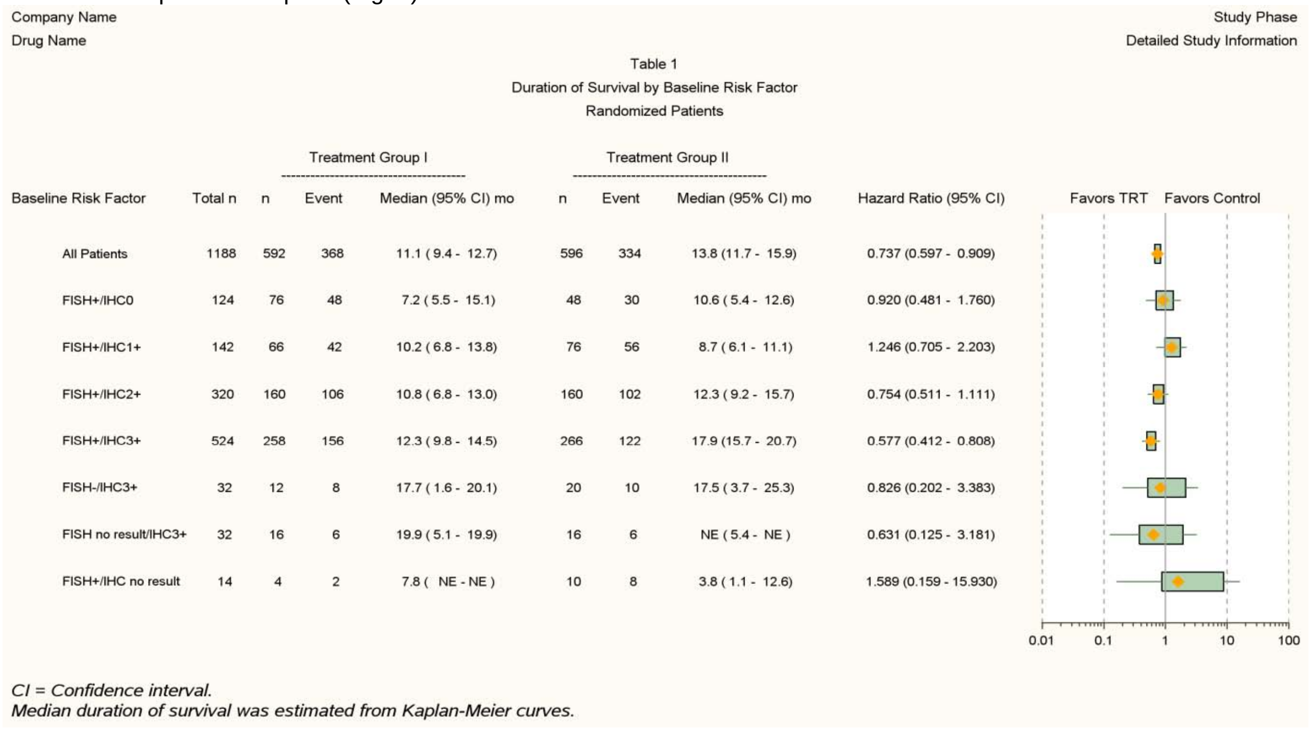
The boxplotparm statement below specifies that the box plot be drawn horizontally and stat=stat indicates which variable in the data set contains the statistics used for the graph.

```
        boxplotparm x=keyvargp y=value stat=stat / orient=horizontal display=(fill notches);
```

The scatter plot here is used for odds ratio to improve the readability.

```
        scatterplot x=hazardratio y=her2 / markerattrs=(color=orange symbol=diamondfilled
size=2pct);
```

Here is the output of example 3 (Fig. 3).



