ABSTRACT
As a visual report, graphical patient profile can improve the readability of correlated data. It is desirable to use for submissions and publications as a way to comprehensively present safety and efficacy data. A common time frame is usually used to combine multiple forms of data, such as lab results, vital signs, and clinical events with demographic information. This type of graph can be highly complex and customized to achieve reviewers’ needs. SAS’s Graph Template Language (GTL) is a powerful and flexible tool used to draw these graphs. This paper provides instructions to start building with GTL, and how to combine various pieces together like playing a LEGO game. Real world examples are given to allow the readers follow step-by-step with notes. The examples are generated by SAS® 9.4 and in publication quality.

INTRODUCTION
Patient profile are normally generated in a listing format. One output document file is created for each ID and presented in a certain order such as Demographic, Medical History, Adverse Event and Safety Results. This meets operational objectives when making patient safety decisions. However, for the efficacy results, visualization allows researchers and reviewers to better discover data trends in different ways, like using time series to track changes, or putting different results within same time frame together for correlation, or comparison before and after baseline.

The Statistical Graphics (SG) Procedures includes three easy-to-use syntax (SGPLOT, SGPANEL, and SGSCATTER) to create most commonly used analytical graphs. They have pre-defined graph templates which may lack of flexibility when several features in the graph templates need to be modified. SAS GTL is the core of ODS Graphics. One can design and build their own single template used for multiple scenarios. To get started on learning GTL, this paper explores the steps needed to create a complex graph in a way that’s similar to assembling a LEGO project, with GTL components treated as structured building-blocks. Rather than a comprehensive introduction, this paper serves as a tutorial for beginners of GTL. Three easy-to-hard examples are presented. One simply needs to follow the workflow and see how each individual block and statement works and the results produced.

BRICKS & PIECES
The parts to build a graph project can be categorized to these types:

**Statements:** define the contents of the graph; constructed by a keyword with arguments and options, mainly including plot, legend, text, draw statement.

**Blocks:** a pair of statements that constructs a block. Typically, multiple statements are nested within the block. A block can be nested within another block.

**Options:** add features to the statements

<table>
<thead>
<tr>
<th>Parts No.</th>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1001</td>
<td>SERIESPLOT</td>
<td>Series plot (Line plot)</td>
</tr>
<tr>
<td>S1002</td>
<td>SCATTERPLOT</td>
<td>Scatter plot</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>S1003</td>
<td>BARCHART</td>
<td>Bar chart</td>
</tr>
<tr>
<td>S1004</td>
<td>BOXPLOT</td>
<td>Box plot</td>
</tr>
<tr>
<td>S1020</td>
<td>REFERENCELINE</td>
<td>Creates a horizontal or vertical reference line; Only use within a 2-D layout, e.g. OVERLAY</td>
</tr>
<tr>
<td>S2001</td>
<td>ENTRYTITLE</td>
<td>To specify graph title</td>
</tr>
<tr>
<td>S2002</td>
<td>ENTRYFOOTNOTE</td>
<td>To specify graph footnote</td>
</tr>
<tr>
<td>S2003</td>
<td>DRAWTXT</td>
<td>Draws and anchors in a graph a text box that contains one or more lines of formatted text; location can be specified</td>
</tr>
<tr>
<td>S2004</td>
<td>DISCRETEATTRVAR</td>
<td>Creates an attribute-map variable to be referred in plot statement. Uses with DISCRETEATTRMAP block</td>
</tr>
<tr>
<td>S2005</td>
<td>LEGENDITEM</td>
<td>Creates the definition for a legend item that can be included in a discrete legend</td>
</tr>
<tr>
<td>S2006</td>
<td>DISCRETELEGEND</td>
<td>To specify legend features such as location, size, title etc.</td>
</tr>
</tbody>
</table>

