Tips and Tricks for Clinical Graphs using ODS Graphics

Sanjay Matange, SAS Institute Inc., Cary, NC

ABSTRACT

Did you know that you can create an adverse event graph using a vector plot? Or, that you can label dosage levels for a medications plot using a scatter plot? How do you place a reference line between two values on a category axis? Statistical graphics (SG) procedures and the graph template language (GTL) provide you myriad ways to mix and match statements to create graphs. What you can achieve is based on creative usage of the statements.

This presentation includes tips and tricks you can use in SG procedures and GTL programs to build your graphs. We use examples from clinical trials and health and life sciences domains to illustrate the techniques using real-world graphs like LFT panels, patient profiles, adverse event plots, and more.

Most examples use the second maintenance release of SAS® 9.2, but this presentation also includes a sneak preview of some powerful new features to be released with SAS® 9.3.

INTRODUCTION

High quality graphs are essential for the analysis of data in the Health and Life Sciences domain. Large volumes of such data are generated in the duration of a clinical trial. These data are often collected and then presented in tabular form. However, analysis and understanding of the results are greatly enhanced by graphical presentation of the data, along with the derived statistics in the same graph. Some of the key features of such graphs include:

- Inclusion of the raw data and the statistics in the same display
- Inclusion of other indicators such as desirable levels, adverse events, and so on
- Comparison of the results for a drug with the corresponding results for other drugs or placebo
- Classification of the results by multiple variables and display over time.

The SAS® Statistical Graphics (SG) Procedures and Graph Template Language (GTL) are powerful tools to create such graphs. They use a flexible "building-block" approach to creating graphs, from the simplest scatter plot to more complex graphs and panels common in the Health and Life Sciences industry.   These tools support plot, layout, axes, insets, and other statements, all of which are like the ingredients in a recipe. These ingredients can be used in unexpected and surprising ways to create novel and complex graphs.   No annotation is needed.

The SG procedures and GTL are part of the ODS Graphics System for creation of modern analytical graphs. GTL forms the basis of all graphs rendered using the ODS Graphics system.

Graph Template Language

This is the syntax used to define a STATGRAPH template in the TEMPLATE procedure. This template is then associated with the appropriate data to create the graph. This is done by using the SGRENDER procedure. The ingredients listed below can be combined in creative ways to build all types of graphs.
- **Layouts**:   Overlay, OverlayEquated, Gridded, Lattice, DataLattice, DataPanel, and Region
- **Plots**:   Scatter, Series, Step, Histogram, Density, BoxPlots, BarChars, Fit plots, and more
- **More Plots**: BlockPlot, Ellipse, LIneParm, Reference and Drop Lines, HighLow, Bubble, Pie, and so on
- **Other**:   EntryTitle, EntryFootnote, Entry, DiscreteLegend, ContinuousLegend, and so on
- **Features**:   Functions, conditionals, dynamics, and macro variables.

SG Procedures

These are "value added" wrappers on GTL. These present a familiar interface to the user and are optimized to build the most commonly used graphs.
- **SGPlot:**   Build single-cell plots.
- **SGPanel:**   Build multi-cell classification panels.
- **SGScatter:** Build multi-cell plots, comparative plots, and matrices

1. SURVIVAL Plot

The survival plot is a commonly used plot in the Health and Life Sciences industry, and shows the survival estimates over time by treatment. In this example, we have output the data from PROC LIFETEST (Example 49.2 [1]) using the ODS output statement to write the data from the "Survival Plot" object to the "SurvivalPlot_49_2_1" data set as shown in Figure 1.1.

```
   ods output Survivalplot=sasuser.SurvivalPlot49_2_1;
  proc lifetest data=BMT plots=survival(atrisk=0 to 2500 by 500);
     time T * Status(0);
     strata Group / test=logrank adjust=sidak;
       run;
```
Figure 1.1

The data set looks like the table shown in Figure 1.2. The data includes the survival probabilities over time for three strata of Leukemia. Only a small subset of the observations is shown here to conserve space.

Obs	Time	Survival	AtRisk	Event	Censored	tAtRisk	Stratum	StratumNum
1	0	1.00000	38	0	.	.	1: ALL	1
2	0	.	38	.	.	0	1: ALL	1
3	1	0.97368	38	1	.	.	1: ALL	1
4	55	0.94737	37	1	.	.	1: ALL	1

Figure 1.2

The survival plot shown in the output from the LIFETEST procedure Example 49.2 is a single-cell graph that can be created directly using the SGPLOT procedure and the following steps.

Step1: Create the graph with the survival curves. We will plot Survival x Time with Group=Stratum as shown in Figure 1.3. The resulting graph is shown in Figure 1.4.   Note the group legend is generated automatically.

```
  title 'Product-Limit Survival Estimates';
 proc sgplot data=sgf2011.SurvivalPlot49_2_1;
    step x=time y=survival / group=stratum name='survival';
    run;
```
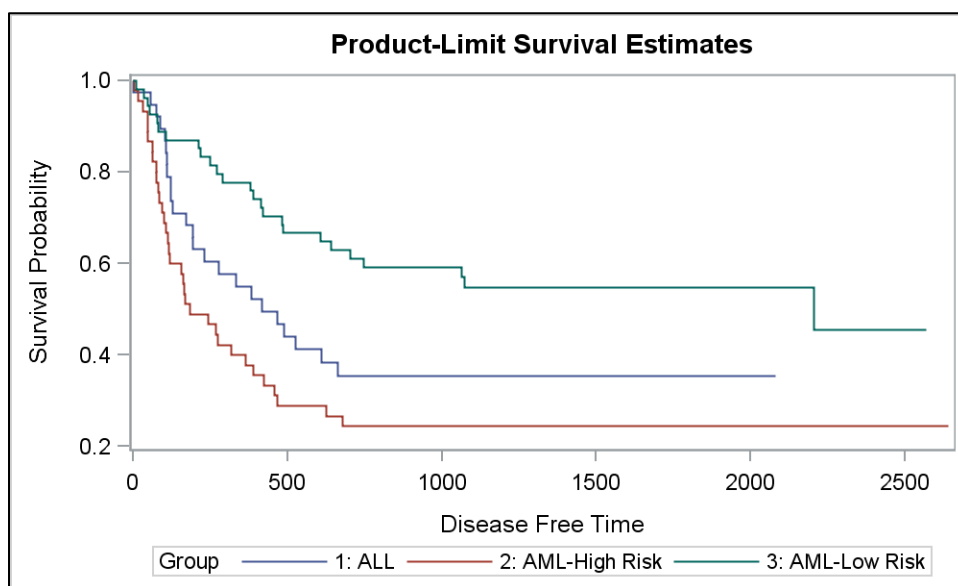**Figure 1.3**

Figure 1.4

Step2: Add the "Censored" observations.   We do this by adding the following syntax:

