Clinical Graphs using ODS Graphics – Analysis of Safety Data for Clinical Trials

Jan René Larsen, SAS Institute A/S, Denmark

Analysis of Safety Data for Clinical Trials

- Interpretation of Safety Data is more effective with:
  - Graphical display of data and statistics.
  - Inclusion of desirable levels and adverse events.
  - Comparison of results with other drugs or placebos.
  - Classification of results by multiple variables.
  - Display of results over time.
- 'Graphical Approaches to the Analysis of Safety Data from Clinical Trials" by Ohad Amit, Richard M. Heiberger and Peter W. Lane *in* Pharmaceutical Statistics 2008; 7: 20-35.

Response parameters (Amit *et al*)

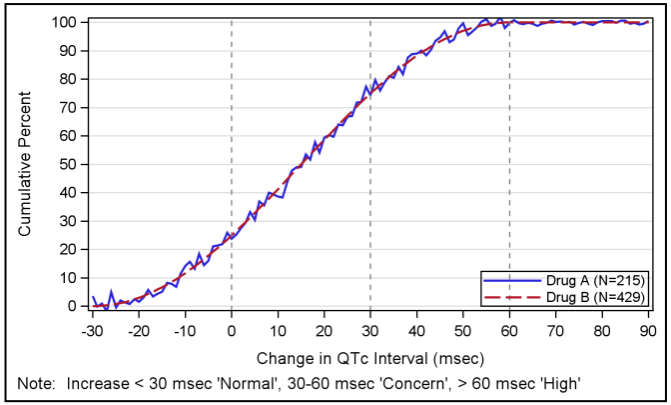
- QT interval (QTc): A measure of heartbeat rhythm derived from electrocardiogram.
- Liver Function Test (LFTs): aspartate aminotransferase (ASAT), alanine aminotransferase (ALAT), alkaline phosphatase (ALKPH) and total bilibrubin (BILTOT).
- Adverse experience data.

The Goal

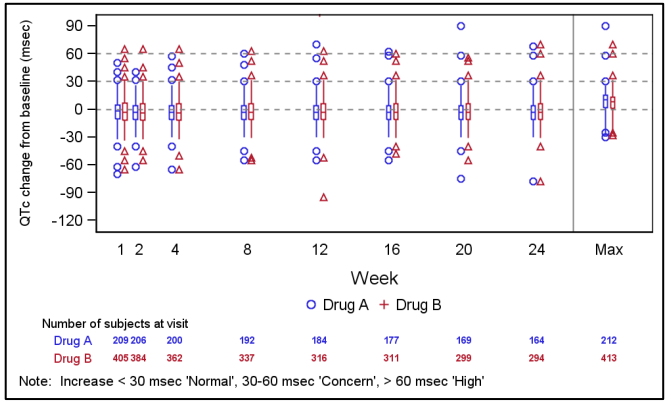
- Demonstrate the kind of graphs that can be done using ODS Graphics.
- Engage in a conversation with you in order to…
  - Understand the graphs you need to create.
  - Figure out how to build those using ODS Graphics.
  - Add new features to the software to make your task easier.

Clinical Graphs Examples - 1

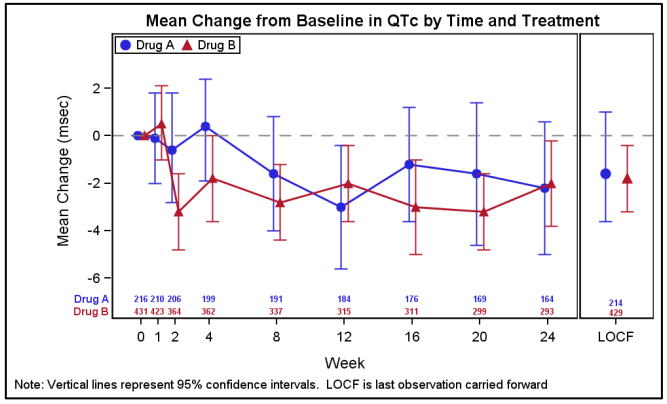
1 Empirical Distribution Function for QTc