**Knob & Tube (Blocks)**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B1001</td>
<td>PROC TEMPLATE; RUN;</td>
<td>To define, compile and save the template.</td>
</tr>
<tr>
<td>B1002</td>
<td>DEFINE STATGRAPH MyPlot; END;</td>
<td>Defines the structure of the graph and name it as e.g. 'MyPlot'</td>
</tr>
<tr>
<td>B1003</td>
<td>BEGINGRAPH; ENDGRAPH;</td>
<td>Builds content of the template within BEGINGRAPH block. Options which control graph size and appearance can be appended to BEGINGRAPH</td>
</tr>
<tr>
<td>B1099</td>
<td>PROC SGRENDER DATA= TEMPLATE= MyPlot; RUN;</td>
<td>To render pre-defined template to draw graph</td>
</tr>
<tr>
<td>B2001</td>
<td>LAYOUT OVERLAY; ENDLAYOUT;</td>
<td>Single-cell; When nested within another layout type, the OVERLAY layout defines the graph display for one cell of the parent layout</td>
</tr>
<tr>
<td>B2002</td>
<td>LAYOUT LATTICE; ENDLAYOUT;</td>
<td>Multi-cell; each cell can have different amount of space; Axes can be shared across columns or rows</td>
</tr>
<tr>
<td>B2003</td>
<td>LAYOUT GRIDDLED; ENDLAYOUT;</td>
<td>Multi-cell; each cell has the same amount of space; each cell is independent</td>
</tr>
<tr>
<td>B2021</td>
<td>LAYOUT GLOBALLEGEND</td>
<td>Combines all discrete and merged legends into one; One template allows only one of this type of block; Any DISCRETELEGEND or MERGEDLEGEND outside the block will be ignored</td>
</tr>
<tr>
<td>B2031</td>
<td>DISCRETEATTRMAP; ENDDISCRETEATTRMAP;</td>
<td>Define the attribute map; Associate the attribute map with a classification variable in the plot data, and can be referred in plot statement. This block is a direct child of BEGINGRAPH statement. It is outside of any other GTL blocks</td>
</tr>
</tbody>
</table>

**TILE (Options)**

**Managing Axes**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>O1001</td>
<td>XAXIS=X</td>
<td>X2</td>
</tr>
<tr>
<td>O1002</td>
<td>YAXIS=Y</td>
<td>Y2</td>
</tr>
<tr>
<td>O1003</td>
<td>XAXISOPTS=</td>
<td>To define X-axis;</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>O1004</td>
<td>X2AXISOPTS= To define 2nd X-axis</td>
<td></td>
</tr>
<tr>
<td>O1005</td>
<td>YAXISOPTS= To define Y-axis</td>
<td></td>
</tr>
<tr>
<td>O1006</td>
<td>Y2AXISOPTS= To define 2nd Y-axis</td>
<td></td>
</tr>
</tbody>
</table>

**Controlling Grouped Data**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1101</td>
<td>GROUP= Sets classification variable in addition to X and Y variables; By default, the plot automatically cycles through appearance features (colors, line styles, and marker symbols) to distinguish group values in the plot</td>
</tr>
<tr>
<td>O1102</td>
<td>GROUPDISPLAY= To specify how group values are displayed, e.g. STACK: Stacks each group value on a single bar CLUSTER: side-by-side</td>
</tr>
</tbody>
</table>

**Plot Appearance**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O2001</td>
<td>LINEATTRS= To specify the appearance of the line, e.g. color, pattern, thickness</td>
</tr>
<tr>
<td>O2002</td>
<td>MARKERATTRS= To specify the appearance of markers, e.g. color, size, symbol</td>
</tr>
<tr>
<td>O2003</td>
<td>TEXTATTRITRS= Changes attributes of the text, e.g. font, font size, or color for title, footnote and even label</td>
</tr>
<tr>
<td>O2004</td>
<td>LABELATTRS= To specify axis label features, used in XAXISOPS or YAXISOPS statement.</td>
</tr>
<tr>
<td>O2005</td>
<td>TICKVALUEATTRS= To specify axis tick features, used in XAXISOPS or YAXISOPS statement.</td>
</tr>
<tr>
<td>O2006</td>
<td>CYCLEATTRS= Assigns different visual attributes to each plot.</td>
</tr>
<tr>
<td>O2007</td>
<td>FILLATTRS= To Specify the fill color and transparency</td>
</tr>
<tr>
<td>O2008</td>
<td>BARLABELATTRS= To Specify the text properties of the bar label text</td>
</tr>
<tr>
<td>O2009</td>
<td>VALUEATTRS= Controls the text properties of the legend. By default, the legend entries use VALUEATTRS = GraphValueText.</td>
</tr>
<tr>
<td>O2101</td>
<td>DISCRETEOPTS= Settings for discrete axes. This option is ignored if the axis type is not DISCRETE</td>
</tr>
<tr>
<td>O2102</td>
<td>LINEAROPTS = Settings for continuous data</td>
</tr>
<tr>
<td>O2103</td>
<td>TIMEOPTS= Settings for Time series data, such as Date, Time, or Datetime values</td>
</tr>
<tr>
<td>O2104</td>
<td>LOGOPTS = Settings for a logarithmic scale</td>
</tr>
<tr>
<td>O2901</td>
<td>COLOR= GTL User Guide Appendix 4</td>
</tr>
<tr>
<td>O2902</td>
<td>PATTERN= GTL User Guide Appendix 3 P.624</td>
</tr>
<tr>
<td>O2903</td>
<td>SYMBOL= GTL User Guide Appendix 3 P.622</td>
</tr>
</tbody>
</table>

**Plot Identification**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O3001</td>
<td>NAME= To specify a name by which this statement can be referenced in a LEGEND statement.</td>
</tr>
<tr>
<td>O3002</td>
<td>LEGENDLABEL= To specify legend label and can override the default legend text for each plot</td>
</tr>
</tbody>
</table>

