PharmaSUG 2017 - Paper DV13

The basics of Graph Template Language and output with ODS DOCUMENT

Fangping Chen, United BioSource Corporation-An Express Scripts Company, Ann Arbor, MI, USA

ABSTRACT

ODS GRAPH TEMPLATE LANGUAGE (GTL) is a powerful language in SAS® 9.2. PROC TEMPLATE procedure in GTL can create customized graphs independently from DATA step. Rendered sophisticated figures to ODS DOCUMENT destination, user can rearrange the individual components of the output and modify the output report at the very last step without touching DATA steps. This paper will present the basic PROC TEMPLATE procedure and ODS DOCUMENT destination with PROC DOCUMENT procedure step by step for those who touch those procedures for the first time.

INTRODUCTION

PROC TEMPLATE procedure with statement DEFINE STATGRAPH is a powerful language to define sophisticated graphs. User, who wants to edit titles and footnotes, reposition the legends, and customize the visual properties of the plots and axes, can use the procedure to personalize the visualization of graphs. With the ODS DOCUMENT, which can store and rearrange any component in the document, user can customize the graph output with more flexibility.

BASIC STRUCTURE OF A STATGRAPH TEMPLATE

SIGLE CELL LAYOUT WITH SINGLE PLOT

This is a basic structure of a STATGRAPH template to define the structure and appearance of a figure. The DEFINE statement with NAME option in PROC TEMPLATE procedure create a name for the template. In this paper, it gives a name to the template as TSTFIG. BEGINGRAPH block is used to design the outermost container of a graph, like the title, footnote and the size of the figure. All STATGRAPH templates must start with a BEGINGRAPH statement and end with an ENDGRAPH statement . SAS provide some options, such as DESIGNWIDTH/ DESIGNHEIGH, to specify the size of your figure outermost container. ENTRYTITEL and ENTRYFOOTNOTE statement is used to implant titles and footnotes for your figure. Options are available to specify the appearance of title and footnote. HALIGN= CENTER |LEFT |RIGHT specify the horizontal alignment of the text. By default, HALIGN is center aligned. TEXTATTRS options control the font and color of the text when used with suboptions COLOR and SIZE. Layout block is the vital part of STATGRAPH template procedure, which builds the inner container mainly occupied by plots. Layout blocks always begin with the LAYOUT keyword and close with the ENDLAYOUT statement. LAYOUT statement construct the layout in a customized manner by arranging plots, legend, labels or other components into a single cell, or rows or columns of cells. There are four types of layout: single-cell, multi-cell, data-driven and legend (table 1). In this paper, single-cell layout and multi-cell layouts will be introduced. OVERLAY is most frequently used option in single-cell layout, which can be used to create a 2-D single cell plot and where we can personalize more on X and Y axes. 2D plots have four independent axes (X, X2, Y, Y2). By default, the X2 and Y2 are same as X and Y and not displayed. Options YAXISOPTS, Y2AXISOPTS, XANISOPTS, and X2AXISOPTS can be used to modify all four axes. Mostly, we only use X and Y axes. Axes also have four different types, LINEAR, DISCRETE, TIME and LOG. We selected the type depending on the nature of the data. TYPE option can be used to override the default type. DISCRETEOPTS suboption specifies the discrete axis, LINEAROPTS suboption specifies the linear axis, LOGOPTS specifies the log axis, and TIMEOPTS specifies the time axis. DISPLAY option controls four features displaying on axis, they are line, ticks, tickvalues and label. For example, DISPLAY= line means only line will display on X or Y axis, and ticks, tickvalues, labels will be suppressed. By default, all of them will present if no DISPLAY option indicate. LABEL and LABELATTRS options are powerful option to overwrite the default label from the name of variables by specifying the label. In the case below, the Y axis had been label as ‘ALT(U/L)’ instead of ‘ALT’ which is the name of Y variable. NAME option allows assigning a name to the axis for reference on other statements. TICKSTYLE and TICKVALUEATTRS are options to locate the tick marker in axis and define the color and font of tick values. For linear type of axis only, TICKVALUESEQUENCE controls the start, the end and the increment of tick values. To including desired data into your plot, VIEWMAX and VIEWMIN can be used. After the design of inner container, specified plot statement could be introduced to create desired plot. SCATTERPLOT, HISTOGRAM, DENSITYPLOT, BOXPLOT, and SERIESPLOT are some most used plots. GTL plot statements provide options to customize plots appearance too. For instance of SEREISPLOT, X and Y options are required to define X variable and Y variable. Line and marker are objects mostly likely to be personalized in line plot. DISPLAY option in SERIESPLOT has three options. They are STANDARD, ALL and MARKER. By default, STANDARD is assigned and means line only without markers. Both suboptions ALL and MARKER mean line will company with markers. In this case, DISPLAY=(MARKER) means that both line and markers will present. Furthermore, we can customize the line and markers to meet your requirement. LINEATTRS is used to specify the appearance of line. The suboption PATTERN is available to override the default line pattern, solid line. Figure 1 shows the most often used line patterns. SYMBOL suboption in MARKERATTRS is a powerful one to specify marker symbols. Figure 2 is the most often used marker symbols. COLOR suboption specifies the color of marker symbol. As same as NAME option in axes, NAME option with SERIESPLOT will give a name to the plot for future reference. Please see the example below and Figure 3 is the output from ALT data of patient USA/005-01 in the style of TSTFIG template.

```
 proc template;
      define statgraph tstfig;
        begingraph / designwidth=640px designheight=320px;
                 entrytitle "LDL Cholesterol, STUDY DRUG and PCSK9 Dosing, and LFT Profiles by Patient";
                 entryfootnote "Patient = USA/005-01";
           layout overlay /
                 yaxisopts=(type=linear
                   linearopts=(viewmin=0 viewmax=300
                      tickvaluesequence=(start=0 end=300 increment=100)
                   )
                   label="ALT (U/L)"
                   labelattrs=(size=8)
                 )
                 xaxisopts=( type=linear
                   linearopts=(viewmin=0 viewmax=36
                   tickvaluesequence=(start=0 end=36 increment=3)
                    )
                   label ="Month"
                   labelattrs=(size=8)
                 );

                seriesplot x=plotmonth y=alt/
                 display=(markers)
                 lineattrs=(color=black pattern=dot)
                 markerattrs=(color=black  symbol=asterisk)
                 name="adlb_ALT";
           endlayout;
         endgraph;
      end;
  run;
```