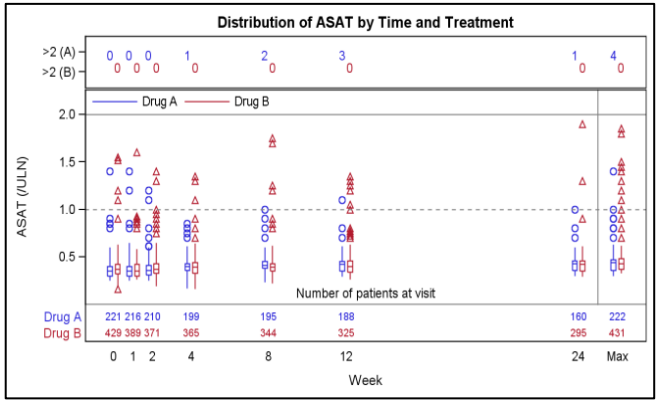
2 Boxplot of Change from Baseline QTc



3 Mean Change from Baseline for QTc

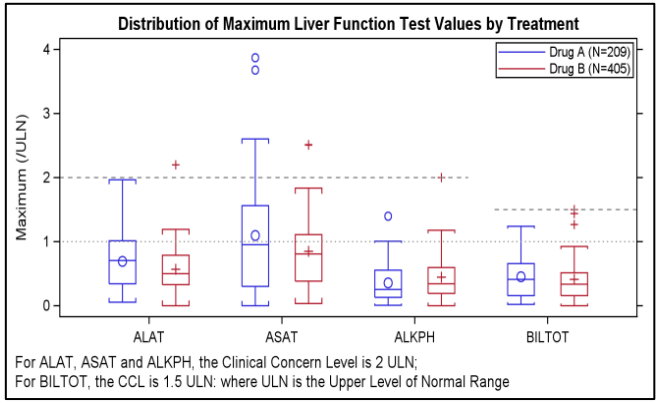


4 Distribution of ASAT

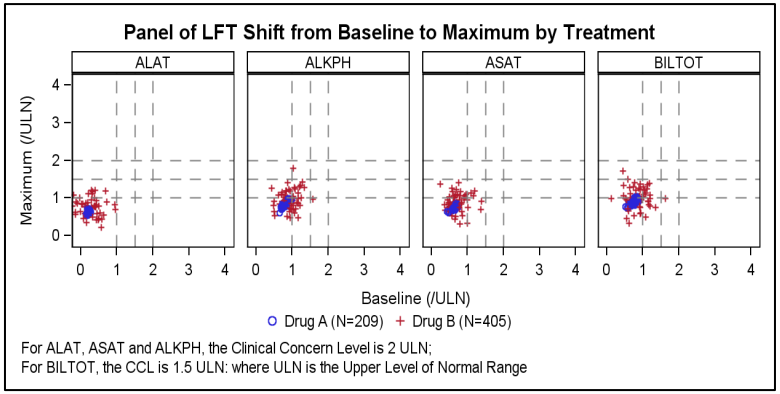


Clinical Graphs Examples - 2

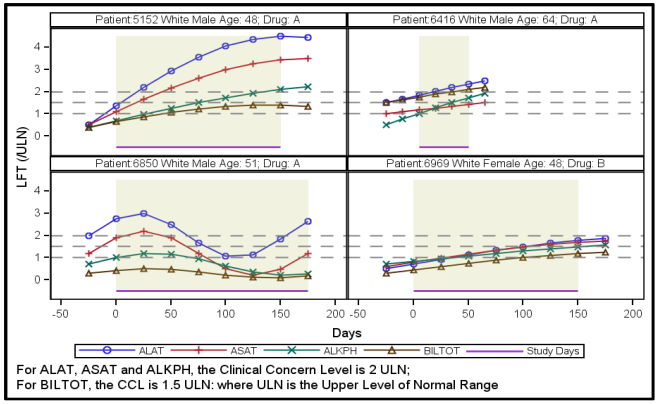
5 Distribution of Max LFT by Treatment



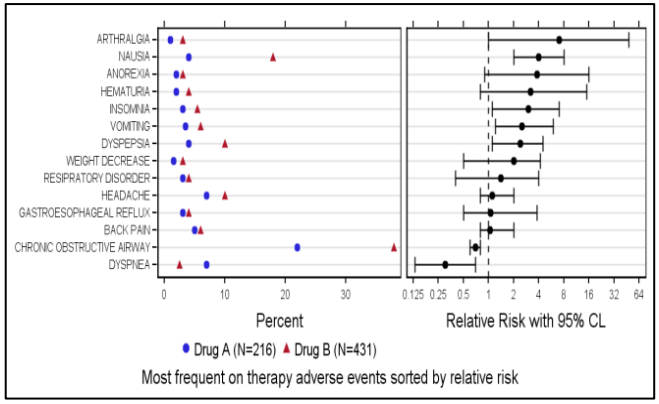
6 Panel of LFT Shift from Baseline



8 LFT Patient Profile