**Labels for Plot Features**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O4001</td>
<td>DATALABEL= To specify the name of a variable that is used to label observations, as a column to label data points in a scatter plot, series plot, etc</td>
</tr>
<tr>
<td>O4002</td>
<td>DATALABELPOSITION= To specify the position of the data labels relative to the data points and arrow heads</td>
</tr>
</tbody>
</table>
PROJECTS

Project I: Single-cell graph with line plot and bar chart overlaid (Figure 1)

![Graph Image]

**Figure 1 - Individual Patient Profile of Lab Test and Medication Dose over Time**

<table>
<thead>
<tr>
<th>Parts</th>
<th>Building Instruction &amp; Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1001</td>
<td>This graph is single-cell to use draw three different tests into one graph with sharing one x-axis.</td>
</tr>
<tr>
<td>B1002</td>
<td>* To create template Myplot;</td>
</tr>
</tbody>
</table>
| B1003  | proc template;
|        | define statgraph myplot;
|        | begingraph /designwidth=800px designheight=400px;
|        | endgraph;
|        | end;
|        | run;
|        | NOTE: STATGRAPH 'Myplot' has been saved to: WORK.TEMPLAT
| B1099  | * A blank graph is produced; |
| B2001  | proc template;
|        | define statgraph myplot;
|        | begingraph/designwidth=800px designheight=400px;
|        | layout overlay;
|        | endlayout;
|        | endgraph;
|        | end;
|        | run;
|        | proc sgrender data=final template=myplot;
|        | run;

**SAS default graph size is 640x480 pixels (4:3 aspect). Need to adjust if legend is not displayed.**
S1001

* Series plot added in LAYOUT OEVERLAY;

```
proc template;
  define statgraph myplot;
    begingraph;
      layout overlay;
        seriesplot x=visitn y=cfcdur;
      endlayout;
    endgraph;
  end;
run;
```

S1002

* Overlaying Scatter plot on Series plot;

```
proc template;
  define statgraph myplot;
    begingraph;
      layout overlay;
        seriesplot x=visitn y=cfcdur;
        scatterplot x=visitn y=cfcdur;
      endlayout;
    endgraph;
  end;
run;
```
* Add options to control appearance of plots;

``` SAS
proc template;
   define statgraph myplot;
   begingraph;
      layout overlay ;
         seriesplot x=visitn y=cfcdur/lineattrs=(thickness=1 color=black pattern=1);
         scatterplot x=visitn y=cfcdur/markerattrs=(size=8 color=black symbol=squarefilled);
      endlayout;
   endgraph;
end;
run;
```

* Overlaying more plots or bar charts;

``` SAS
proc template;
   define statgraph myplot;
   begingraph;
      layout overlay ;
         seriesplot x=visitn y=cfcdur / lineattrs=(thickness=2.5 color=black pattern=1);
         scatterplot x=visitn y=cfcdur / markerattrs=(size=8 color=black symbol=squarefilled);
         barchart x=visitn y=pns / barwidth=0.3;
         seriesplot x=visitn y=alt / lineattrs=(thickness=2.5 color=blue pattern=1);
         scatterplot x=visitn y=alt / markerattrs=(size=8 color=blue symbol=circlefilled transparency = .05);
      endlayout;
   endgraph;
end;
run;
```

Multiple Y-axis scales are laying over each other, which needs to be differentiated by defining different Y-axis.
* Pre-defined a set of axis options by using LAYOUT OVERLAY statement: X-axis, Y-axis and Y2-axis; Macro variables can define the maximum or minimum value of each scale.

```sas
%let xvalue=%str(0 4 8 14 22 28 34 38 42 51 58 66 70 78 85 89 98 108 114 120);
xaxisopts =
(type=linear
  offsetmin=0.03 offsetmax=0.01
  tickvalueattrs=(size=8pt weight=bold)
  linearopts=(viewmin=0 viewmax=120 tickvaluelist=(&xvalue)
    tickvaluefitpolicy=rotate)
  labelattrs=(size=11 weight=bold
    label='Study Day'))
yaxisopts =
  (griddisplay=on offsetmin=0 offsetmax=0.01
    tickvalueattrs=(size=10pt weight=bold)
    linearopts=(tickvaluesequence=(start=0 end=12 increment=3) viewmin=0 viewmax=12)
    labelattrs=(size=12 weight=bold color=black
      label="Duration (hours) of Event")
y2axisopts =
  (offsetmin=0 offsetmax=0.00
    tickvalueattrs=(size=10pt weight=bold)
    linearopts=(tickvaluesequence=(start=0 end=&maxy increment=10) viewmin=0 viewmax=&maxy)
    labelattrs=(size=0 label='');
```

* Use pre-defined axis options to draw multiple plots with one shared axis;

```sas
proc template;
  define statgraph myplot;
  begingraph;
    layout overlay/
      xaxisopts =..;
    endgraph;
  end;
```
Tip: The order of S1xxx Statement determines which plot presents in front. The first plot that specified in the layout block is drawn first. The second plot is then added on top of the first plot, and so on. It is possible for one plot's data to obscure the data beneath it.