1. Add a SCATTER plot (**B** in Figure 1.5) statement of Censored x Time with Group=Stratum and Symbol=PLUS after the STEP statement.
2. To display the "Censored" legend in the plot, add the KEYLEGEND statement positioned inside the data area.
3. Note, if we add the grouped scatter plot (**B**) to the KEYLEGEND, the legend will show each group in the scatter plot separately, like the legend at the bottom. We do not want this.
4. So, we add another SCATTER plot (**A**) statement (without groups) before the grouped scatter plot (**B**). This statement has Name="censored" and will be used in the KEYLEGEND to create the legend as shown in Figure 1.6. The markers for scatter plot (**A**) will be overdrawn by the markers of the grouped scatter plot (**B**).
5. To get the "Censor" legend, we added an explicit KEYLEGEND statement. This action disabled the automatically generated legend for the survival curves. So, to get it back, we now add an explicit KEYLEGEND statement with the step plot (name="survival") as the associated plot.

```
title 'Product-Limit Survival Estimates';
proc sgplot data=sgf2011.SurvivalPlot49_2_1;
  step x=time y=survival / group=stratum name='survival';
  scatter x=time y=censored / markerattrs=(symbol=plus) name='censored';
  scatter x=time y=censored / group=stratum markerattrs=(symbol=plus);
  keylegend 'survival';
  keylegend 'censored' / location=inside position=topright;
  run;
```
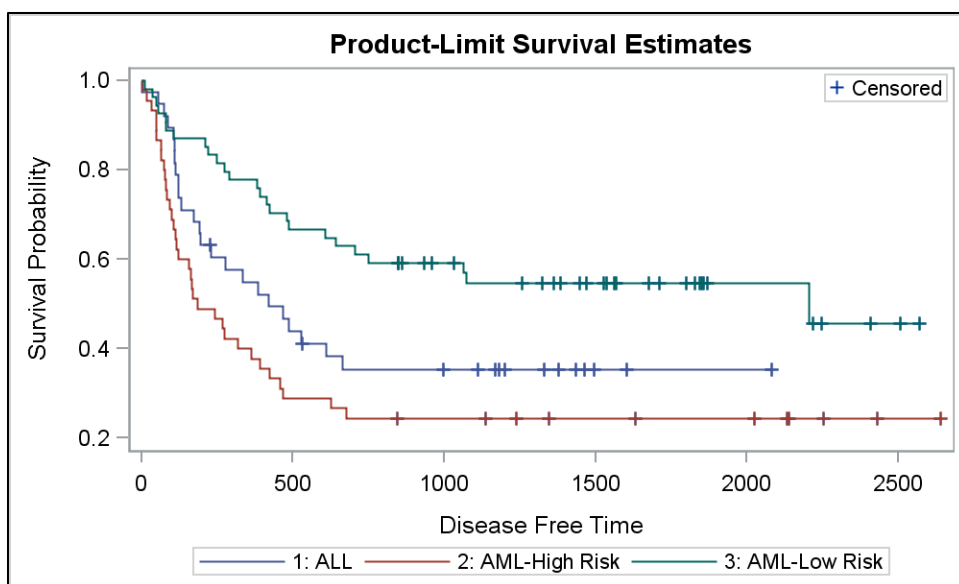
A →
B →

Figure 1.5

Figure 1.6

Step3: Add the "Number of Subjects at Risk" table below the graph as follows:

1. Create some space between the lower X axis and the lowermost survival curve to place the "At Risk" numbers. We do this by setting YAXIS OFFSETMIN=0.2. This reserves 20% of the space at the bottom of the graph.
2. Use a SCATTER plot X=tAtRisk, Y=StratumNum, and MARKERCHAR=AtRisk
3. This scatter plot will display the AtRisk numbers at (tAtRisk, StratumNum) coordinates for each observation. We associate this plot with the Y2 axis so its data range will not be merged with the "Survival" data on the Y axis.

3

4. tAtRisk is missing for all values of time except 0, 500, 1000, and so on. The AtRisk values are displayed only at these intervals.
5. To ensure that the plotted data uses only the space reserved for it below the survival curves, we set the Y2Axis OFFSETMAX=0.85. This forces the AtRisk data to be displayed in the lower 15% space of the graph.
6. To provide an easy association between each curve and the associated at risk values, we have used Group=StratumNum for the AtRisk scatter plot.

```
title 'Product-Limit Survival Estimates';
title2 h=7pt 'With Number of Subjects at Risk';
proc sgplot data=sgf2011.SurvivalPlot49_2_1;
   step x=time y=survival / group=stratum name='survival';
   scatter x=time y=censored / markerattrs=(symbol=plus) name='censored';
   scatter x=time y=censored / group=stratum markerattrs=(symbol=plus);
   scatter x=tatrisk y=stratumnum / markerchar=atrisk y2axis group=stratumnum;
   keylegend 'survival';
   keylegend 'censored' / location=inside position=topright;
   yaxis offsetmin=0.2 min=0;
   y2axis offsetmax=0.85 display=none min=1 max=3;
   run;
```
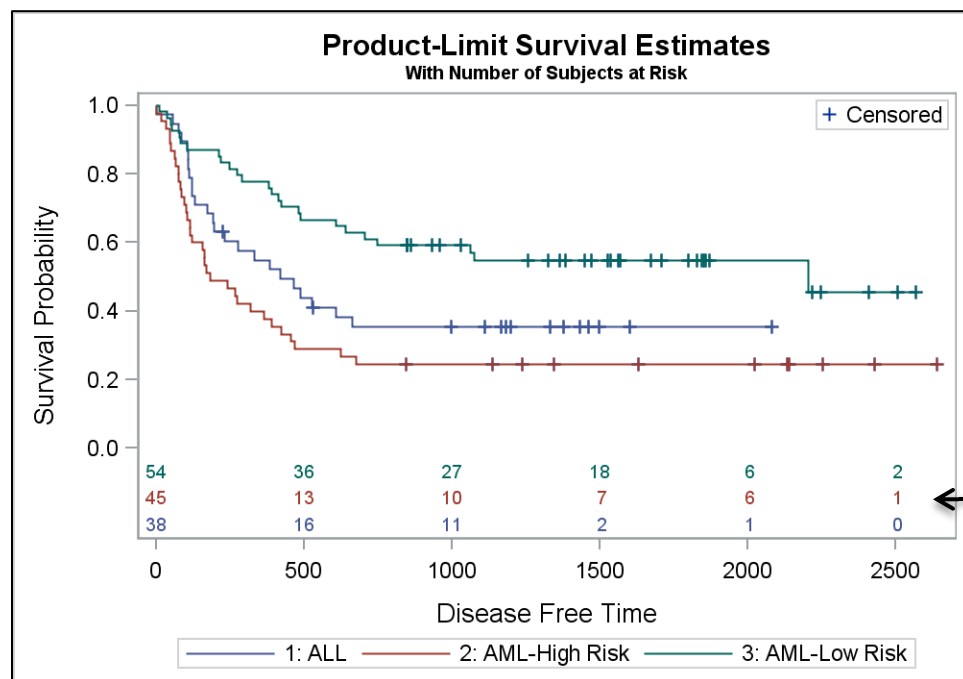
Figure 1.7

Figure 1.8

Survival Plot Summary: We used the following techniques for this plot:

1. Use multiple scatter plots, one for the censored values and one for the censored legend.
2. Use the Y and Y2 axes to split a single cell graph into two distinct regions.
3. Use OFFSETMIN and OFFSETMAX on each Y axis to reserve the regions.
4. Use a scatter plot with the MARKERCHAR= option to place data values in the graph.
5. Use color to create association between parts of the graph, making the graph easier to decode.
6. No annotation is required.

4

2. MAXIMUM LFT VALUES BY TREATMENT – Multi-Response Data.

A plot of the Liver Function Test values by Treatment is commonly used in the Health Care and Life Sciences community. The distribution of the values is plotted using a box plot for each different treatment. For this case, the data are often in the form as shown in Figure 2.1.

The data has separate columns for Drug A and B. In this case, two box plots, one for each drug, can be overlaid. However, both box plots are overlaid on the midpoint. With SAS 9.2M3, the DISCRETEOFFSET option was added to GTL for such a use case. Let us see how to do this plot using SAS 9.2M3.