Signal-cell Layouts	Multi-cell Layouts

OVERLAY	2-D (1 cell)	LATTICE	Axes can be shared across columns or rows and be external to grid. Many grid labeling and alignment features.
VERLAYEQUATED	Specialized OVERLAY with equated axes. X and Y axes are always numeric (TYPE=LINEAR). The slope of a line in the display is the same as the slope in the data since the display distance is the same in both axes.	GRIDDED	Axes independent for each cell.
PROTOTYPE	Specialized LAYOUT used only as child layout of DATAPANEL or DATALATTICE	DATAPANEL	Displays a panel of similar graphs based on data subsets by classification variable(s). Number of cells is based on crossings of n classification variable(s).
REGION	General purpose layout for displaying a single-cell plot that does not use axes	DATALATTICE	Displays a panel of similar graphs based on data subsets by classification variable(s). Number of cells is based on crossings of 1 or 2 classification variables
OVERLAY3D	General purpose 3-D layout for superimposing 3-D plots.		

Table 1, single-cell and multi-cells layout.

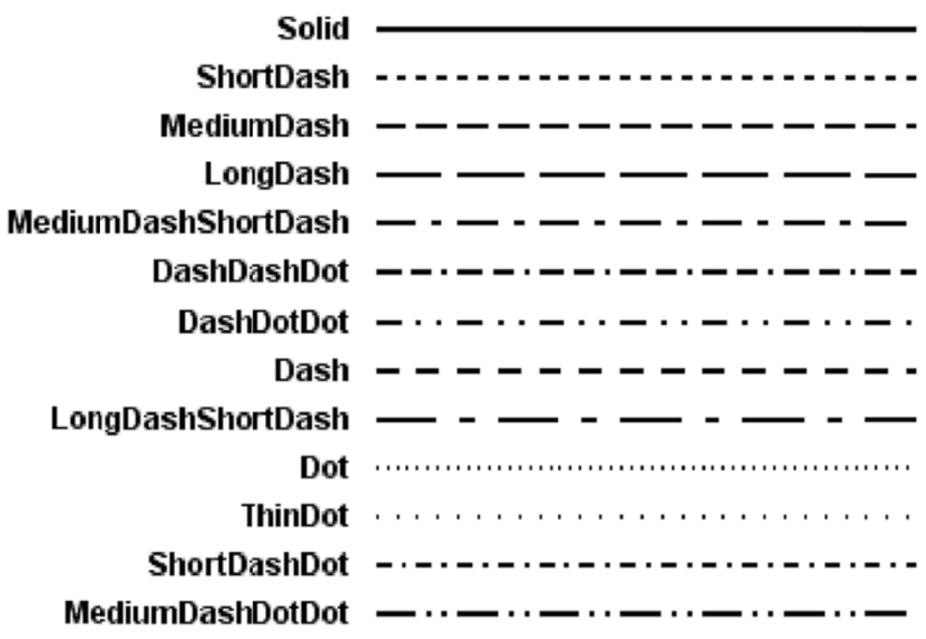


Figure 1, SAS name of line options.