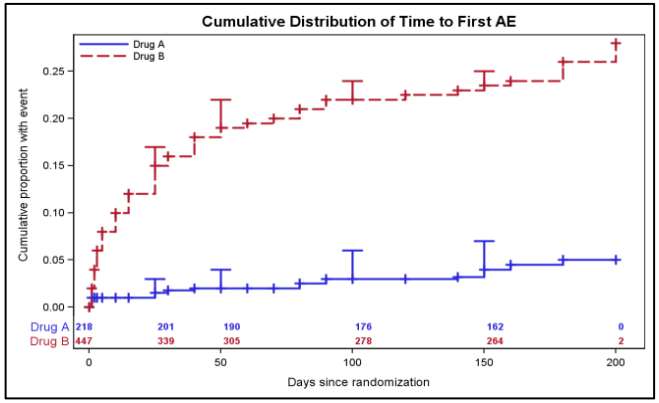


9 Most Freq On-Therapy Adverse Event

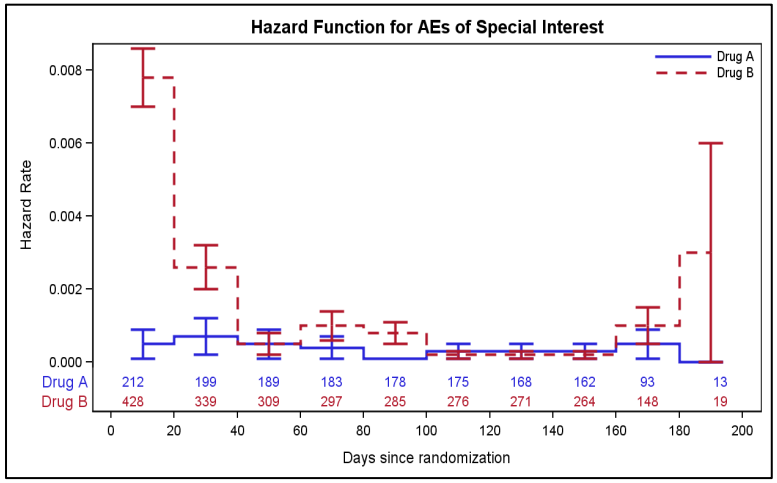


Clinical Graphs Examples - 3

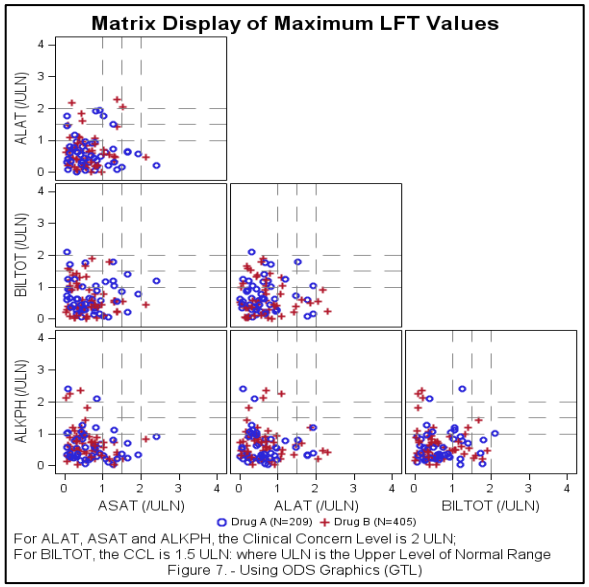
10 Cumulative Distribution of Time to First AE



11 Hazard Function for Adverse Event



7 Matrix Display of Maximum LFT Values



ODS Graphics

- All the graphs shown in the previous slides have been built using ODS Graphics software.
- So…What is ODS Graphics?