Obs	Drug A	Drug B	Test
1	1.05198	0.97755	ALAT
2	0.78177	0.59554	ASAT
3	0.20475	0.20589	ALKPH
4	0.12868	0.10760	BILTOT
5	1.00211	1.19132	ALAT

Figure 2.1

Step 1. Given the data set above, we use GTL to create a box plot for Drug A and Drug B by Test as shown in Figure 2.2. Note, only the relevant GTL code is shown in the example below. The full detailed program can be obtained at the support.sas.com Web site. The resulting graph is shown in Figure 2.3.

```
proc template;
  define statgraph Max_LFT_By_Trt_1;
    begingraph;
      entrytitle 'Distribution of Maximum Liver Function Test Values by Treatment';
      layout overlay / yaxisopts=(linearopts=(viewmin=0 viewmax=4)
                       label='Maximum (/ULN)')
                       xaxisopts=(display=(line ticks tickvalues)) cycleattrs=true;
        boxplot x=Test y=A / name='a' legendlabel='Drug A (N=209)';
        boxplot x=Test y=B / name='b' legendlabel='Drug B (N=405)';
        discretelegend 'a' 'b' / location=inside valign=top halign=right across=1;
      endlayout;
      entryfootnote halign=left "For ALAT, ASAT and ALKPH, the Clinical  ...;";
      entryfootnote halign=left "For BILTOT, the CCL is 1.5 ULN: where ULN ...";
    endgraph;
  end;  run;

ods graphics / reset border=off width=5in height=3in imagename="Max_LFT_by_TRT_1";
proc sgrender data=LFT template=Max_LFT_By_Trt_1;  run;
```
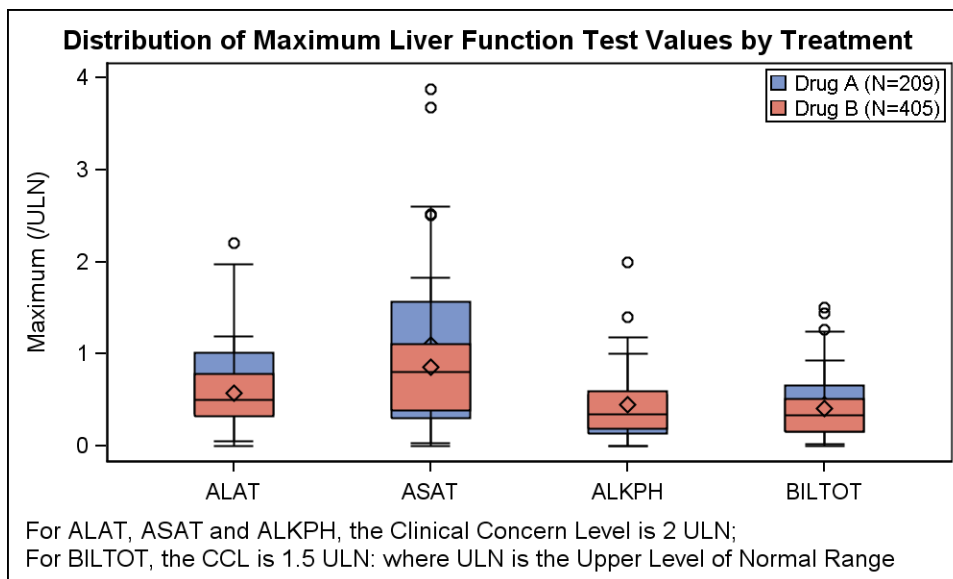
Figure 2.2

Figure 2.3

Step 2. In Figure 2.3, the two box plots for Drug A and Drug B are overlaid on the midpoint for each category. To separate the two box plots, we can use the new DISCRETOFFSET option. This option is available for all plots that display data on a discrete axis. It offsets the graph by a fraction of the midpoint spacing. This is shown in Figure 2.4.

```
proc template;
  define statgraph Max_LFT_By_Trt_2;
    begingraph;
      entrytitle 'Distribution of Maximum Liver Function Test Values by Treatment';
      layout overlay / yaxisopts=(linearopts=(viewmin=0 viewmax=4)
                       label='Maximum (/ULN)')
                       xaxisopts=(display=(line ticks tickvalues)) cycleattrs=true;
        boxplot x=Test y=A / discreteoffset=-0.2 boxwidth=.2
                name='a' legendlabel='Drug A (N=209)';
        boxplot x=Test y=B / discreteoffset= 0.2 boxwidth=.2
                name='b' legendlabel='Drug B (N=405)';
        referenceline y=1 / lineattrs=(pattern=dot);
        discretelegend 'a' 'b' / location=inside valign=top halign=right across=1;
      endlayout;
      entryfootnote halign=left "For ALAT, ASAT and ALKPH, the Clinical  ...;";
      entryfootnote halign=left "For BILTOT, the CCL is 1.5 ULN: where ULN ...";
    endgraph;
  end;  run;

ods graphics / reset border=off width=5in height=3in imagename="Max_LFT_by_TRT_2";
proc sgrender data=LFT template=Max_LFT_By_Trt_2;  run;
```

Figure 2.4

The resulting graph is shown in Figure 2.5. Note the following features of the graph:
- The Box Plot for Drug A is now offset to the left of the midpoint by 20%
- The Box Plot for Drug B is now offset to the right of the midpoint by 20%.
- The size of each Box Plot is now 20% of the midpoint spacing.
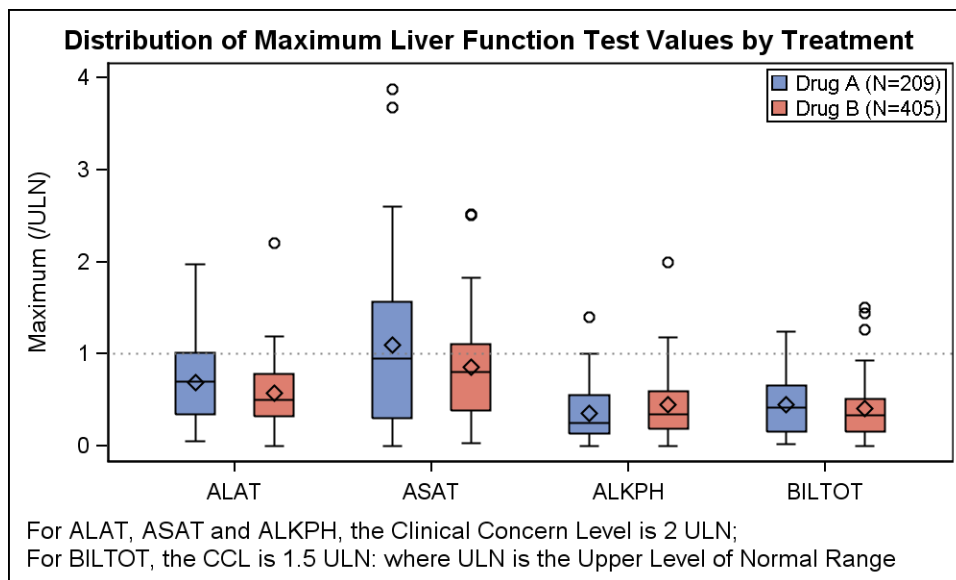- A Reference Line is added on the Y axis to show the ULN (Upper Level of Normal Range).

Figure 2.5

Step 3. The Clinical Concern Level for BILTOT (1.5) is different from the other tests (2.0).