O1002
O2001

* Adjust layers of plots;
* Add reference line using series plot;
* Adjust height and width;

S2003 x 2
O2003 x 2

* Add customized Y2-Axis Label using DRAWTEXT in LAYOUT OVERLAY block;
proc template;
   define statgraph myplot;
   begingraph/designwidth=800px designheight=350px pad=(right=7%);
      layout overlay /
         xaxisopts =...
         yaxisopts =...
         y2axisopts =...
         barchart...
         seriesplot...
         seriesplot...
         scatterplot...
         scatterplot...
         drawtext textattrs=(size=12 weight=bold color=blue) "Lab Test A (U/L)" /
            xspace=graphpercent yspace=wallpercent x=96 y=50 rotate=90 anchor=center
            width=50;
         drawtext textattrs=(size=12 weight=bold color=grey) "Medication A (mg/day)" /
            xspace=graphpercent yspace=wallpercent x=98 y=50 rotate=90 anchor=center
            width=50;
      endlayout;
   endgraph;
end;

* Add Title and Global Legend;

proc template;
   define statgraph myplot;
   begingraph/designwidth=800px designheight=350px border=false pad=(right=7%);
   layout overlay /
      xaxisopts =...
      yaxisopts =...
      y2axisopts =...
      barchart...
      seriesplot...
      seriesplot...
      scatterplot...
      scatterplot...
      drawtext textattrs=(size=12 weight=bold color=blue) "Lab Test A (U/L)" /
         xspace=graphpercent yspace=wallpercent x=96 y=50 rotate=90 anchor=center
         width=50;
      drawtext textattrs=(size=12 weight=bold color=grey) "Medication A (mg/day)" /
         xspace=graphpercent yspace=wallpercent x=98 y=50 rotate=90 anchor=center
         width=50;
   endlayout;
   endgraph;
end;
Project II: Multi-cell graph with data grouping, customized by Attribute Maps (Figure 2)
Figure 2 - Diet Diary Average Values Over Time: Pre- and Post-Baseline Period

<table>
<thead>
<tr>
<th>Parts</th>
<th>Building Instruction &amp; Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This graph is multi-cell. The top one presents the overall result value while the bottom one is the percentage for each component for the corresponding Total Caloric Intake as above. Two graphs share one x-axis with different y-axis.</td>
</tr>
</tbody>
</table>
| B1001 B1002 B1003 | * To create template Mybarchart:  
proc template;  
define statgraph mybarchart;  
begingraph;  
  endgraph;  
end;  
run;  
NOTE: STATGRAPH 'Mybarchart' has been saved to: WORK.TEMPLATE |
| B2001 X 2 B2002 S1003 X 2 B1099 | * To create two-cell graph using LAYOUT LATTICE without additional options;  
proc template;  
define statgraph mybarchart;  
begingraph;  
  layout lattice / columns=1 rows=2 rowweights=(0.4 0.6);  
  row1  
    layout overlay;  
      barchart x=avisitn y=totci;  
      endlayout;  
  end;  
  row2  
    layout overlay;  
      barchart x=avisitn y=aval;  
      endlayout;  
  endgraph;  
end;  
run;  
proc sgrender data=final template=mybarchart;  
run;  
quit;  
Define proportion size of each cell |
* Combine two x-axis and only show at bottom;

```
proc template;
  define statgraph mybarchart;
  begingroup;
    layout lattice / columns=1 rows=2 rowweights=(0.4 0.6);
    layout overlay /
      xaxisopts=(label=" " display=none)
      yaxisopts=(label="Total Caloric Intake" labelattrs=(size=9pt weight=bold) tickvalueattrs=(size=9pt weight=bold))
    ;
    barchart x=avisitn y=totci;
    endgroup;
    layout overlay /
      xaxisopts=(label=" " tickvalueattrs=(size=9pt weight=bold) discreteopts=(tickvaluelist=(&xticklst))
      yaxisopts=(label="% of calories Nutrients" labelattrs=(size=9pt weight=bold) tickvalueattrs=(size=9pt weight=bold))
    ;
    barchart x=avisitn y=aval;
    endgroup;
  endgroup;
endgroup;
run;
```