Company Name | Study Phase
Drug Name | Detailed Study Information

Table 1
Duration of Survival by Baseline Risk Factor
Randomized Patients

Baseline Risk Factor	Total n	Treatment Group I n	Treatment Group I Event	Treatment Group I Median (95% CI) mo	Treatment Group II n	Treatment Group II Event	Treatment Group II Median (95% CI) mo	Hazard Ratio (95% CI)
All Patients	1188	592	368	11.1 (9.4 - 12.7)	596	334	13.8 (11.7 - 15.9)	0.737 (0.597 - 0.909)
FISH+/IHC0	124	76	48	7.2 (5.5 - 15.1)	48	30	10.6 (5.4 - 12.6)	0.920 (0.481 - 1.760)
FISH+/IHC1+	142	66	42	10.2 (6.8 - 13.8)	76	56	8.7 (6.1 - 11.1)	1.246 (0.705 - 2.203)
FISH+/IHC2+	320	160	106	10.8 (6.8 - 13.0)	160	102	12.3 (9.2 - 15.7)	0.754 (0.511 - 1.111)
FISH+/IHC3+	524	258	156	12.3 (9.8 - 14.5)	266	122	17.9 (15.7 - 20.7)	0.577 (0.412 - 0.808)
FISH-/IHC3+	32	12	8	17.7 (1.6 - 20.1)	20	10	17.5 (3.7 - 25.3)	0.826 (0.202 - 3.383)
FISH no result/IHC3+	32	16	6	19.9 (5.1 - 19.9)	16	6	NE (5.4 - NE)	0.631 (0.125 - 3.181)
FISH+/IHC no result	14	4	2	7.8 (NE - NE)	10	8	3.8 (1.1 - 12.6)	1.589 (0.159 - 15.930)

CI = Confidence interval.
Median duration of survival was estimated from Kaplan-Meier curves.

Fig. 3

The extreme end points of the plot lines are the MIN and MAX while the two ends of the box indicate Q1 and Q3. From the graph above you can quickly visualize how your data is distributed in each individual category rather than interpreting the numbers in the corresponding text column. Note that the width of the boxes can be adjusted to be proportional to any variable you desire.

MAKING GRAPHICS LESS RESTRICTED TO INPUT DATA BY USING GTL EVAL FUNCTION

Not only can GTL be used to create graphics independent of the statistical procedure, it can also make graphic generation more data independent. The compiled GTL can be reused with different data input. For example, the yerrorlower and yerrorupper options when used in combination with the EVAL function allow us to easily draw a graph with dynamic standard error bars.

EXAMPLE 2. CREATE A MEAN CHANGE PLOT WITH STANDARD ERROR BARS IN IT.

For analysis based on survey data over time, we would typically like to know if the overall response is changed during the study and if it shows any treatment effect. The mean change from baseline is a very popular statistic to use.

```
        layout overlay / cycleattrs=true
                         yaxisopts=(griddisplay=on label='Mean Change from Baseline +- SE'
                         linearopts=(viewmin=-6.05))
```

Innovative uses of ODS and GTL, continued

```
                                   linearopts=(viewmin=-6.05 viewmax=15))
                                   xaxisopts=(linearopts=(tickvaluesequence=(start=1 end=20
                                   increment=1)));
                                   seriesplot  x=timrng y=mean / group=trtc name='Line1'
                                   lineattrs=graphData5;
                                   scatterplot x=timrng y=mean / group=trtc
                                             yerrorlower=eval(mean - 3*stdmean)
                                             yerrorupper=eval(mean + 3*stdmean) name='Line3';
                                                discretelegend 'Line1' 'Line3'/location=inside
                                      valign=bottom halign=center;
       endlayout;
```