- To display this, we will use two separate drop lines as shown in Figures 2.6 and Figure 2.7.
- The (X, Y) points for the drop lines are set using the axis values.
- For a discrete X axis, the point value is the tick value of the midpoint.
- Discrete offsets are used to move the position along the discrete axis.
- The box plot for Drug B is assigned to the Y2 (right) axis.
- The drop line for BILTOT is dropped to the Y2 axis.
- Y2 axis ranges are set to match the Y axis.
- Some box plot display options are set (not shown) and represented by <options>. These include various "attr" options to set the line patterns used for box, whiskers, median, and an option for empty boxes.

```
proc template;
  define statgraph Max_LFT_By_Trt_3;
    begingraph;
      entrytitle 'Distribution of Maximum Liver Function Test Values by Treatment';
      layout overlay / yaxisopts=(linearopts=(viewmin=0 viewmax=4)
                        label='Maximum (/ULN)')
                    y2axisopts=(linearopts=(viewmin=0 viewmax=4) display=(ticks))
                    xaxisopts=(display=(line ticks tickvalues)) cycleattrs=true;
        boxplot x=Test y=A / discreteoffset=-0.2 boxwidth=.2
              name='a' legendlabel='Drug A (N=209)' <options>;
        boxplot x=Test y=B / discreteoffset= 0.2 boxwidth=.2
              name='b' legendlabel='Drug B (N=405)' yaxis=y2 <options>;
        referenceline y=1 / lineattrs=(pattern=dot);
        dropline x='BILTOT' y=1.5 / dropto=Y yaxis=y2 discreteoffset=-.4;
        dropline x='ALKPH' y=2.0 / dropto=Y yaxis=y discreteoffset=.4;
        discretelegend 'a' 'b' / location=inside valign=top halign=right across=1;
      endlayout;
      entryfootnote halign=left "For ALAT, ASAT and ALKPH, the Clinical  ...;";
      entryfootnote halign=left "For BILTOT, the CCL is 1.5 ULN: where ULN ...";
    endgraph;
  end;  run;

ods graphics / reset border=off width=5in height=3in imagename="Max_LFT_by_TRT_2";
proc sgrender data=LFT template=Max_LFT_By_Trt_2;  run;
```
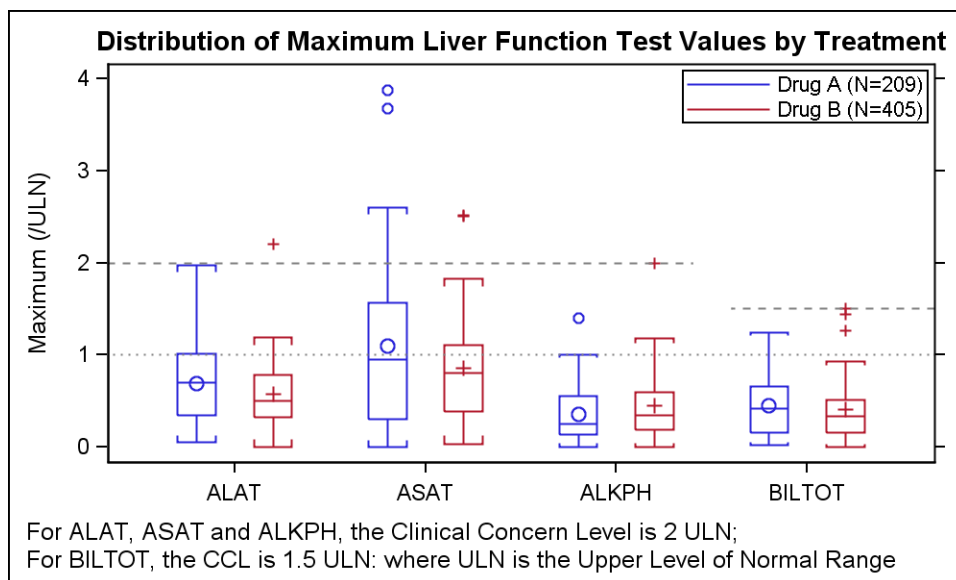
Figure 2.6

Figure 2.7

3. MAXIMUM LFT VALUES BY TREATMENT – Using SAS 9.3 SGPlot Procedure

For SAS 9.3, support for DISCRETEOFFSET has been added to SG Procedures for various plots with discrete data. In addition, support has been added to display grouped box plots in both GTL and SG Procedures.

Obs	Test	Drug	Value
1	ALAT	A	1.05198
2	ALAT	B	0.97755
3	ASAT	A	0.78177
4	ASAT	B	0.59554

Figure 3.1

Figure 3.1 shows the same data using a column to identify the drug. For this use case, a grouped box plot with GROUPDISPLAY=CLUSTER can be used to create the same graph as shown in Figure 3.2.

```
title h=10pt 'Distribution of Maximum Liver Function Test Values by Treatment';
footnote1 h=8pt j=left "For ALAT, ASAT and ALKPH, the Clinical Concern...";
footnote2 h=8pt j=left "For BILTOT, the CCL is 1.5 ULN: where ULN ...";
proc sgplot data=LFT_Group;
   format drug $drug.;
   vbox value / category=test group=drug lineattrs=(pattern=solid)
       medianattrs=(pattern=solid) whiskerattrs=(pattern=solid) nofill;
   keylegend / location=inside position=topright across=1;
   refline 1 1.5 2 / lineattrs=(pattern=dot);
   xaxis discreteorder=data display=(nolabel);
   yaxis label='Maximum (/ULN)';
   run;
```

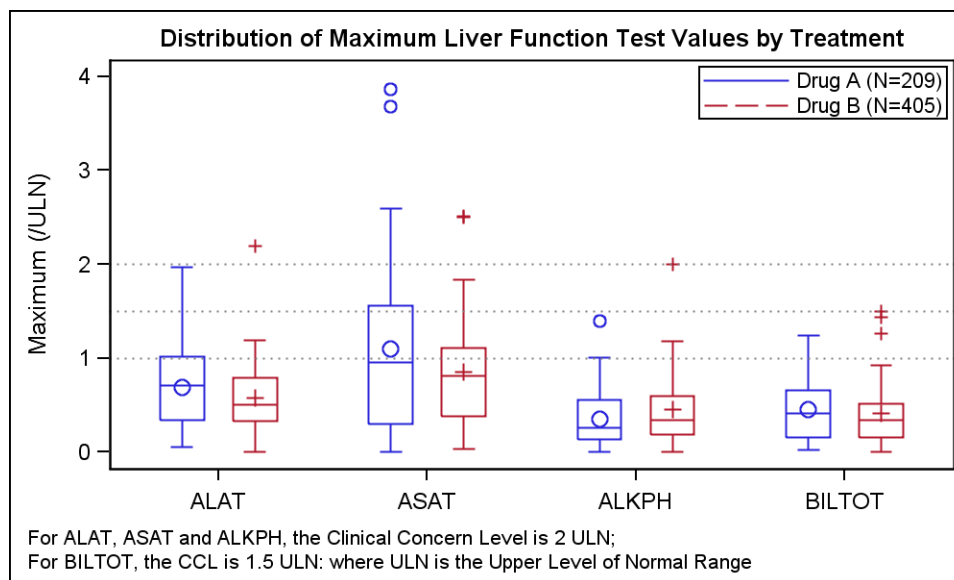**Figure 3.2 (SAS 9.3)**

Figure 3.3 (SAS 9.3)

As can be seen from the program shown in Figure 3.2, this graph can be done using the SGPLOT procedure with only a few lines of code. The graph is almost the same as that in Figure 2.7.

Note: The SG Procedures do not support the DROPLINE statement. Hence, we used three regular Ref Lines to display the Clinical Concern Level values of 1, 1.5 and 2.0. If the drop line feature is important, a grouped box plot graph can be created using GTL, where the DROPLINE statement is available.

4. Forest Plot

A forest plot is a graphical display of the relative strength of treatment effects in multiple quantitative scientific studies addressing the same question (Wikipedia). An example of one common display of a Forest Plot is included in the list of Health Care and Life Sciences graph samples in the support.sas.com Web site.

One key graphical aspect of the forest plot is the graphical display of the odds ratio and confidence limits reported by each study, along with the display of the statistics themselves as an aligned stat table. The example shown in the sample mentioned above is a sophisticated multi-cell graph created using GTL. Here we will explore how you can create such a graph using the SGPLOT procedure.

The primary motivation for the SGPLOT procedure is the easy creation of sophisticated single-cell graphs. As we have seen for the survival plot, the data region for a single cell can be effectively split using the Y and Y2 axes. Here we explore how to split the region using the X and X2 axis. Also, we once again exploit the power of the scatter plot to display various statistics in the graph that are axis-aligned.