ODS Graphics

- ODS Graphics is a new system for creating statistical graphs first released with SAS 9.2(TS2M0).
- This software includes
  - Automatic Statistical Graphs from analytical procedures.
  - SAS/GRAPH® Statistical Graphics Procedures.
  - SAS/GRAPH® Graph Template Language (GTL).
  - ODS Graphics Designer – An interactive application.

ODS Graphics Audience Segmentation - 1

- Graph Consumer:
  - Uses automatic graphs created by Analytical Procedures
  - Uses ODS Graphics Editor for small cosmetic changes.

- Non-programmatic Graph Creator:
  - Uses ODS Graphics Designer.
  - Creates graphs using interactive application (no coding).
  - Runs graphs in batch for different data.

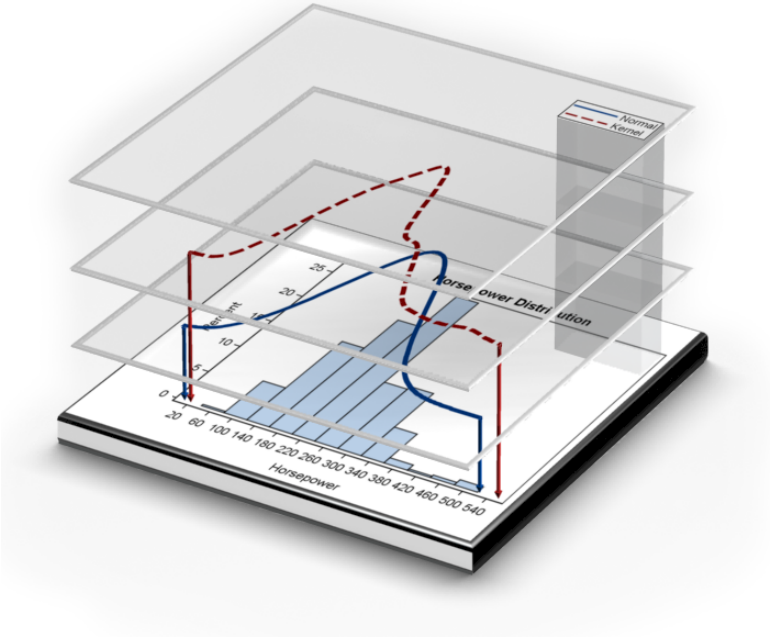
ODS Graphics Audience Segmentation - 2

- Graph Programmer:
  - Can use SAS/GRAPH SG Procedures.
  - Creates custom graphs using procedure syntax.

- Advanced Graph Programmer:
  - Can use Graph Template Language (GTL).
  - Creates complex graphs using GTL.
  - Customizes procedure templates.