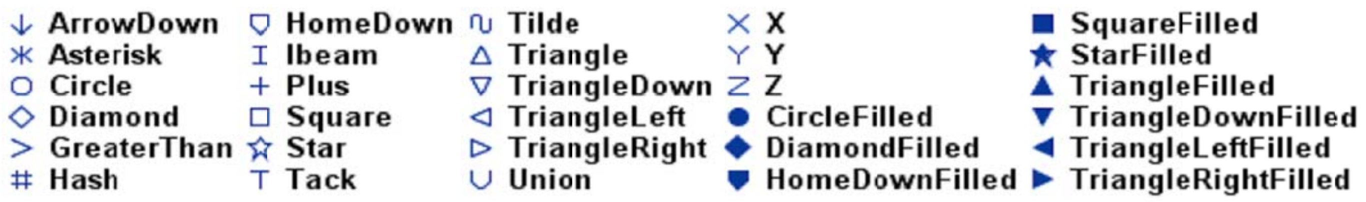


Figure 2, SAS name of markers.

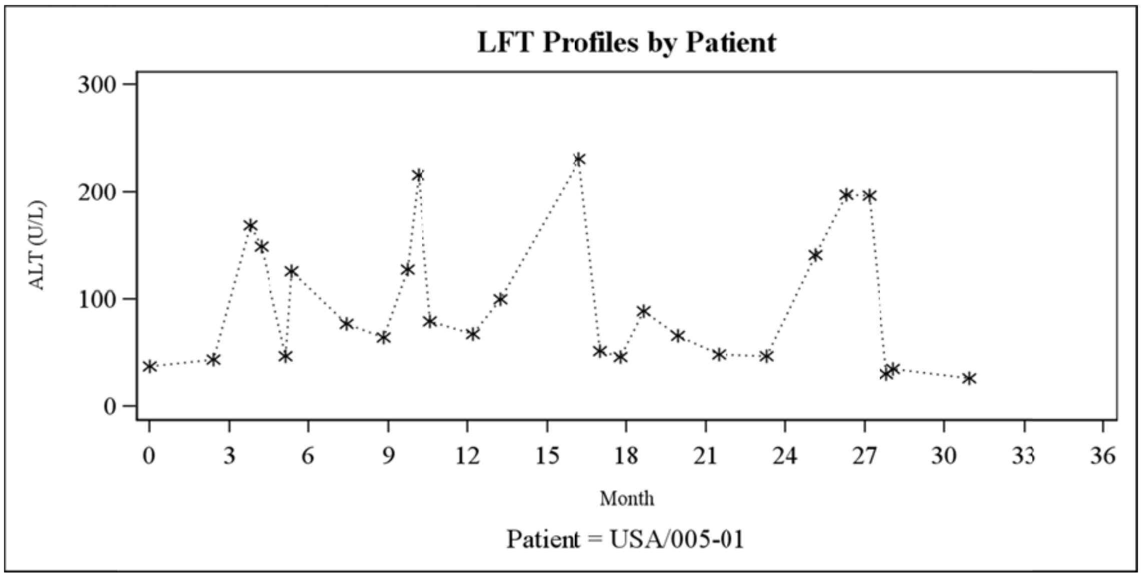


Figure 3. The single cell figure with one plot.

MULTI-PLOTS IN A CELL LAYOUT

Sometimes, more than one plot will be created in one cell. To achieve it, several PLOT statements can be introduced into the layout block. Here, to display liver function test (LFT) profile for the patient, both ALT and AST data are required to display in a single cell. Two implantations of SERIESPLOT created both plots in the cell, and NAME and LEGENDLABEL options were used to identify each. DISCRELEGEND statement was used to add legend in the cell to identify each plot with the help of NAME option. LEGENDLABEL option gives legend a name for each plot. Sometimes, some plots include different groups. Legends were used to identify the groups as nested layout by DISCRELEGEND statement. The plot's GROUP option automatically assigns distinct visual properties to each group value. In this case, DISCRETELEGEND statement has been used to generate the legend for each plot since the data from different variables instead of the groups of one variable. There are some very useful options in DISCRETELEGEND statement to personalize the appearance of legend, like HALIGN, VALIGN, and LOCATION options are working together to relocate the legend ideally. ACROSS and ROWN help to specify the number of legend entries in each row and column. Here, a legend has been used for plots 'ADLB_ALT' and 'ADLB_AST', with one entry per row only, surrounded by border, and the legend is located inside and right top of the cell. Please see the code and figure (Figure 4) below.

```
proc template;
      define statgraph TSTFIG1;
        begingraph / designwidth=640px designheight=320px;
                     entrytitle "LFT Profiles by Patient";
                     entryfootnote "Patient = USA/005-01";
               layout overlay /
                     yaxisopts=(
                       linearopts=(viewmin=0 viewmax=400
                          tickvaluesequence=(start=0 end=400 increment=100)
                       )
                       label="ALT (U/L)"
                       labelattrs=(size=8)
                     )
                     xaxisopts=(
                       linearopts=(viewmin=0 viewmax=36
                       tickvaluesequence=(start=0 end=36 increment=3)
                        )
                       label ="Month"
                       labelattrs=(size=8)
                     );
                    seriesplot x=plotmonth y=alt/
                     display=(markers)
                     connectorder=xaxis
                     lineattrs=(color=black pattern=dot)
                     markerattrs=(color=black  symbol=asterisk)
                     name="adlb_ALT"
                     legendlabel = 'ALT';
                   seriesplot x=plotmonth y=ast/
                     display=(markers)
                     connectorder=xaxis
```

```
                  lineattrs=(color=black pattern=solid)
                  markerattrs=(color=black  symbol=asterisk)
                  name="adlb_AST"
                  legendlabel = 'AST';
                discretelegend 'adlb_ALT' 'adlb_AST' /across =1 border=true halign=right
                 valign=top location=inside;
              endlayout;
           endgraph;
        end;
     run;
```