The data for the forest plot includes columns for the Study Name, Odds Ratio, Upper and Lower CL values, and Weight. The other columns in this data set are computed to facilitate display of the stat table in the graph.

- The study names are separated into two columns, one for the individual studies, and one for overall.
- UCL2, LCL2 – used to draw the upper and lower limits for individual studies (but not for overall).
- OR, LCL, UCL & WT – used to provide the category midpoints for the stat tables.
- For the "Overall" observation, the Study name is moved to a different column (Study2).

Study	OddsRatio	LowerCL	UpperCL	Weight	Q1	Q3	study2	ObsId	lcl2	ucl2	OR	LCL	UCL	WT
Modano (1967)	0.590	0.096	3.634	5%	0.56	0.62		1	0.096	3.634	OR	LCL	UCL	Weight
Borodan (1981)	0.464	0.201	1.074	18%	0.38	0.55		2	0.201	1.074	OR	LCL	UCL	Weight
Leighton (1972)	0.394	0.076	2.055	10%	0.35	0.43		3	0.076	2.055	OR	LCL	UCL	Weight
Novak (1992)	0.490	0.088	2.737	10%	0.44	0.54		4	0.088	2.737	OR	LCL	UCL	Weight
Stawer (1998)	1.250	0.479	3.261	15%	1.06	1.44		5	0.479	3.261	OR	LCL	UCL	Weight
Truark (2002)	0.129	0.027	0.605	13%	0.11	0.15		6	0.027	0.605	OR	LCL	UCL	Weight
Fayney (2005)	0.313	0.054	1.805	10%	0.28	0.34		7	0.054	1.805	OR	LCL	UCL	Weight
Modano (1969)	0.429	0.070	2.620	10%	0.39	0.47		8	0.070	2.620	OR	LCL	UCL	Weight
Soloway (2000)	0.718	0.237	2.179	15%	0.61	0.83		9	0.237	2.179	OR	LCL	UCL	Weight
Adams (1999)	0.143	0.082	0.250	20%	0.11	0.17		10	0.082	0.250	OR	LCL	UCL	Weight
Truark2 (2002)	0.129	0.027	0.605	13%	0.11	0.15		11	0.027	0.605	OR	LCL	UCL	Weight
Fayney2 (2005)	0.313	0.054	1.805	10%	0.28	0.34		12	0.054	1.805	OR	LCL	UCL	Weight
Modano2 (1969)	0.429	0.070	2.620	10%	0.39	0.47		13	0.070	2.620	OR	LCL	UCL	Weight
Soloway2(2000)	0.718	0.237	2.179	15%	0.61	0.83		14	0.237	2.179	OR	LCL	UCL	Weight
Adams2 (1999)	0.143	0.082	0.250	20%	0.11	0.17		15	0.082	0.250	OR	LCL	UCL	Weight
	0.328	0.233	0.462	.	.	.	Overall	16	.	.	OR	LCL	UCL	Weight

Figure 4.1

Step 1. Create a scatter plot of the Odds Ratio by the Study Name, with the lower and upper confidence limits.
- Here we use the SGPLOT procedure as shown in Figure 4.2.
- Use a scatter plot of OddsRatio x Study Name, with upper and lower CL for the individual studies.
- Use a scatter plot of OddsRatio x Study2 to plot the "overall" observation.
- Set the X axis type to Log with base=10. Set axis ranges from 0.01 to 100, with minor ticks.
- Restrict the display of the odds Ratio plot to the lower 70% of the axis by setting OFFSETMAX=0.3.
- Turn off the Y axis label.

```
title "Impact of Treatment on Mortality by Study";
title2 h=8pt 'Odds Ratio and 95% CL';
proc sgplot data=forest3 noautolegend;
  scatter y=study x=oddsratio / group=grp
          xerrorupper=ucl2 xerrorlower=lcl2;
  xaxis type=log offsetmin=0 offsetmax=0.3 min=0.01 max=100
        minor display=(nolabel);
run;
```
Figure 4.2

The resulting graph for this program is shown in Figure 4.3. Note the following features so far:
- The Odds Ratio and CL are plotted to the X axis, which is restricted to the leftmost 70%.
- We have used different marker attributes for these two plots.
- This leaves an empty space on the right side, which we will use to display the statistics.
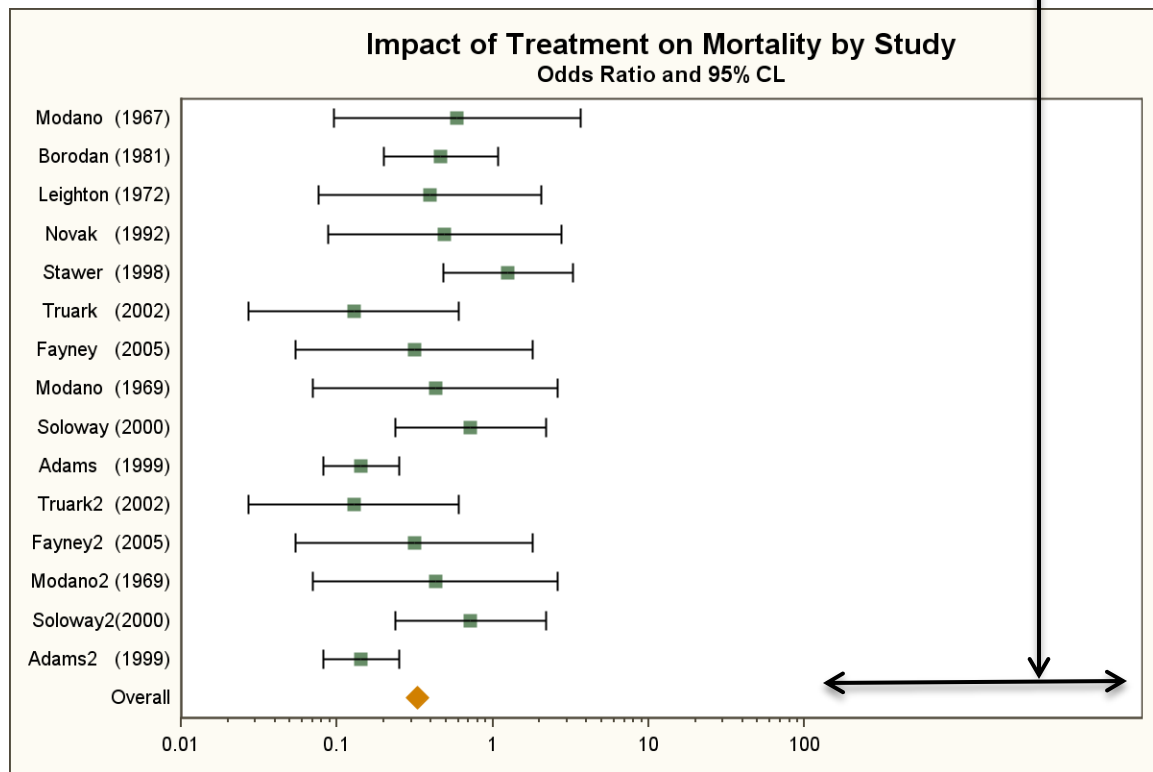

Figure 4.3

Step 2 – Add stat tables and reference lines.
- Note, 30% of the X axis space is left unused because we set the X axis OFFSETMAX=0.3.
- We will draw the individual statistics in this space, aligned with the Y axis, but using the X2 axis.
- We use one scatter plot to draw each of the statistics using the MARKERCHAR option.
- The axis automatically gathers the discrete values from all the plots that are assigned to it.
- We restrict the X2 axis to use the right side 30% of the space to avoid overlap with the graph.
- Add Reference lines for X=1 and 100, with a solid pattern.
- Add Reference lines for X=0.01, 0.1, and 10, with a dash pattern.
- Note that we reserve some space near bottom of Y axis for some annotation.
- OffsetMin and OffsetMax are set to macro variables values computed in the program (not shown).