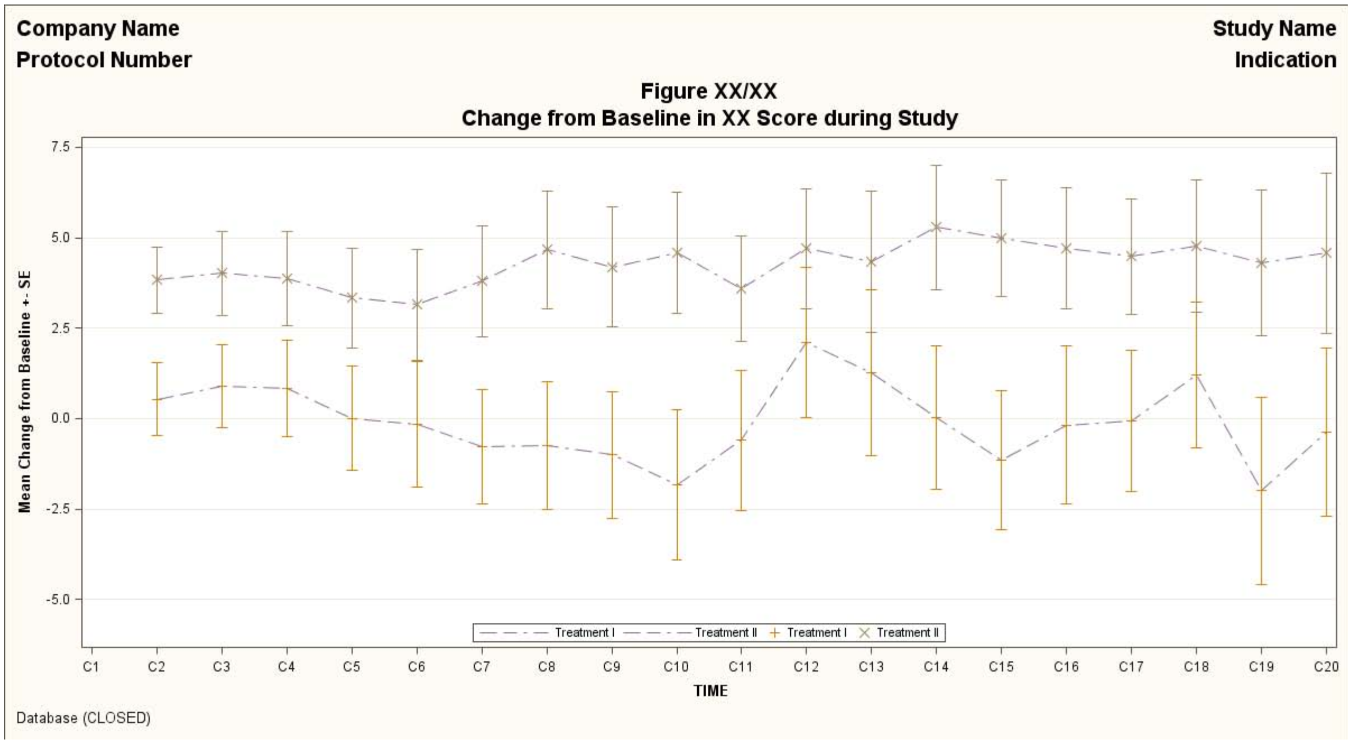


Fig. 4

CREATING STATISTICAL GRAPHS FROM SAS PROCEDURE WITH ODS GRAPHICS

In earlier versions of SAS® producing statistical graphs involved three steps, the survival curves for several risk groups with the LIFETEST procedure is a good example. First we needed to create output data sets with curve values and metadata such as p-values, and then we needed to modify the data sets for graphing and finally, we needed to produce the plot with the annotate facility. Today, with the ODS graphics option turned on, the statistical graphs are very easy to produce.

EXAMPLE 3. CREATE SURVIVAL PLOT FOR DURATION OF TREATMENT EFFECT WITH 95% CONFIDENCE INTERVAL IN IT.

Survival Curves can also be used for survey data analysis, so here is an example to show how customer satisfaction scores dropped one point over time between different treatment groups.

```
       ods listing close;

       ods graphics / width=15.5in height=8.5in imagename="&jobname";
       ods pdf file="../results/&jobname..pdf";

       ods graphics on;
```

Innovative uses of ODS and GTL, continued

```
proc lifetest data=survey1
     plots=(survival (atrisk=0 to 25 by 1 cl test) logsurv);
     time duration1*censor1(1);
     strata trt /group=trt test=(logrank);
run;
ods graphics off;

ods listing;
```

The PLOTS=SURVIVAL statement is one option within PROC LIFETEST which requests the plot of the estimated survival curves, and the ATRISK= modifier requests that the numbers of patients at risk at 0, 5, 10, . . ., 25 months are to be displayed. You can specify the CL modifier for confidence intervals for the survivor functions.
.
With stratified data, PROC LIFETEST computes a test of the homogeneity of the survival curves across strata. The TEST=LOGRANK modifier requests the p-value of the log-rank test which is a default test and this can be added as a plot inset as shown