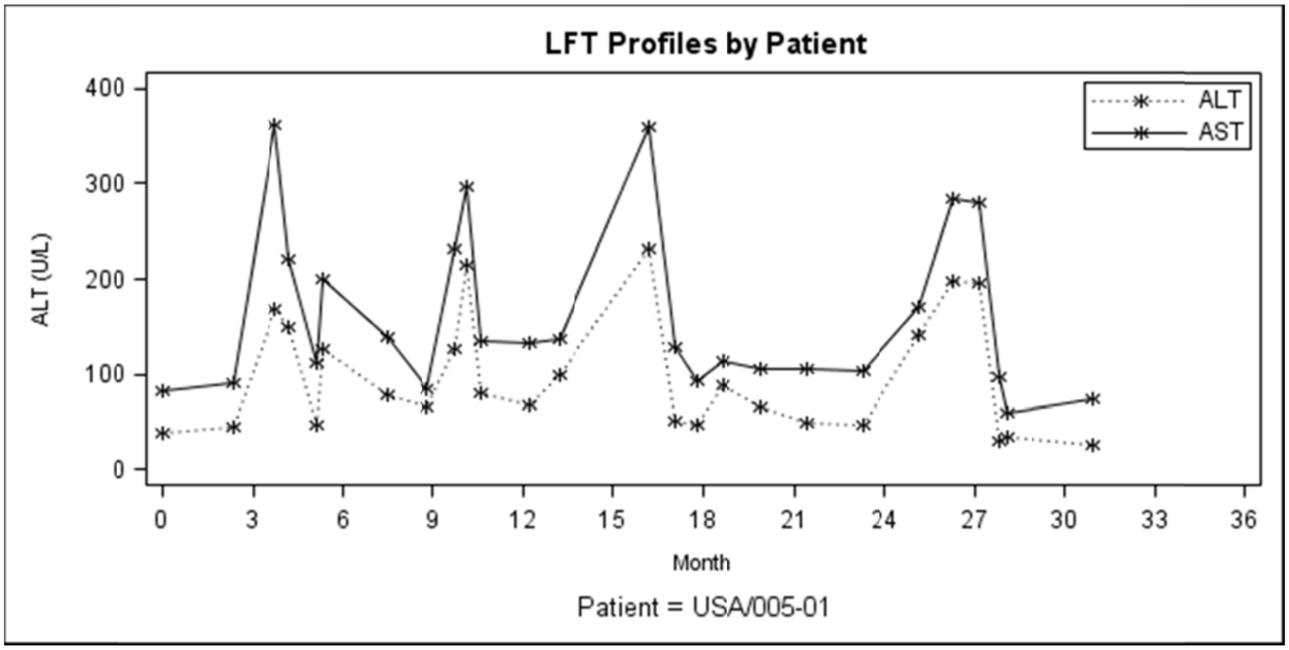


Figure 4. The single cell figure with multi-plot.