10

```
title "Impact of Treatment on Mortality by Study";
title2 h=8pt 'Odds Ratio and 95% CL';
proc sgplot data=forest noautolegend;
   scatter y=study2 x=oddsratio / markerattrs=(symbol=diamondfilled size=10);
   scatter y=study x=oddsratio / xerrorupper=uppercl xerrorlower=lowercl
           markerattrs=(symbol=squarefilled);
   scatter y=study x=or / markerchar=oddsratio x2axis;
   scatter y=study x=lcl / markerchar=lowercl x2axis;
   scatter y=study x=ucl / markerchar=uppercl x2axis;
   scatter y=study x=wt / markerchar=weight x2axis;
   refline 1 100  / axis=x;
   refline 0.01 0.1 10 / axis=x lineattrs=(pattern=shortdash)
           transparency=0.5;
   inset '                Favors Treatment'  / position=bottomleft;
   inset 'Favors Placebo'  / position=bottom;
   xaxis type=log offsetmin=0 offsetmax=0.35 min=0.01 max=100 minor
         display=(nolabel) ;
   x2axis offsetmin=0.7 display=(noticks nolabel);
   yaxis display=(noticks nolabel) offsetmin=&pct2 offsetmax=&pct;
run;
```
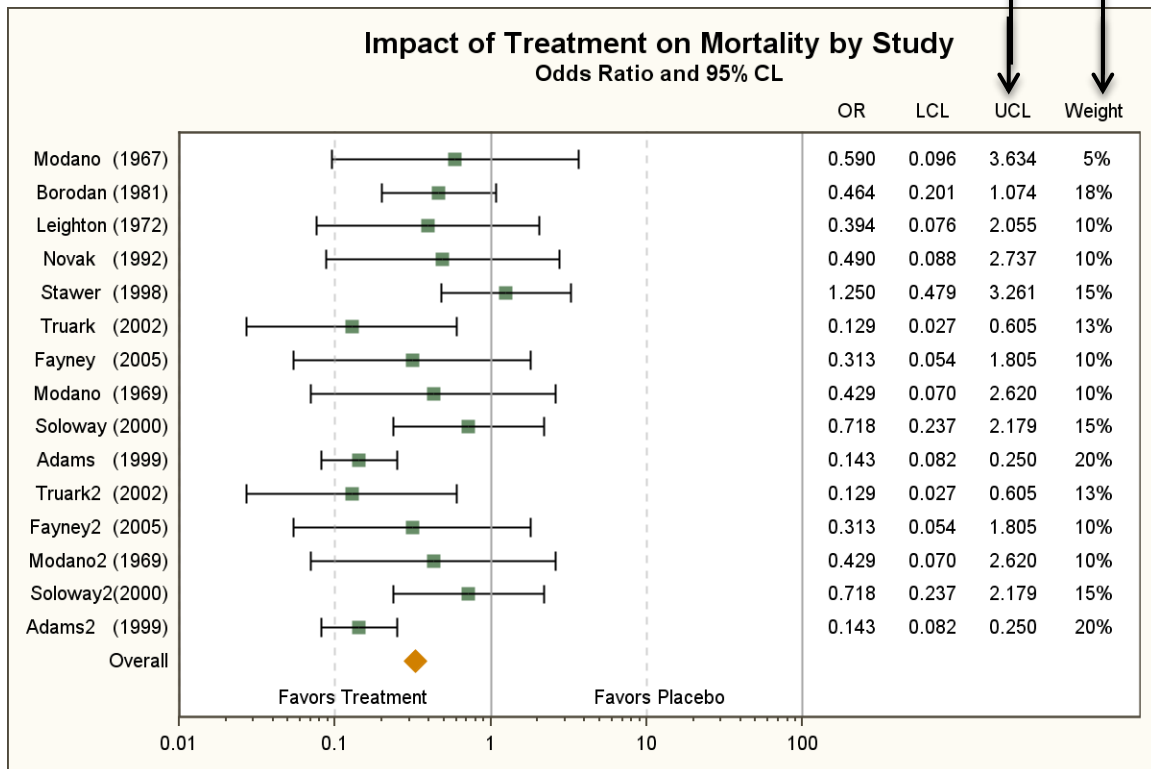
Figure 4.4

Figure 4.5

As we can see from the completed example shown in Figures 4.5, the X-X2 split technique is very useful to create a 2-cell graph using the SGPLOT procedure. This relatively sophisticated plot can be created with under 20 lines of code. No annotation is required.

5. Adverse Event Timeline

This graph is frequently used to track individual patients in a study. The data is in CDISC format, and the data shown in Figure 5.1 is extracted for one patient.

- Columns include the start and end dates for each adverse event, the name of the event, and its severity.
- Some additional columns are computed to aid in the creation of the graph.
- Three dummy observations have been added, one with each severity type for the legend.
- If an event does not have a start or end date, the earliest or latest date for the study is substituted.
- Markers are drawn at each end to indicate the start and end date of the event.

Obs	aestdate	aeendate	aeseq	aedecod	aesev	y	xc	aestdy	aeendy	startday	endday
1	06MAR13	06MAR13	1	DUMMY	MILD	-100	-1.4	3	3	3	3
2	06MAR13	06MAR13	1	DUMMY	MODERATE	-100	-1.4	3	3	3	3
3	06MAR13	06MAR13	1	DUMMY	SEVERE	-100	-1.4	3	3	3	3
7	27MAR13	27MAR13	4	DIZZINESS	MILD	4	17.4	24	24	24	24
14	27MAY13	18JUN13	11	PRURITUS	MODERATE	11	78.9	85	107	85	107

Figure 5.1

Step 1 – We use the SGPLOT procedure to create this graph, using the VECTOR plot to draw the events.
- A vector plot (without arrow heads) is used to draw each event.
- Two overlaid vector plots are used for aesthetic reasons only – to provide a border for each event.
- Both vector plots use Study Day as the X variable.
- Y axis Min =0, so the dummy observations are not shown.
- X axis extents are set to start from Day=-10, using the macro variable "minday10".

```
title "Adverse Events for Patient Id = xx-xxx-xxxx";
proc sgplot data=ae1 noautolegend nocycleattrs;
   refline 0 / axis=x lineattrs=(thickness=1 color=black);
   vector x=endday y=y / xorigin=startday yorigin=y noarrowheads group=aesev
          lineattrs=(thickness=9px pattern=solid color=black);
   vector x=endday y=y / xorigin=startday yorigin=y noarrowheads
          lineattrs=(thickness=7px pattern=solid) group=aesev  name='sev';
   yaxis display=(nolabel noticks novalues) min=0;
   xaxis grid label='Study Days' offsetmin=0.03 offsetmax=0.03
          values=(&minday10 to &maxday by 2);
   keylegend 'sev'/ title='Severity :';
run;
```
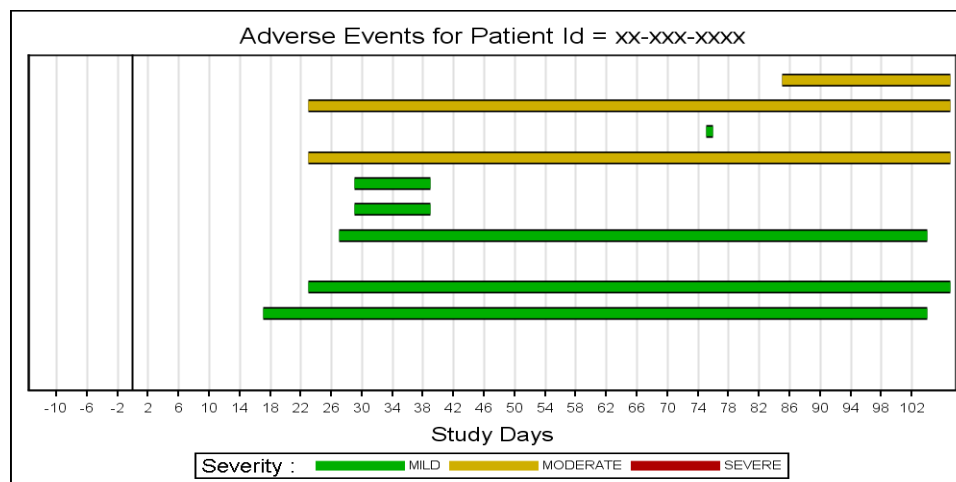
Figure 5.2

Figure 5.3

The graph shown in Figure 5.3 has the following features:
- A bar is drawn for each event.
- The color of the bar represents the severity of the event.
- The legend includes all possible severities, even if this data has only Mild and Moderate.
- The X axis starts at day -10, with a reference line to indicate the start day of the study.