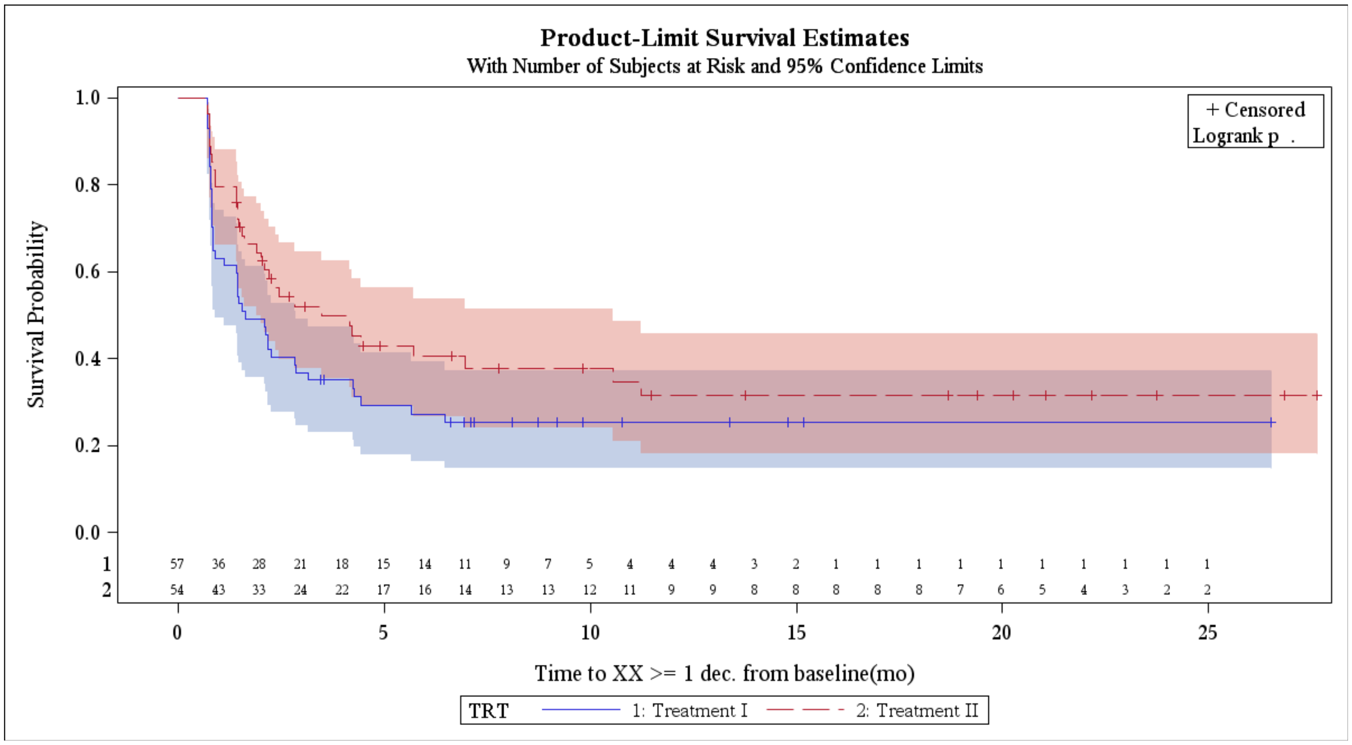


Fig. 5

At 10 months, the numbers of patients at risk are 5 and 12 for the two treatment groups respectively.