MULTI-CELL LAYOUT

Multi-cell layout means more than one cell in one row or one column. LAYOUT keyword with LATTICE or GRIDDED options are designed to create multi-cell layout. The difference is that LATTICE can share axes but GRIDDED creates independent axes. To achieve the alignment, LAYOUT LATTICE automatically assign plot areas and tick display areas in the plot and overrides axis-offset setting in LAYOUT OVERLAY statement in later steps. If you don't want the data range to be unified automatically, you can use LAYOUT GRIDDED to create the multi-cell layout. In this paper, LAYOUT LATTICE has been used to generated two cells which are aligned by XAXIS –PLOTMONTH. COLUMNDATARANG= UNION asked to unify the plot area and x-axes area by x-variable. COLUMNS=option design the how many column will display in the layout. ROWGUTTER option controlled the space between cells. ROWWEIGTS= option designed the fractional proportion of each cell relative to the overall grid height, not including headers, sidebars and row axes. The sum of reweights should be 1. Code and figure (Figure 5) are showed as below:

```
  proc template;
     define statgraph TSTFIG2;
        begingraph /
          designwidth=6.25in designheight=7.35in;
          layout lattice / columns=1 rowgutter = 5  rowweights=(.6 .4) columndatarang=union;
```

```
layout overlay /
    yaxisopts=(
      linearopts=(viewmin=0 viewmax=500
        tickvaluesequence=(start=0 end=500 increment=100)
      )
      label="LDL (mg/dL)"
      labelattrs=(size=8)
      )
    xaxisopts=(
      linearopts=(viewmin=0 viewmax=36)
      display=(line)
    );
  seriesplot x=plotmonth y=ldl /
    display=(markers)
    connectorder=xaxis
    lineattrs=(color=black pattern=solid)
    markerattrs=(color=black  symbol=trianglefilled)
    name="adlb_LDL";
endlayout;

layout overlay /
    yaxisopts=(
      linearopts=(viewmin=0 viewmax=400
        tickvaluesequence=(start=0 end=400 increment=100)
      )
      label='ALT and AST (U/L)'
      labelattrs=(size=8)
    )
    xaxisopts=(label='Months'
      linearopts=(viewmin=0 viewmax=36
        tickvaluesequence=(start=0 end =36 increment = 3)
    )
      tickvalueattrs=(size=8)
    );
  seriesplot x=plotmonth y=alt /
    display = (markers)
    connectorder=xaxis
    lineattrs=(color=black pattern=dot)
    markerattrs = (color=black symbol=asterisk)
    legendlabel='ALT '
```

```
                name="adlb_alt";
              seriesplot x=plotmonth y=ast /
                display = (markers)
                connectorder=xaxis
                lineattrs=(color=black pattern=solid)
                markerattrs=(color=black  symbol=asterisk)
                legendlabel='AST '
                name = 'adlb_ast';
              discretelegend 'adlb_alt' 'adlb_ast'  /
                border=true location=inside halign=right valign=top
                valueattrs=(size=8) autoitemsize=true across=1;
            endlayout;
          endlayout;
        endgraph;
      end;
   run;
```

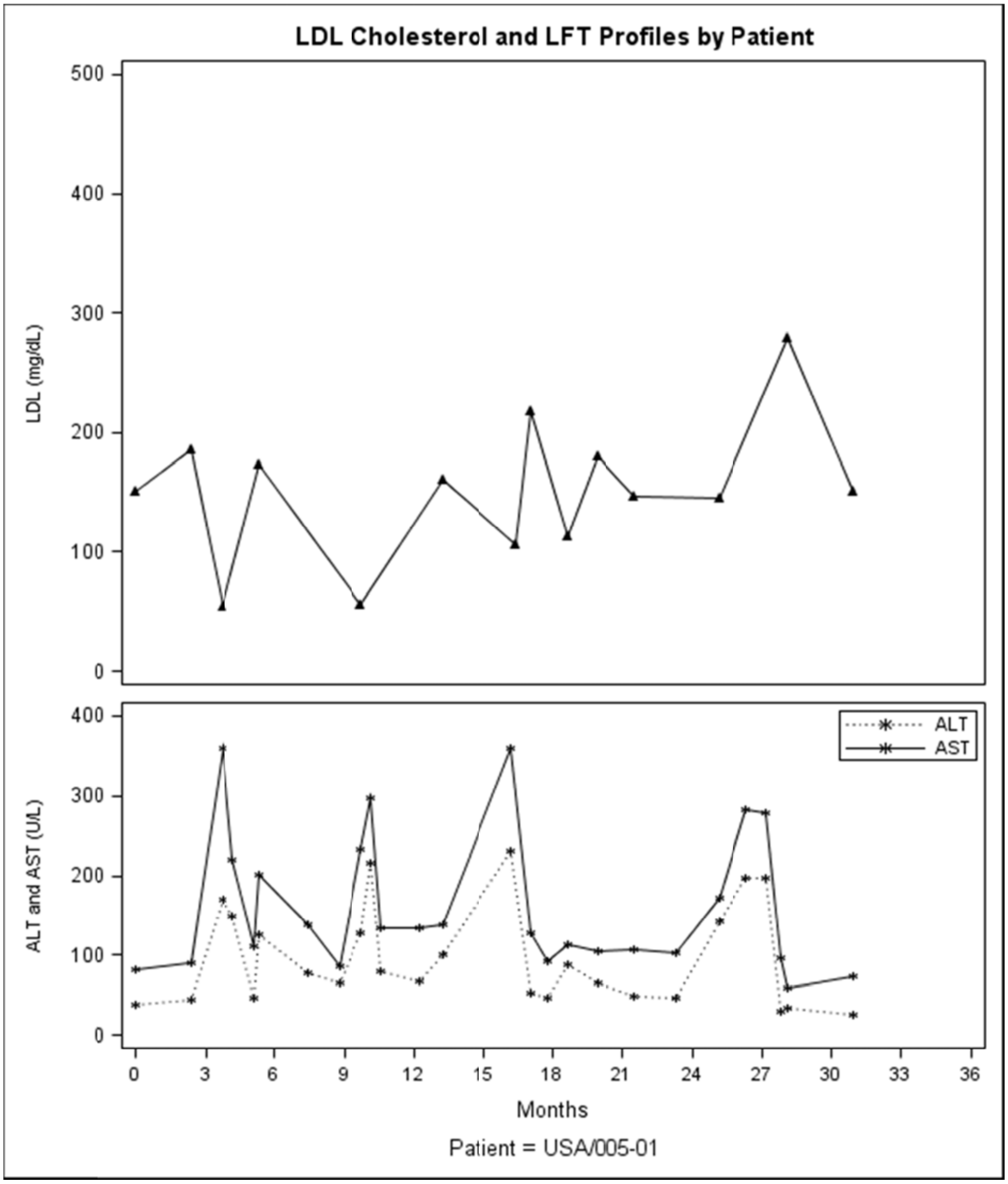


Figure 5. The multi-cell layout.