Tip: Step 4 pre-defined a set of variable attributes. DISCRETEATTRMAP block is a direct child of BEGINGRAPH statement. This method can dynamically assign colors to plots.

```
%macro attmap;
  discreteattrmap name="symbols" / ignorecase=true ;
  value '5'/fillattrs=(color=yellow) lineattrs=(color=black);
  value '4'/fillattrs=(color=green) lineattrs=(color=black);
  value '3'/fillattrs=(color=brown) lineattrs=(color=black);
  value '2'/fillattrs=(color=lightblue) lineattrs=(color=black);
  value '1'/fillattrs=(color=orange) lineattrs=(color=black);
  enddiscreteattrmap;
  discreteattrvar attrvar=groupmarkers var=paramn attmap="symbols" ;

  discreteattrmap name="symbols2" / ignorecase=true ;
  value '0'/fillattrs=(color=purple) lineattrs=(color=black);
  enddiscreteattrmap;
  discreteattrvar attrvar=groupmarkers2 var=paramn attmap="symbols2" ;
```

```
* Color block bar chart by using defined attribute map (Macro %attmp):

```plaintext
proc template;
    define statgraph mybarchart;
    begingraph;
    %attmap;
    layout lattice / columns=1 rows=2 rowweights=(0.4 0.6);
    layout overlay /
        xaxisopts=(label=" " display=none)
        yaxisopts=(label="Total Caloric Intake" labelattrs=(size=9pt weight=bold)
            tickvalueattrs=(size=9pt weight=bold))
    ;
    barchart x=avisitn y=totci / group=groupmarkers2 groupdisplay=stack;
    endlayout;

    layout overlay /
        xaxisopts=(label=" "
            tickvalueattrs=(size=9pt weight=bold)
        )
        discreteopts=(tickvaluelist=(&xticklst))
    ;
    barchart x=avisitn y=aval / group=groupmarkers groupdisplay=stack;
    endlayout;
    endgraph;
end;
run;
```

* Adjust appearance of the charts, e.g. width of bar;
* Add statistical results using MARKERCHARACTER;
proc template;
define statgraph mybarchart;
begingraph;
%
attmap;
layout lattice / columns=1 rows=2 rowweights=(0.4 0.6);
layout overlay /
xaxisopts=(label="" display=none)
yaxisopts=(label="Total Caloric Intake" labelattrs=(size=9pt weight=bold)
tickvalueattrs=(size=9pt weight=bold))
;
barchart x=avisitn y=totci / group=groupmarkers2 groupdisplay=stack
barlabelattrs=(size=9pt) barwidth=0.4 barlabel=true;
endlayout;
layout overlay /
xaxisopts=(label="" tickvalueattrs=(size=9pt weight=bold)
discreteopts=(tickvaluelist=(&xticklst))
yaxisopts=(label="% of calories of Nutrients" labelattrs=(size=9pt weight=bold) tickvalueattrs=(size=9pt weight=bold))
;
barchart x=avisitn y=aval / group=groupmarkers groupdisplay=stack
barwidth=0.4;
scatterplot x=avisitn y=accupct /group=groupmarkers index=paramn
groupdisplay=overlay markercharacter=aval markercharacterposition=center
markercharacterattrs=(color=black size=9pt);
endlayout;
endlayout;
endgraph;
end;
run;

S2006
O2009
* Add legend and title; The display order for the color is defined in DISCRETELEGEND inside LAYOUT OVERLAY block.

Use bar label to add statistical numbers above the bar

Use scatter plot to add statistical numbers within the bar
proc template;
define statgraph mybarchart;
begingraph;
%attmap;
layout lattice / columns=1 rows=2 rowweights=(0.4 0.6);
layout overlay / xaxisopts=(label="" display=none) yaxisopts=(label="Total Caloric Intake" labelattrs=(size=9pt weight=bold) tickvalueattrs=(size=9pt weight=bold));
barchart x=avisitn y=totci / group=groupmarkers2 groupdisplay=stack barlabel=true barlabelattrs=(size=9pt) barwidth=0.4;
endlayout;

layout overlay / xaxisopts=(" Total Caloric Intake" labelattrs=(size=9pt weight=bold) tickvalueattrs=(size=9pt weight=bold) discreteopts=(tickvaluelist=(&xticklst))) yaxisopts=("% of calories of Nutrients" labelattrs=(size=9pt weight=bold) tickvalueattrs=(size=9pt weight=bold));
barchart x=avisitn y=aval / group=groupmarkers groupdisplay=stack barwidth=0.4;
scatterplot x=avisitn y=accupct / group=groupmarkers index=paramn groupdisplay=overlay markercharacter=aval markercharacterposition=center markercharacterattrs=(color=black size=9pt);

discretelegend "N_A" "N_B" "N_C" "N_D" "N_E"/ border=false location=outside halign=left across=1 valueattrs=(size=10pt weight=bold) title="";
endlayout;
endgraph;
end;
run;
Project III: Structure a Complex dataset to facilitate multiple features on one data point (Figure 3)

**Figure 3 - Distribution of Major Clinical Events**

**Analysis Data:**

<table>
<thead>
<tr>
<th>Patient ID</th>
<th>Major Clinical Event</th>
<th>Event Type</th>
<th>Precipitating Factor</th>
<th>Study Day</th>
<th>Duration</th>
<th>Duration Category</th>
<th>id</th>
<th>Data Collection Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxx-xxx</td>
<td>MCE A</td>
<td>Emergency Intervention</td>
<td>I</td>
<td>25</td>
<td>1</td>
<td>2-1</td>
<td>1</td>
<td>PRE-TREATMENT</td>
</tr>
<tr>
<td>xxx-xxx</td>
<td>MCE C</td>
<td>Emergency Intervention</td>
<td>E</td>
<td>151</td>
<td>3</td>
<td>3-3</td>
<td>1</td>
<td>POST-TREATMENT</td>
</tr>
<tr>
<td>xxx-xxx</td>
<td>MCE A</td>
<td>Hospitalization</td>
<td>U</td>
<td>276</td>
<td>10</td>
<td>4-5</td>
<td>1</td>
<td>POST-TREATMENT</td>
</tr>
<tr>
<td>xxx-xxx</td>
<td>MCE A</td>
<td>Hospitalization</td>
<td>I</td>
<td>420</td>
<td>1</td>
<td>2-6</td>
<td>1</td>
<td>POST-TREATMENT</td>
</tr>
<tr>
<td>xxx-xxx</td>
<td>MCE B</td>
<td>Emergency Room Visit</td>
<td>O</td>
<td>-387</td>
<td>6</td>
<td>3-1</td>
<td>0</td>
<td>PRE-TREATMENT</td>
</tr>
<tr>
<td>xxx-xxx</td>
<td>MCE A</td>
<td>Hospitalization</td>
<td>I</td>
<td>-239</td>
<td>3</td>
<td>3-2</td>
<td>0</td>
<td>PRE-TREATMENT</td>
</tr>
<tr>
<td>xxx-xxx</td>
<td>MCE A</td>
<td>Hospitalization</td>
<td>I</td>
<td>-80</td>
<td>2</td>
<td>3-3</td>
<td>0</td>
<td>PRE-TREATMENT</td>
</tr>
</tbody>
</table>

* Use different x-variables to present event type to be defined with different feature in graph;

* Use different y-variables to present duration group to be defined with different feature in graph;