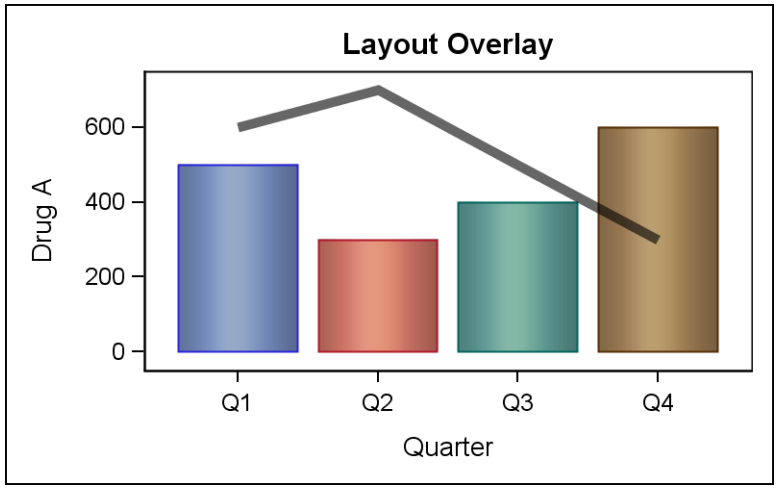
Some Background

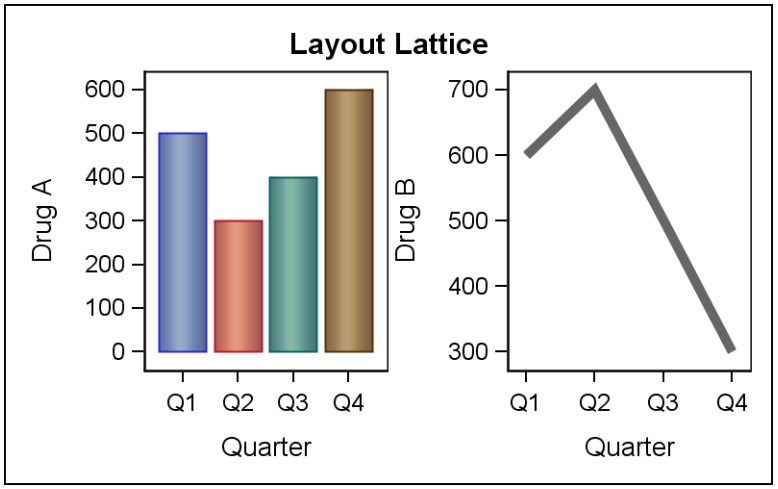
- All ODS Graphics are done using GTL.
- SAS analytical procedures and the SG procedures use GTL behind the scenes.
- GTL uses a simple “Building-Block” approach.