ODS DOCUMENT

Why I introduce ODS DOCUMENT even since the output can be done with ODS PDF, RTF and HTML? The reason is ODS DOCUMENT is more flexible for the last minute adjustment.
The ODS DOCUMENT destination doesn't create visual output like PDF, RTF or HTML. With the help of PROC DOCUMENT, you can not only change the styles and options when you render the output, but also store and rearrange the individual components of a report. ODS DOCUMENT opens the destination with ODS statement and close it with an ODS CLOSE statement as other ODSs'.

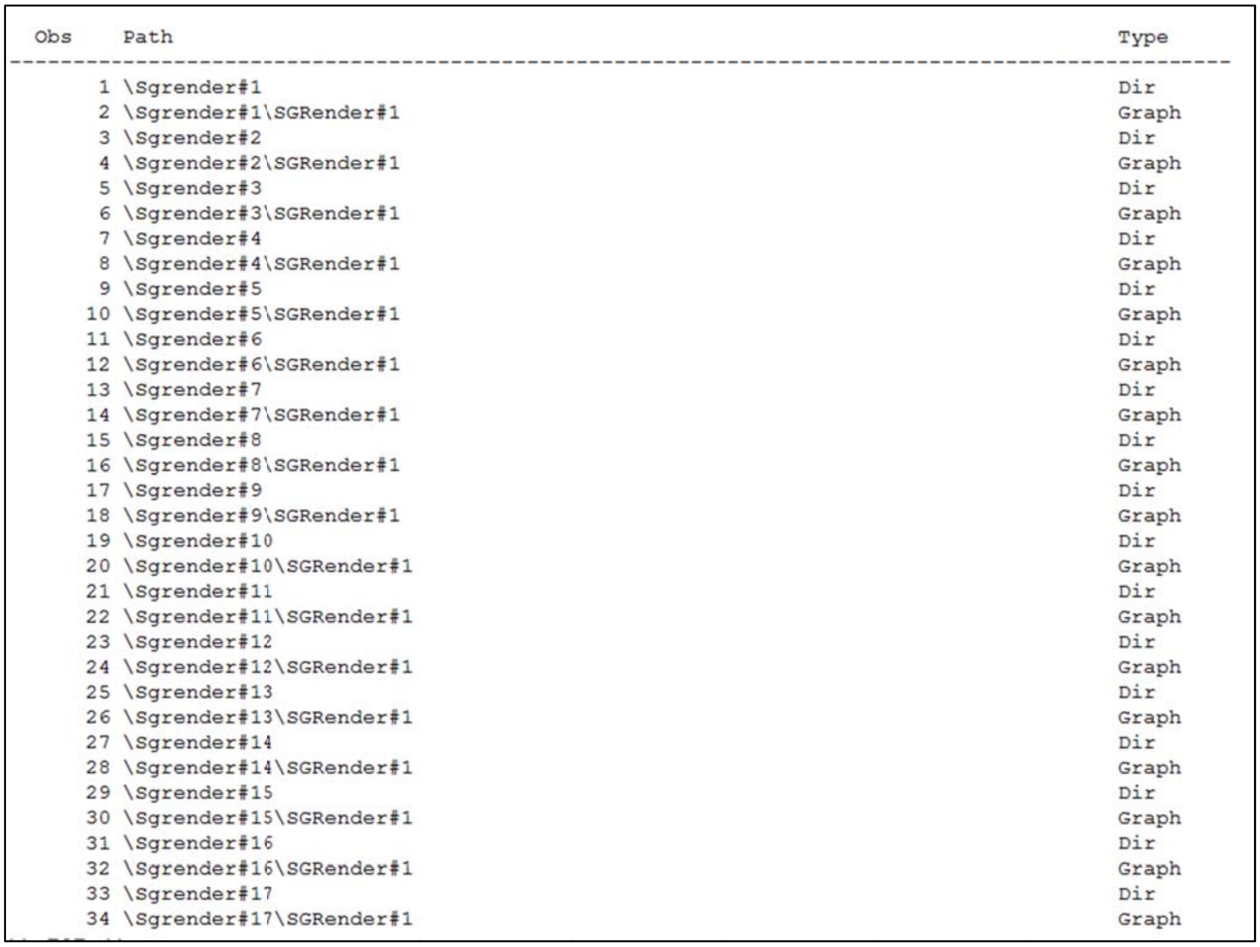
ODS DOCUMENT statement gives a name to ODS document by option NAME= in a binary format, such as WORK.FIGURE_ALL. By default, if the document name already exists, the new output will appended to the end, which is called update mode. The other mode is write mode, which creates a new document and the document is overwritten if it already exist. In this example, write mode has been used.