```plaintext
data final;
  set dsin;
  if evtype='Hospitalization' then ady_h=ady;
  else if evtype='Emergency Room Visit' then ady_r=ady;
  else if evtype='Emergency Intervention' then ady_i=ady;
  if evtype='Hospitalization' then do;
    if durgrp=1 then y1=id;
    else if durgrp=2 then y2=id;
    else if durgrp=3 then y3=id;
    else if durgrp=4 then y4=id;
  end;
  else y=id;
run;
```
**Data ready-for-graph:**

<table>
<thead>
<tr>
<th>parts</th>
<th>Major Clinical Event</th>
<th>Event Type</th>
<th>Precipitation Event</th>
<th>Study Day</th>
<th>Duration Category</th>
<th>id</th>
<th>Data Collection Period</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>w</th>
<th>v</th>
<th>u</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxxx</td>
<td>MCE A</td>
<td>Emergency Event</td>
<td>I</td>
<td>26</td>
<td>POST-TREATMENT</td>
<td>.</td>
<td>.</td>
<td>26</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxx</td>
<td>MCE C</td>
<td>Emergency Event</td>
<td>I</td>
<td>28</td>
<td>POST-TREATMENT</td>
<td>.</td>
<td>.</td>
<td>28</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxx</td>
<td>MCE A</td>
<td>Hospitalization</td>
<td>E</td>
<td>151</td>
<td>POST-TREATMENT</td>
<td>.</td>
<td>.</td>
<td>151</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxx</td>
<td>MCE C</td>
<td>Hospitalization</td>
<td>E</td>
<td>181</td>
<td>POST-TREATMENT</td>
<td>.</td>
<td>.</td>
<td>181</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxx</td>
<td>MCE A</td>
<td>Hospitalization</td>
<td>U</td>
<td>276</td>
<td>POST-TREATMENT</td>
<td>.</td>
<td>.</td>
<td>276</td>
<td>-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxx</td>
<td>MCE A</td>
<td>Hospitalization</td>
<td>I</td>
<td>420</td>
<td>POST-TREATMENT</td>
<td>.</td>
<td>.</td>
<td>420</td>
<td>-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxx</td>
<td>MCE A</td>
<td>Hospitalization</td>
<td>O</td>
<td>-387</td>
<td>PRE-TREATMENT</td>
<td>.</td>
<td>.</td>
<td>-387</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxx</td>
<td>MCE A</td>
<td>Hospitalization</td>
<td>I</td>
<td>-239</td>
<td>PRE-TREATMENT</td>
<td>.</td>
<td>.</td>
<td>-239</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxx</td>
<td>MCE A</td>
<td>Hospitalization</td>
<td>I</td>
<td>-80</td>
<td>PRE-TREATMENT</td>
<td>.</td>
<td>.</td>
<td>-80</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Parts**

**Building Instruction & Result**

This graph is to structure a dataset first in order to draw different shapes, different sizes and different colors per different categories (Event Type, Duration of Event and MCE) on different time slots (Pre/Post-Treatment).

*Use scatter plot to display event;*