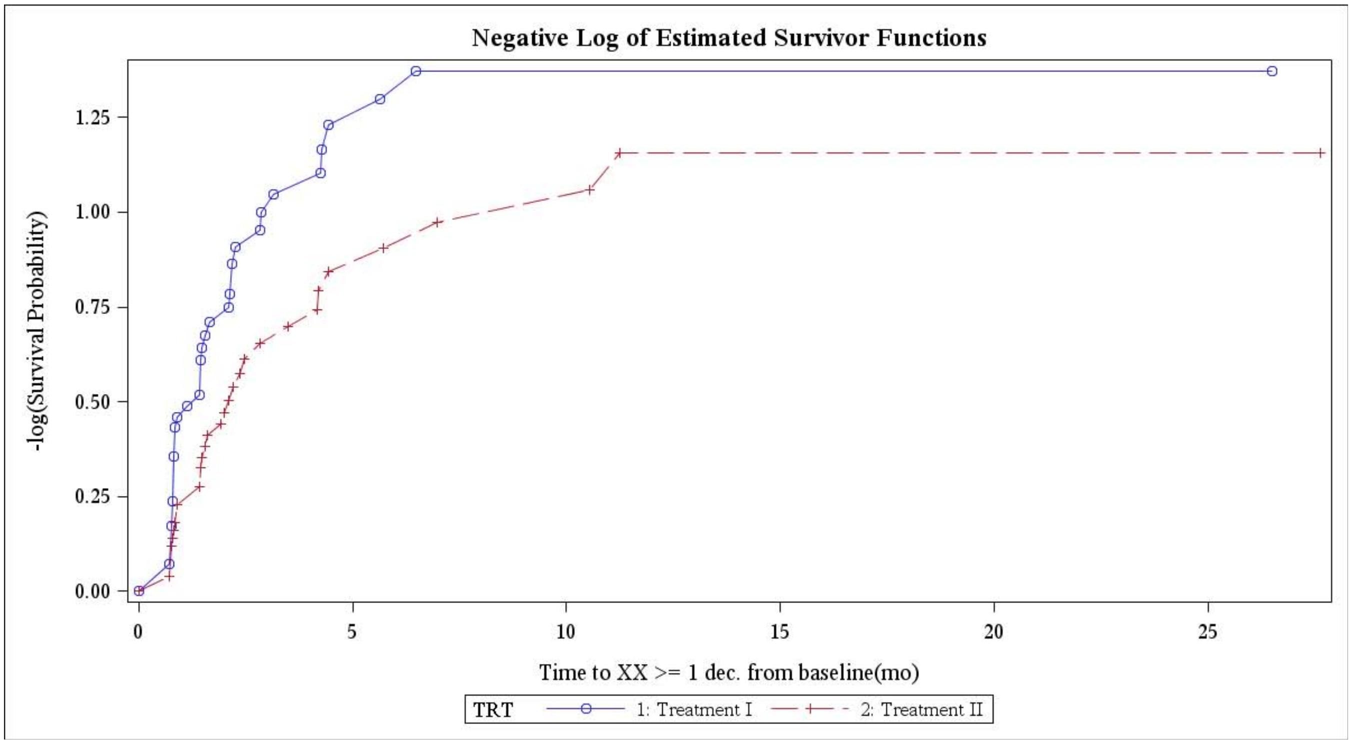


Fig. 6

This is the logsurv graph from the proc lifetest.

EDITING GRAPHS

While ODS Graphics are designed to automate the creation of sophisticated statistical graphics, one may still wish to customize their graphs or on occasion produce ad hoc graphs. To that end the ODS Graphics Editor provides an interactive (point-and-click) environment which enables one to make immediate changes to the graph that are data independent. One can also modify the ODS graph template for a plot to make changes which persist over time.

MODIFYING YOUR GRAPHS WITH GRAPH TEMPLATE

```
        ods listing sge=on;

        proc sgplot data=enso noautolegend tmplout="test1";
              title 'Atmospheric Pressure Differences Between '
                    'Easter Island and Darwin, Australia';
              pbspline y=pressure x=year;
        run;
```

By running the code above, the template code listed below will be generated for you. You can make any necessary changes to it as needed.

```
        proc template;
             define statgraph sgplot;
             begingraph;
             EntryTitle "Atmospheric Pressure Differences Between Easter Island and Darwin,
        Australia" /;
           layout overlay;
             ScatterPlot X=year Y=Pressure / primary=true;
             PBSplinePlot X=year Y=Pressure / NAME="PBSPLINE" LegendLabel="Penalized B-
        spline";
           endlayout;
           endgraph;
```

```
end;
run;
```

MODIFYING YOUR GRAPHS WITH THE ODS GRAPHICS EDITOR

The ODS Graphics Editor allows us to customize a graph. Text can be added in the form of titles, footnotes or labels, individual data points can be annotated, and graphic properties such as fonts, colors, and line styles can be modified. After you have customized your graph, you can save it as a Portable Network Graphic (PNG) file or as an SGE file. A new SGE=ON option added to the ODS LISTING statement can be used to create editable ODS graphs. When you run a graph procedure with this option, the ODS Graphics Editor generates an ODS Graphics Editor file (with the extension .sge). SGE files can be opened from the Results window or from an installed stand-alone ODS Graphics Editor (Windows and Linux hosts only). If you later change and rerun the SAS program, SAS creates a new SGE file. The original SGE file remains in the SAS Results window.