To store the procedure from dataset, we only need to simply include DATA step and run the procedure in ODS DOCUMENT. In the case which I presented, there are 17 patients in the dataset, and all figures in format figure() by each subject are stored in ODS document WORK.FIGURE_ALL separately.

```
  ods listing close ;
     ods document name=work.figure_all (write);

     *** LOOP THROUGH SUBJECTS -- SUBJECT PER PAGE IN FIGURE ***;
     %do i=1 %to 17;
       *** DATA FOR AN INDIVIDUAL SUBJECT ***;
       data subject;
         set combine(where=(strip(country_subj) eq "&&subject&i."));
         by country_subj;
       run ;

       *** PLOT DATA FOR A SUBJECT ***;
       proc sgrender template=figure data=subject;
       run;
     %end;

     ods document close;

     ods listing;
```

Sometimes, we want to review the direction and contents in the ODS destination. The LIST statement in PROC DOCUMENT can help us to read information from the document (Figure 6). With the option of DETAILS in LIST statement, size, the data of created and modified, template and label of components in the document will be displayed with path and type. The output can help us to rearrange components in final report, which will be discussed later.

```
  ods listing ;
     proc document name=   work.figure_all;
       list/ levels=all;
     run ;
    ods listing close;
```



```
  Obs    Path                                                                                   Type
----------------------------------------------------------------------------------------------------
      1 \Sgrender#1                                                                             Dir
      2 \Sgrender#1\SGRender#1                                                                  Graph
      3 \Sgrender#2                                                                             Dir
      4 \Sgrender#2\SGRender#1                                                                  Graph
      5 \Sgrender#3                                                                             Dir
      6 \Sgrender#3\SGRender#1                                                                  Graph
      7 \Sgrender#4                                                                             Dir
      8 \Sgrender#4\SGRender#1                                                                  Graph
      9 \Sgrender#5                                                                             Dir
     10 \Sgrender#5\SGRender#1                                                                  Graph
     11 \Sgrender#6                                                                             Dir
     12 \Sgrender#6\SGRender#1                                                                  Graph
     13 \Sgrender#7                                                                             Dir
     14 \Sgrender#7\SGRender#1                                                                  Graph
     15 \Sgrender#8                                                                             Dir
     16 \Sgrender#8\SGRender#1                                                                  Graph
     17 \Sgrender#9                                                                             Dir
     18 \Sgrender#9\SGRender#1                                                                  Graph
     19 \Sgrender#10                                                                            Dir
     20 \Sgrender#10\SGRender#1                                                                 Graph
     21 \Sgrender#11                                                                            Dir
     22 \Sgrender#11\SGRender#1                                                                 Graph
     23 \Sgrender#12                                                                            Dir
     24 \Sgrender#12\SGRender#1                                                                 Graph
     25 \Sgrender#13                                                                            Dir
     26 \Sgrender#13\SGRender#1                                                                 Graph
     27 \Sgrender#14                                                                            Dir
     28 \Sgrender#14\SGRender#1                                                                 Graph
     29 \Sgrender#15                                                                            Dir
     30 \Sgrender#15\SGRender#1                                                                 Graph
     31 \Sgrender#16                                                                            Dir
     32 \Sgrender#16\SGRender#1                                                                 Graph
     33 \Sgrender#17                                                                            Dir
     34 \Sgrender#17\SGRender#1                                                                 Graph
```

Figure 6. Listed path for all components in document FIGURE_ALL.

Now that we created the document FIGURE_ALL, we need to replay our report. This is done with the REPLAY statement. REPLAY statement in PROC DOCUMENT procedure will replay all figures for those 17 patients (please see the appendix).

```
  ods listing;
      proc document name=   work.figure_all;
         replay;
      run ;
   ods listing close;
```

As we mentioned above, one of the advantage of ODS DOCUMENT is that we can rearrange the report. Like in the case we used in this paper, how to only replay partial figures, like from patient 1 to 4 only? There are couple ways to do it. One of them is to replay part of a report by specifying the object's path. In this way, we can use the REPLAY statement as follows:

```
    ods listing close;
    ods rtf  file="'x:\Desktop\Figures\Figures Catalog\Output\figure_all.rtf" nogtitle nogfootnote ;

      proc document name=work.figure_all;
         replay \work.figure_all\Sgrender#1,
                \work.figure_all\Sgrender#2,
                \work.figure_all\Sgrender#3,
                \work.figure_all\Sgrender#4;
      run;
```

```
ods rtf close ;
ods listing ;
```