```
proc template;
  define statgraph mybubble;
  begingraph;
    layout overlay;
      scatterplot x=ady_h y=y1 ;
      scatterplot x=ady_h y=y2 ;
      scatterplot x=ady_h y=y3 ;
      scatterplot x=ady_h y=y4 ;
      scatterplot x=ady_i y=y ;
      scatterplot x=ady_r y=y ;
    endlayout;
  endgraph;
end;
run;
ods graphics on;
proc sgrender data=final template=mybubble;
run;
ods graphics off;
```

Ignore this label due to six scatter plots overlaid and shared X-axis.
* Add Event factor (I, E, U, O) using DATALABEL;

```plaintext
proc template;
  define statgraph mybubble;
  begingraph;
    layout overlay;
      scatterplot x=ady_h y=y1 / datalabel=prevt datalabelposition=left ;
      scatterplot x=ady_h y=y2 / datalabel=prevt datalabelposition=left ;
      scatterplot x=ady_h y=y3 / datalabel=prevt datalabelposition=left ;
      scatterplot x=ady_h y=y4 / datalabel=prevt datalabelposition=left ;
      scatterplot x=ady_i y=y  / datalabel=prevt datalabelposition=left ;
      scatterplot x=ady_r y=y  / datalabel=prevt datalabelposition=left ;
    endlayout;
  endgraph;
end;
run;
```
* To define X- and Y-axis;

```
proc template;
  define statgraph mybubble;
  begingraph;
    layout overlay /
      xaxisopts=(griddisplay=on labelattrs=(size=8 weight=bold) label="Study Day" linearopts=(tickvaluelist=(&xlabel) viewmin=-500 viewmax=500) offsetmin=0.01 offsetmax=0.01 tickvalueattrs=(size=8pt))
      yaxisopts=(griddisplay=on labelattrs=(size=1) label="" discreteopts=(tickvaluefitpolicy=splitalways tickvaluelist=(&ylabel) tickvalueattrs=(size=8) offsetmin=0.1 offsetmax=0.1)
    ;
    scatterplot x=ady_h y=y1 / datalabel=prevt datalabelposition=left ;
    scatterplot x=ady_h y=y2 / datalabel=prevt datalabelposition=left ;
    scatterplot x=ady_h y=y3 / datalabel=prevt datalabelposition=left ;
    scatterplot x=ady_h y=y4 / datalabel=prevt datalabelposition=left ;
    scatterplot x=ady_i y=y / datalabel=prevt datalabelposition=left ;
    scatterplot x=ady_r y=y / datalabel=prevt datalabelposition=left ;
    endlayout;
  endgraph;
end;
run;
```

LINEAROPTS: for continuous variable, here for X role

DISCRETEOPTS: for discrete (categorical, ordinal) variable, here for Y role. If a numeric variable is assigned to the Y role, it will be treated as discrete values, never as a continuous range of values
* Add Marker attributes;

```sas
proc template;
  define statgraph mybubble;
  begingraph;
    layout overlay / 
      xaxisopts.. 
      yaxisopts.. 
    ;
    scatterplot x=ady_h y=y1 / datalabel=prevt datalabelposition=left 
      markerattrs=(symbol=circleFilled size=4pt);
    scatterplot x=ady_h y=y2 / datalabel=prevt datalabelposition=left 
      markerattrs=(symbol=circlefilled size=6pt);
    scatterplot x=ady_h y=y3 / datalabel=prevt datalabelposition=left 
      markerattrs=(symbol=circleFilled size=9pt);
    scatterplot x=ady_h y=y4 / datalabel=prevt datalabelposition=left 
      markerattrs=(symbol=circleFilled size=12pt);
    scatterplot x=ady_i y=y / datalabel=prevt datalabelposition=left 
      markerattrs=(symbol=diamondfilled size=5.5pt);
    scatterplot x=ady_r y=y / datalabel=prevt datalabelposition=left 
      markerattrs=(symbol=squarefilled size=6pt);
    endlayout;
  endgraph;
end;
run;
```

* Data are displayed by defined attribute map;