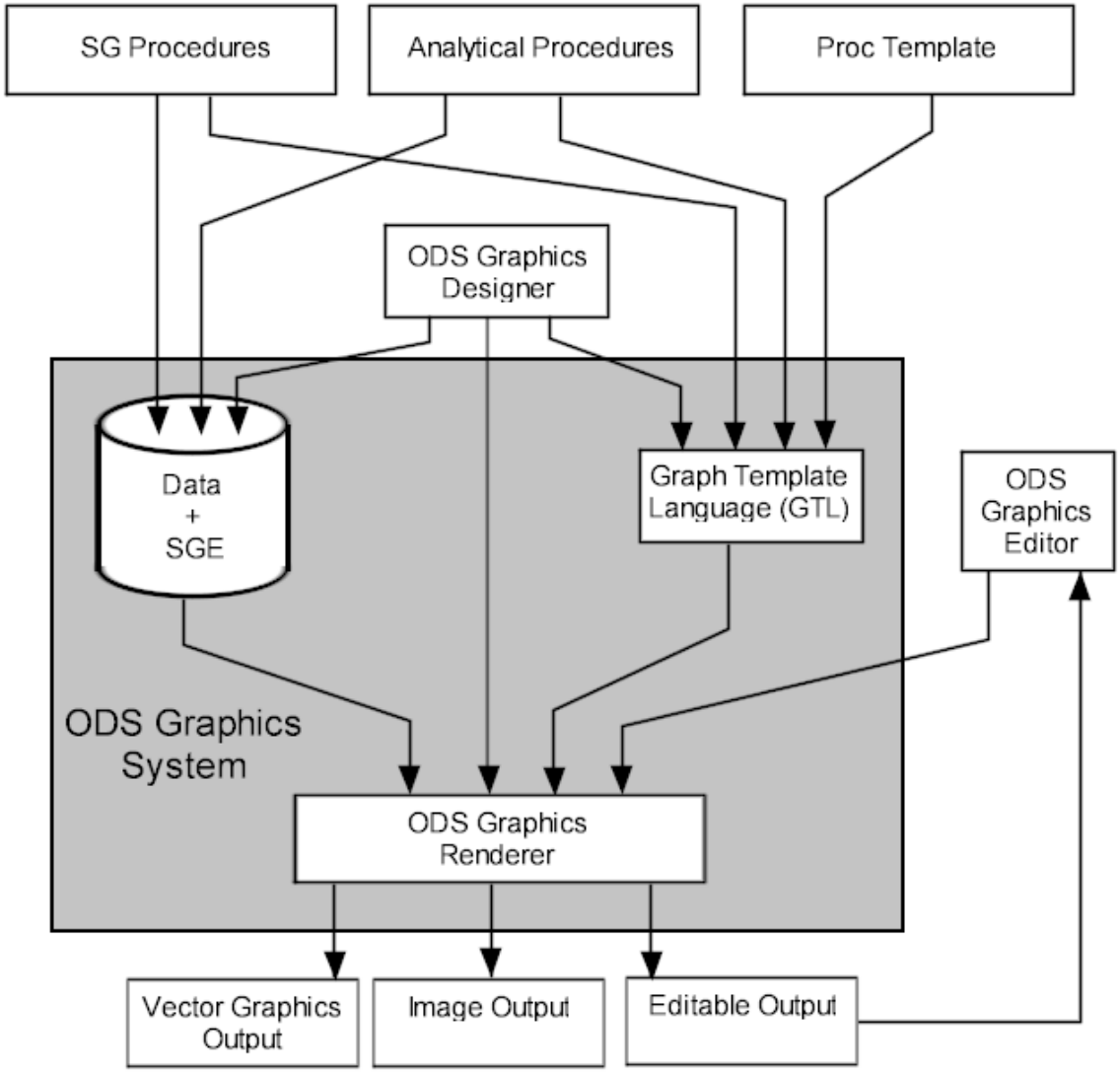
The Building Block Approach

- To build graphs, use containers, plots and insets
- Plot and inset determine what your graph displays.
- Containers determine how your graph is laid out.
- Same two plot statements in different layout containers.





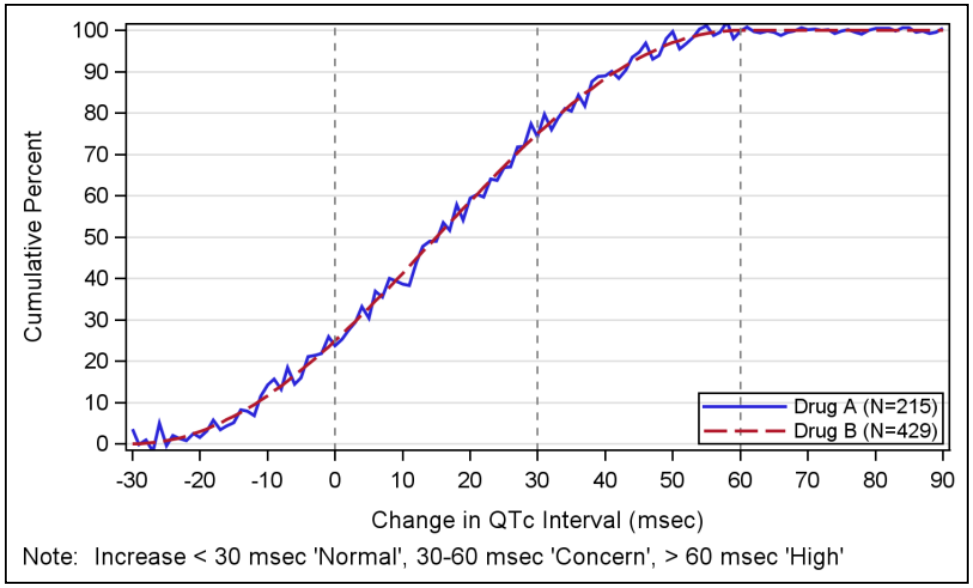
ODS Graphics System



The code!