The other way replay part of a report is to move the path of those components to create a new document as the following code:

```
proc document name=work.figure_part;
   %do i=1 %to 4;
      copy \work.figure_all\Sgrender#&i to ^;
   %end;
run ;

ods listing close;
ods rtf  file="'x:\Desktop\Figures\Figures Catalog\Output\figure_part.rtf" nogtitle nogfootnote ;
proc document name=work.figure_part;
   replay ;
run;
ods rtf close ;
ods listing ;
```

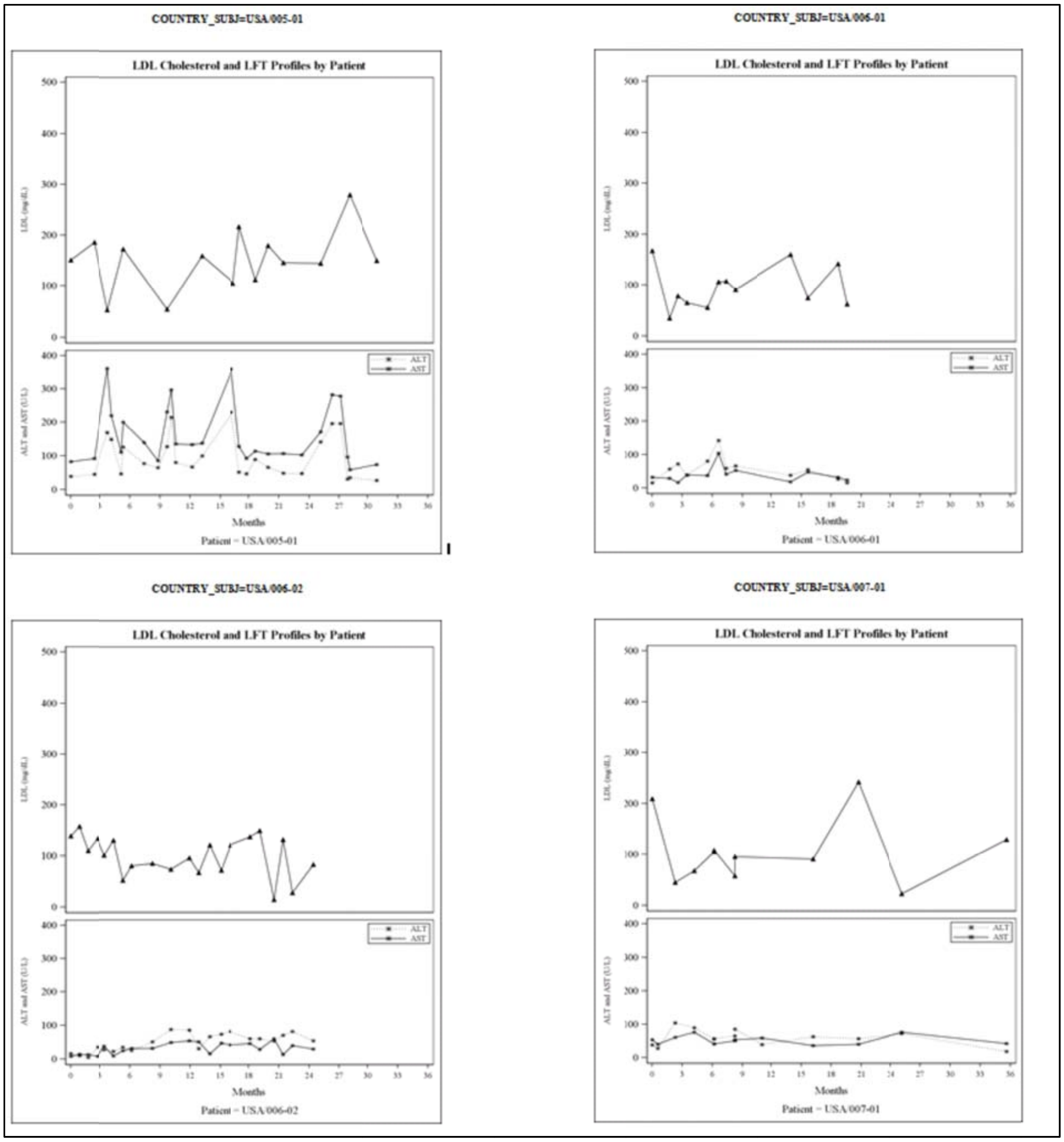


Figure 7. Only 4 patients are output.

Here, COPY statement help to move the component of a document to another document, which is like in Windows to copy, delete, and move elements in a document to another.

CONCLUSION

As you can see from the example of this paper,

1. PROC TEMPLATE procedure will give the new template a name with NAME option in DEFINE STATGRAPH statement ;
2. BEGINEGRAPH block allow to edit outer most container, such as size, title, footnote and so on;
3. Layout block allow to edit inner container and plots;
4. DISCRETELEGEND statement allow to edit the customized legend including position, entries and so on;
5. ODS DOCUMENT will store all figures or part of figures into one document and make sophisticate output.

CONTACT INFORMATION

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

Fangping Chen
UBC: An Express Scripts Company
Fangping.chen@ubc.com