```sas
proc template;
  define statgraph mybubble;
  begingraph;
    discreteattrmap name="symbols" / ignorecase=true ;
    value "MCE A" / markerattrs=(color=black) ;
    value "MCE B" / markerattrs=(color=blue) ;
    value "MCE C" / markerattrs=(color=red) ;
    endgraph;
  end;
end;
```
Tip: Step 5 the DISCRETEATTRVAR statement associates the attribute map with input data column 'Major Clinical Event' and assigns the name GROUPMARKERS to the named association.

* Create legend entries for the markers:

```plaintext
legenditem type=marker name='hosp'/markerattrs=(symbol=circlefilled color=black size=6pt) label='Hospitalization';
legenditem type=marker name='erv'/markerattrs=(symbol=squarefilled color=black size=6pt) label='Emergency Room Visit';
legenditem type=marker name='ei'/markerattrs=(symbol=diamondfilled color=black size=5.5pt) label='Emergency Intervention';
legenditem type=marker name='blank1'/markerattrs=(color=white size=8pt) label='';
legenditem type=marker name='t1'/markerattrs=(color=white size=8pt) label='Duration of Event (days)';
legenditem type=marker name='d1'/markerattrs=(symbol=circlefilled color=black size=4pt) label='Missing';
legenditem type=marker name='d2'/markerattrs=(symbol=circlefilled color=black size=6pt) label='1';
legenditem type=marker name='d3'/markerattrs=(symbol=circlefilled color=black size=9pt) label='2-7';
legenditem type=marker name='d4'/markerattrs=(symbol=circlefilled color=black size=12pt) label='>=8';
legenditem type=marker name='blank2'/markerattrs=(color=white size=8pt) label='';
legenditem type=marker name='t2'/markerattrs=(color=white size=8pt) label='Major Clinical Event';
legenditem type=marker name='mcea'/markerattrs=(symbol=circlefilled color=black size=6pt) label='MCE A';
legenditem type=marker name='mceb'/markerattrs=(symbol=circlefilled color=blue size=6pt) label='MCE B';
legenditem type=marker name='mcec'/markerattrs=(symbol=circlefilled color=red size=6pt) label='MCE C';
legenditem type=text name='I'/textattrs=(family="Arial" color=black size=8pt weight=bold) text='I - Infectious Disease';
legenditem type=text name='E'/textattrs=(family="Arial" color=black size=8pt weight=bold) text='E - Exercise';
legenditem type=text name='O'/textattrs=(family="Arial" color=black size=8pt weight=bold) text='O - Other';
legenditem type=text name='U'/textattrs=(family="Arial" color=black size=8pt weight=bold) text='U - Unknown';
```
Add a legend on the right and the bottom:

```sas
proc template;
    define statgraph mybubble;
    begingraph;
        discreteattrmap..
        enddiscreteattrmap;
        discreteattrvar..;
        legenditem..
        layout overlay /
            xaxisopts..
            yaxisopts..
            legend..
            discretelegend 'hosp' 'erv' 'ei' 'blank1' 't1' 'd1' 'd2' 'd3' 'd4' 'blank2' 't2' 'mcea' 'mceb' 'mcec' / title="Event Type" titleattrs=(size=8 weight=bold) across=1 border=false location=outside halign=right displayclipped=true valueattrs=(size=8 weight=bold);
            discretelegend 'I' 'E' 'O' 'U' / title="Precipitating Factor: " titleattrs=(size=8 weight=bold) down=2 border=false location=outside valign=bottom displayclipped=true;
        endlayout;
    endgraph;
end;
run;
ods graphics on / maxlegendarea=30;
```

```sas
proc sgrender data=final template=mybubble;
run;
ods graphics off;
```

Specifies the maximum percentage of the overall graph area that a legend can occupy. The default is 20. Larger legends are dropped from the display.
CONCLUSION

This paper demonstrates how to generate Patient Profile graphs by using SAS GTL. The key is to identify which procedures are the blocks, knob & tube or tiles and what the relationships between them such as, which information present first, which line is on the top and which features need to be customized-defined. Once we get familiar with the parts and structures and also develop a blueprint, using tool of GTL can be as fun as playing LEGO games. We only illustrated the steps we think to be easy-following. But there are different ways to plug-in or broken-down. The final outputs can be more creative.
REFERENCES

RECOMMENDED READING
• Graph Template Language Tip Sheet.

DISCLAIMER
The scope of this paper is to present the opinions and suggestions from the authors. It does not reflect recommendations or practices of Ultragenyx Pharmaceutical, Inc. The data used in this paper is simulation data.

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

Ruohan Wang
Ultragenyx Pharmaceutical Inc.
60 Leveroni Court
Novato, CA 94949
415-483-8800
rwang@ultragenyx.com

Chris Qin
Ultragenyx Pharmaceutical Inc.
60 Leveroni Court
Novato, CA 94949
(415) 483-8800
cqin@ultragenyx.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.