Step 2 – Add Event markers and Date axis.
- Add scatter plots to display a marker at each end of the event (if it has a date).
- Multiple scatter plots are used for aesthetic reasons to display a border for each marker.
- Another scatter plot statement is used with association to the X2 axis using Date values.
- Use a scatter plot with the MarkerChar option to display the adverse event names for each event.
- Location for each name is computed based on the length of the event name (not shown).
- X2 axis extents are set to correctly match the X axis.
- The program is shown in Figure 5.4. The resulting graph is shown in Figure 5.5.

```
title "Adverse Events for Patient Id = xx-xxx-xxxx";
proc sgplot data=ae1 noautolegend nocycleattrs;
   refline 0 / axis=x lineattrs=(thickness=1 color=black);
   vector x=endday y=y / xorigin=startday yorigin=y noarrowheads
           lineattrs=(thickness=9px pattern=solid);
   vector x=endday y=y / xorigin=startday yorigin=y noarrowheads
           lineattrs=(thickness=7px pattern=solid) group=aesev  name='sev';
   scatter x=aestdy y=y / markerattrs=graphdatadefault(options);
   scatter x=aestdy y=y / markerattrs=(options)group=aesev;
   scatter x=aeendy y=y / markerattrs=graphdatadefault(options);
   scatter x=aeendy y=y / markerattrs=(options)group=aesev;
   scatter x=xc y=y / markerchar=aedecod;
   scatter x=aestdate y=y /  markerattrs=(size=0) x2axis;
   yaxis display=(nolabel noticks novalues) min=0;
   xaxis grid label='Study Days' offsetmin=0.03 offsetmax=0.03
         values=(&minday10 to &maxday by 2);
   x2axis notimesplit display=(nolabel) offsetmin=0.03 offsetmax=0.03
         values=(&mindate10 to &maxdate) tickvalueformat=date7.;
   keylegend 'sev'/ title='Severity :'; run;
```
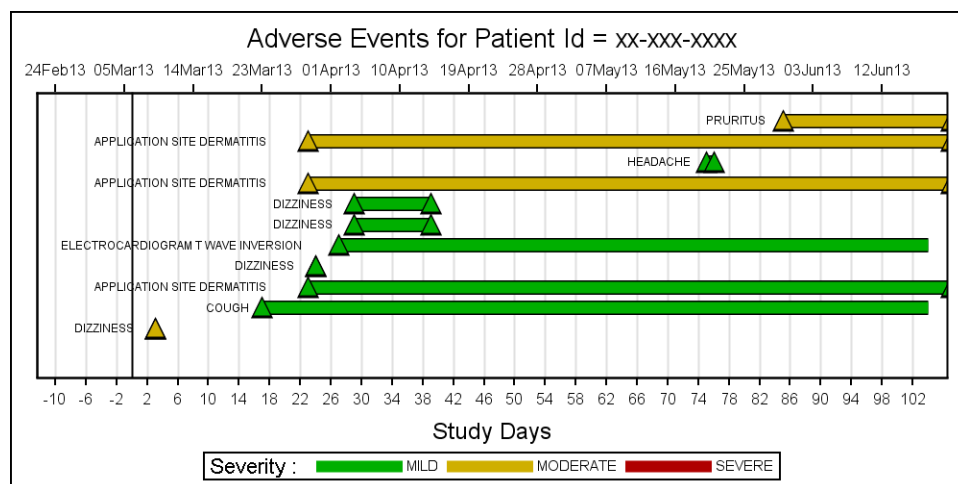**Figure 5.4**

Figure 5.5

13

6. Adverse Event Plot using SAS 9.3 features

SAS 9.3 provides new plot types that make it easier to create adverse event plots. The key new feature used in this graph is the HighLow Plot, as shown in the program in Figure 6.2:

- This plot allows placement of the High and/or Low labels at the end of the bar.
- This plot also allows drawing arrow heads to indicate continuing events.
- The column "xs' is used for the band plot that creates the legend for the dummy observations.
- Data for the graph is as shown in Figure 6.1.
- The resulting graph is shown in Figure 6.3.

aestdate	aeendate	aeseq	aedecod	aesev	legend	y	aestdy	aeendy	startday	endday	lowcap	highcap	xs
06MAR13	06MAR13	.		MILD	Legend	-100	3	3	3	3			0
06MAR13	06MAR13	.		MODERATE	Legend	-100	3	3	3	3			0
06MAR13	06MAR13	.		SEVERE	Legend	-100	3	3	3	3			0
06MAR13	06MAR13	1	DIZZINESS	MODERATE		1	3	3	3	3			0

Figure 6.1

```
title "Adverse Events for Patient Id = xx-xxx-xxxx";
proc sgplot data=ae1 noautolegend nocycleattrs;
  highlow y=y low=xs high=xs / group=aesev type=bar barwidth=0.8
          name='sev' Y2axis;
  highlow y=aeseq low=startday high=endday / group=aesev
          lowlabel=aedecod type=bar barwidth=0.8
          lineattrs=(pattern=solid color=black thickness=1)
          labelattrs=(size=6) lowcap=lowcap highcap=highcap;
  scatter x=aestdate y=aeseq /  markerattrs=(size=0) x2axis;
  refline 0 / axis=x lineattrs=(thickness=1 color=black);
  yaxis display=(nolabel noticks novalues) type=discrete min=0;
  y2axis display=none min=0;
  xaxis grid label='Study Days'  offsetmin=0.02 offsetmax=0.03
        values=(&minday10 to &maxday by 2);
  x2axis notimesplit display=(nolabel) tickvalueformat=date7.
        offsetmin=0.02 offsetmax=0.03 values=(&mindate10 to &maxdate);
  keylegend 'sev'/ title='Severity :';   run;
```
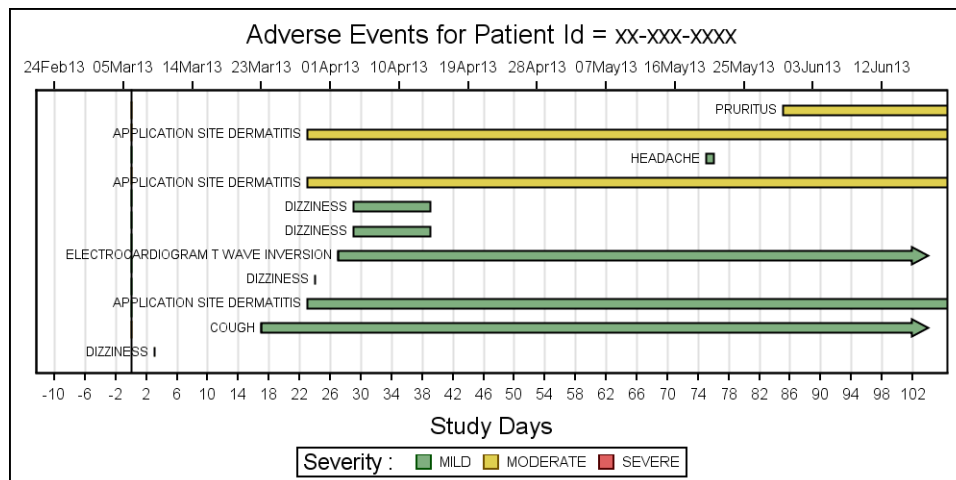
Figure 6.2

Figure 6.3

14

7. Distribution of Patients by Cause of Death and Sex

The X-X2 and Y-Y2 split techniques can be used to create a 2 x 2 graph using the SGPLOT procedure. The data is shown in Figure 7.1. The program is shown in Figure 7.2 and has the following features:

- The graph is split in the X direction using min and max offsets of 0.85 and 0.25.
- The graph is split in the Y direction using min and max offsets of 0.85 and 0.25.
- Four different graphs are populated in each "cell".
- The resulting graph is shown in Figure 7.3.