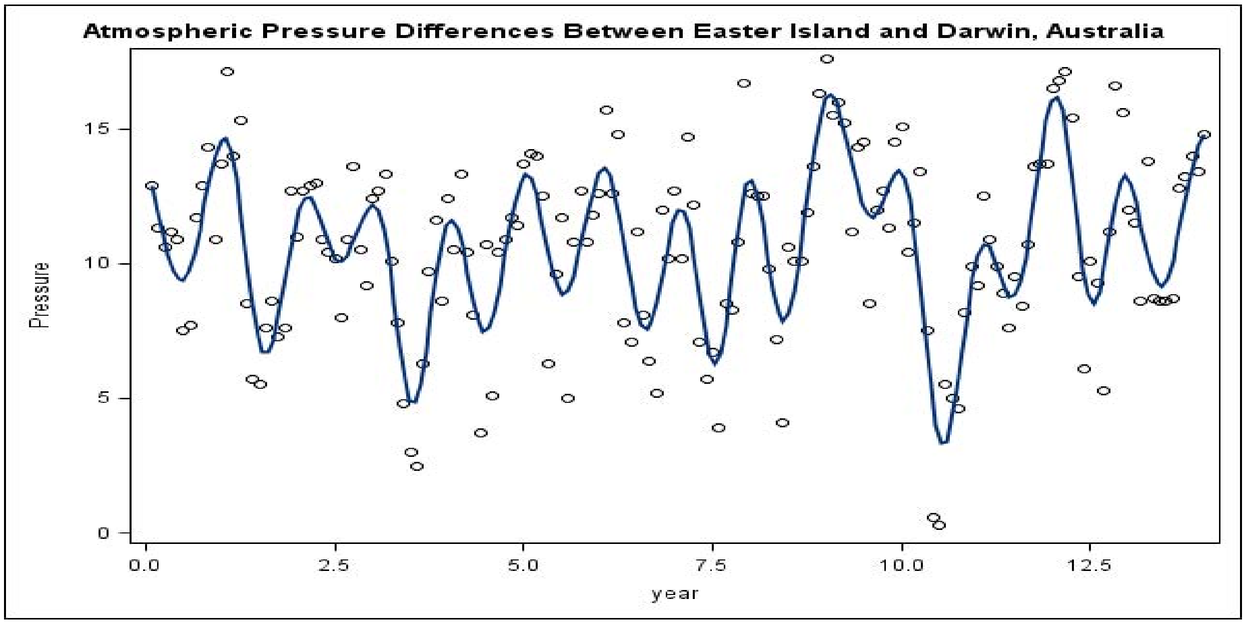


Fig. 7

This is the graph created by the sgplot procedure above. For publication purposes, we often need to add some annotation to the graph so you can easily see marked outliers. We can also add annotation, add footnote, add a pointer or even change the data content in the graph. The X-axis was previously labeled as year Fig. 7, but we can change it to Year Fig. 8.

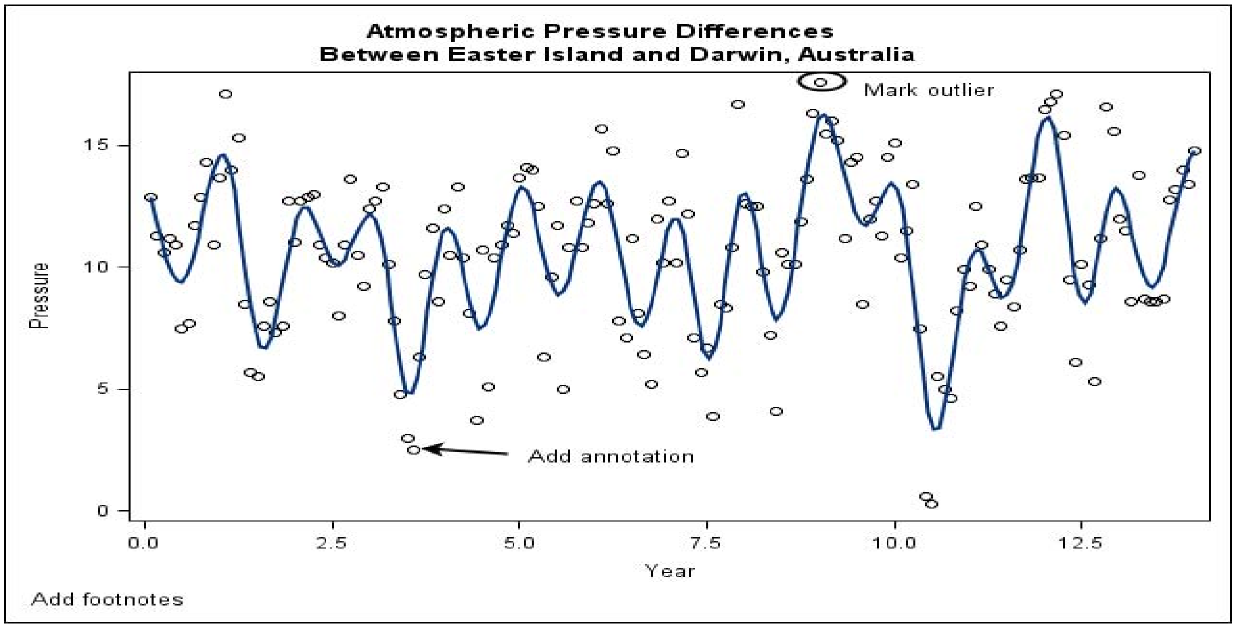


Fig. 8

ADVANTAGES

• You don't need to know anything about GTL details to create statistical graphs.
• It is very easy to understand, therefore can be easily modified to reproduce different graphs.
• The quality of graphics is good.
• Save time on creating the same output over the old version of SAS with annotated dataset.
• Platform independent.

CONCLUSIONS

With the new features that ODS Graphs and GTL have to offer, a high quality summary/graphics output can be created on the fly. It not only cuts down the program developing time significantly, it helps with the data interpretation efficiency. In the end, we can make our job more efficient by creating fewer outputs with higher quality.

REFERENCES

- Forest Plot from SAS®. http://support.sas.com/kb/35/143.html
- SAS documentation SAS® 9.2 and 9.3
- GTL (Graphics Template Language) in SAS® 9.2. http://www.phuse.eu/download.aspx?type=cms&docID=540
- A Programmer's Introduction to the Graph Template Language. http://www2.sas.com/proceedings/sugi31/262-31.pdf
- Gaining Power from GTL and ODS Style to Control Graphical Output. http://www2.sas.com/proceedings/forum2007/092-2007.pdf
- A Programmer's Introduction to the Graphics Template Language. http://www2.sas.com/proceedings/sugi31/262-31.pdf
- ODS Statistical Graphics for Clinical Research. http://www2.sas.com/proceedings/sugi31/095-31.pdf
- An Over Overview of ODS Statistical Graphics in SAS® 9.3 http://support.sas.com/resources/papers/76822_ODSGraph2011.pdf
- Creating Statistical Graphics in SAS® 9.2: What Every Statistical User Should Know**.** http://www2.sas.com/proceedings/sugi31/192-31.pc
- Getting Started with ODS Statistical Graphics in SAS® 9.2—Revised 2009. http://support.sas.com/rnd/app/papers/ods/intodsgraph.pdf

ACKNOWLDEFMENTS

Sanjay Matange from SAS Institute.
Vivian Ng, Biostat from Genentech. A Member of the Roche Group
Song Liu, SPA from Genentech. A Member of the Roche Group

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Qinghua (Kathy) Chen
Genentech Inc
1 DNA way,
South San Francisco, CA, 94080
(o): 650-225-8427
Email: chen.kathy@gene.com