DeathCause	Sex	Weight	Systolic	SexWt	MeanWt	minWt	maxWt	DeathCauseBP	MeanBP	minBP	maxBP
Cancer	Female	194	144	Female	148.596	83	300	Cancer	138.395	90	240
Cerebral Vascular Disease	Male	129	124	Male	167.849	106	276	Cerebral Vascular Disease	154.381	96	294
Cancer	Male	122	120		.	.	.	Coronary Heart Disease	149.245	94	276
Coronary Heart Disease	Male	157	142		.	.	.		.	.	.
Cancer	Male	172	118		.	.	.		.	.	.

Figure 7.1

```
title 'Patient Profile';
proc sgplot data=heart2 nocycleattrs;
  scatter x=systolic y=weight / group=sex markerattrs=(symbol=circlefilled)
transparency=0.8 name='scat';
  reg x=systolic y=weight / cli degree=2 nomarkers;
  highlow y=deathcauseBP low=minbp high=maxbp / y2axis type=bar
          fillattrs=graphdata6 transparency=0.4;
  scatter y=deathcauseBP x=meanbp / y2axis markerattrs=(symbol=diamondfilled)
          name='bp' legendlabel="Mean";
  highlow x=sexWt low=minWt high=maxWt / x2axis group=sexwt type=bar transparency=0.4;
  scatter y=meanWt x=sexWt / x2axis markerattrs=(symbol=diamondfilled) name='wt';
  scatter x=sex y=deathcause / y2axis x2axis markerattrs=(symbol=circlefilled size=9);
  refline 'Female' / discreteoffset=-0.5 axis=x2 lineattrs=(thickness=2);
  refline 'Cancer' / discreteoffset=-0.5 axis=y2 lineattrs=(thickness=2);
  xaxis offsetmin=0.05 offsetmax=0.25 grid label='Systolic BP';
  x2axis offsetmin=0.85 offsetmax=0.05 display=(nolabel) grid valueattrs=(size=7);
  yaxis offsetmin=0.05 offsetmax=0.25 grid label='Weight';
  y2axis offsetmin=0.85 offsetmax=0.03 display=(nolabel) grid valueattrs=(size=7);
  keylegend 'scat' 'bp';   run;
```
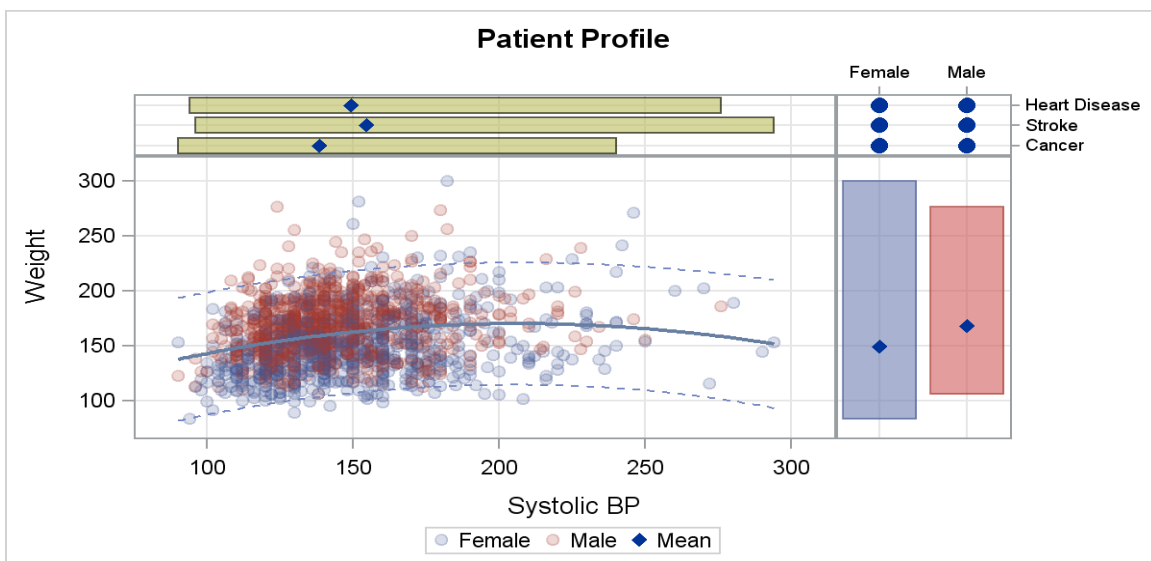
Figure 7.2

Figure 7.3

CONCLUSION

SAS SG Procedures and Graph Template Language provide an extensive and flexible syntax for the creation of graphs for all domains, including the Health Care and Life Sciences industry. These procedures use a "building-block" approach to the design of the graphs. A large set of plots and layouts makes it possible to create many graphs by combining these graph elements.

While some combinations of these statements are obvious, these elements can often be combined in creative ways to achieve results that might not be so obvious. The examples in this paper illustrate many such use cases of how the plot statements themselves can be used in place of custom annotation. Creative use of the X, Y, *X2* and *Y2* axes, along with scatter plots (with MARKERCHAR option), reference lines, and so on, can help build graphs that go beyond the most obvious use cases.

RECOMMENDED READING

Matange, Sanjay. 2010 "Clinical Graphs using ODS Graphics." *Proceedings of the 2010 Western Users of SAS Software Conference*. Cary, NC: SAS Institute Inc. Available at http://www.wuss.org/proceedings10/analy/3022_3_ANL-Matange.pdf.

Schwartz, Susan. 2009 "Clinical Trial Reporting Using SAS/GRAPH® SG Procedures." *Proceedings of the SAS Global Forum 2009 Conference*. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings09/174-2009.pdf.

Amit, Ohad, et al. "Graphical Approaches to the Analysis of Safety Data from Clinical Trials." Pharmaceutical Statistics 2008; 7: 20-35.

Heath, Dan. 2007. "New SAS/GRAPH Procedures for Creating Statistical Graphics in Data Analysis." *Proceedings of the SAS Global Forum 2007 Conference*. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/events/sasglobalforum/previous/index.html.

Cartier, Jeff. 2006. "A Programmer's Introduction to the Graphics Template Language." *Proceedings of the Thirty-first Annual SAS Users Group International Conference*. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/events/sasglobalforum/previous/index.html.

Rodriguez, R. N., and T. E. Balan. 2006. "Creating Statistical Graphics in SAS 9.2: What Every Statistical User Should Know." *Proceedings of the Thirty-first Annual SAS Users Group International Conference*. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/events/sasglobalforum/previous/index.html.

CONTACT INFORMATION

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

Sanjay Matange
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27512
(919) 531-6753
Fax (919) 677-4444
Sanjay.Matange@sas.com

---

[1] SAS/STAT User's Guide, Version 9.2, 2d ed. (Cary, NC: SAS Institute Inc., 2009)