APPENDIX

```
*** Example #1 ***;

proc template;
  define statgraph ForestPlot ;
  begingraph / designwidth=1000px designheight=600px;
  entrytitle halign=left "Analysis Population: Patients Evaluable for Duration of Treatment
Effect" / pad=(bottom=5px);
  entrytitle " " / pad=(bottom=5px);
  layout lattice / columns=6   columngutter=0
                   columnweights=(.29 .31 .10 .10 .10 .10);
  layout overlay / walldisplay=none border=false
                   y2axisopts=(reverse=true type=discrete display=(tickvalues) offsetmin=0.09
offsetmax=0.09)
                   xaxisopts=(display=none offsetmin=0 offsetmax=0);
    entry halign=left "Category Subgroup"  / textattrs=GraphLabelText location=outside
valign=top;
    scatterplot y=&keyvar x=constant  / yaxis=y2 markerattrs=(size=0);
  endlayout;

  layout overlay / walldisplay=(fill)
                   yaxisopts=( display=none reverse=true  offsetmin=0.1 offsetmax=0.1
                   linearopts=(integer=true))
                   xaxisopts=(type=log offsetmin=0 offsetmax=0
                                   label="Hazard Ratio"
                              logopts=(base=10 tickintervalstyle=logexpand
                              minorticks=true viewmin=0.02 viewmax=20));
    entry "Favors Treatment                 Favors Control" / location=outside valign=top
textattrs=GraphLabelText;
    seriesplot y=keyvargp x=value / group=keyvargp lineattrs=(pattern=solid);
    scatterplot x=hazardratio y=&keyvar / markerattrs=(color=black symbol=plus size=1pct);

    referenceline x=1  / lineattrs=(pattern=shortdash);
    referenceline x=.02 / lineattrs=(pattern=solid);
    referenceline x=20 / lineattrs=(pattern=solid);
  endlayout;

  layout overlay  / walldisplay=none border=false
                    yaxisopts=(reverse=true type=discrete display=none)
                    xaxisopts=(display=none offsetmin=0 offsetmax=0);
    entry "N" / location=outside valign=top textattrs=GraphLabelText;
    scatterplot y=&keyvar x=constant  /   markercharacter=n
                                     markercharacterattrs=GraphDataText;
  endlayout;
  layout overlay  / walldisplay=none border=false
                    yaxisopts=(reverse=true type=discrete display=none)
                    xaxisopts=(display=none offsetmin=0 offsetmax=0);
    entry "Lower 95% CI" / location=outside valign=top textattrs=GraphLabelText;
    scatterplot y=&keyvar x=constant  /  markercharacter=hrlowercl
                                    markercharacterattrs=GraphDataText;
  endlayout;
  layout overlay  / walldisplay=none border=false
                    yaxisopts=(reverse=true type=discrete display=none)
                    xaxisopts=(display=none offsetmin=0 offsetmax=0);
    entry "Estimate" / location=outside valign=top textattrs=GraphLabelText;
    scatterplot y=&keyvar x=constant  /  markercharacter=hazardratio
                                      markercharacterattrs=GraphDataText;
  endlayout;
  layout overlay  / walldisplay=none border=false
                    yaxisopts=(reverse=true type=discrete display=none)
                    xaxisopts=(display=none offsetmin=0 offsetmax=0);
    entry "Upper 95% CI" / location=outside valign=top textattrs=GraphLabelText;
    scatterplot y=&keyvar x=constant  /  markercharacter=hruppercl
                                      markercharacterattrs=GraphDataText;
  endlayout;
  entryfootnote halign=left "CI = Confidence Interval." /  pad=(bottom=2px);
  endlayout;
  endgraph;
  end;
run;


*** Example #2 ***;

proc template;
  define statgraph meanstd;
  begingraph / designwidth=1100 designheight=800;
  entrytitle halign=left "Company Name" halign=right "Study Name" /  pad=(bottom=5px);
  entrytitle halign=left "Protocol Number" halign=right "Indication" /  pad=(bottom=5px);

  entrytitle 'Figure XX/XX;
  entrytitle 'Change from Baseline in XX Scores during Treatment';
  entrytitle 'Randomized Female Patients with Baseline and at Leaset One Post-Baseline Valid
Score';

    layout overlay / cycleattrs=true
                     yaxisopts=(griddisplay=on label='Mean Change from Baseline +- SE'
linearopts=(viewmin=-6.05))
                     y2axisopts=(label='Mean Change in Total Score' linearopts=(viewmin=-6.05
viewmax=15))
                     xaxisopts=(linearopts=(tickvaluesequence=(start=1 end=25 increment=1)));
      seriesplot  x=timrng y=mean / group=trtc name='Line1' lineattrs=graphData5;
      scatterplot x=timrng y=mean / group=trtc
                                    yerrorlower=eval(mean - 3*stdmean)
                                    yerrorupper=eval(mean + 3*stdmean) name='Line3';
      discretelegend 'Line1' 'Line3'/location=inside valign=bottom halign=center;
    endlayout;
      entryfootnote halign=left "&FN_9L" /  pad=(bottom=5px);
  endgraph;
  end;
run;
```