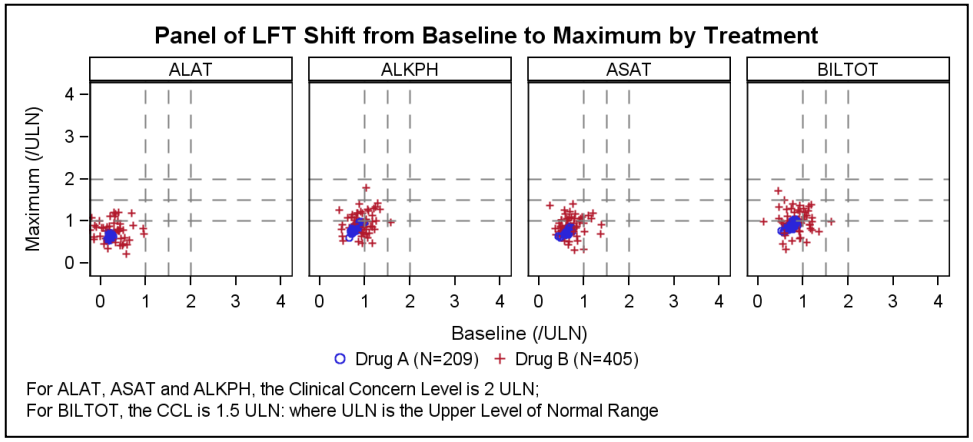
- Many of the graphs were built using the SG Procedures. The rest of the graphs were built using GTL.
- We will discuss the key features of the graph.
- Detailed examples are available at the support.sas.com web site: Samples & SAS Notes -> Browse By Topic -> SAS/GRAPH samples by industry -> Health and Life Sciences Industry.

Figure 1 – Empirical Distribution of QTc



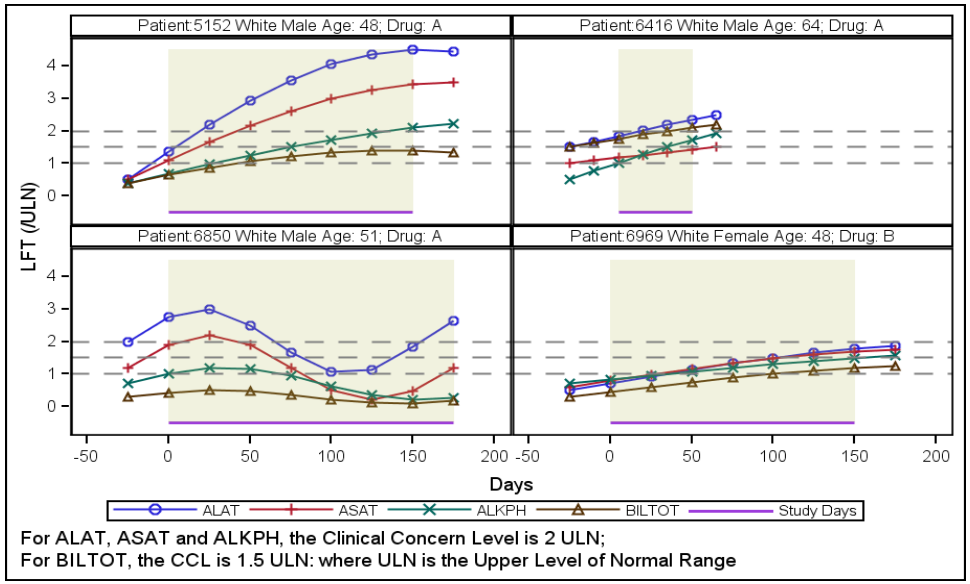

```
footnote j=left "Note: Increase < 30 msec 'Normal', 30-60 msec 'Concern', > 60 msec 'High' ";
proc sgplot data=qtc cycleattrs;
  series x=qtc y=cpa / legendlabel='Drug A (N=215)' lineattrs=(thickness=2);
  series x=qtc y=cpb / legendlabel='Drug B (N=429)' lineattrs=(thickness=2);
  keylegend / location=inside position=bottomright across=1;
  xaxis values=(-30 to 90 by 10);
  yaxis grid values=(0 to 100 by 10);
  refline 0 30 60 / axis=X lineattrs=(pattern=shortdash);
  run;
```

Figure 6 – Panel of LFT Shifts from Baseline



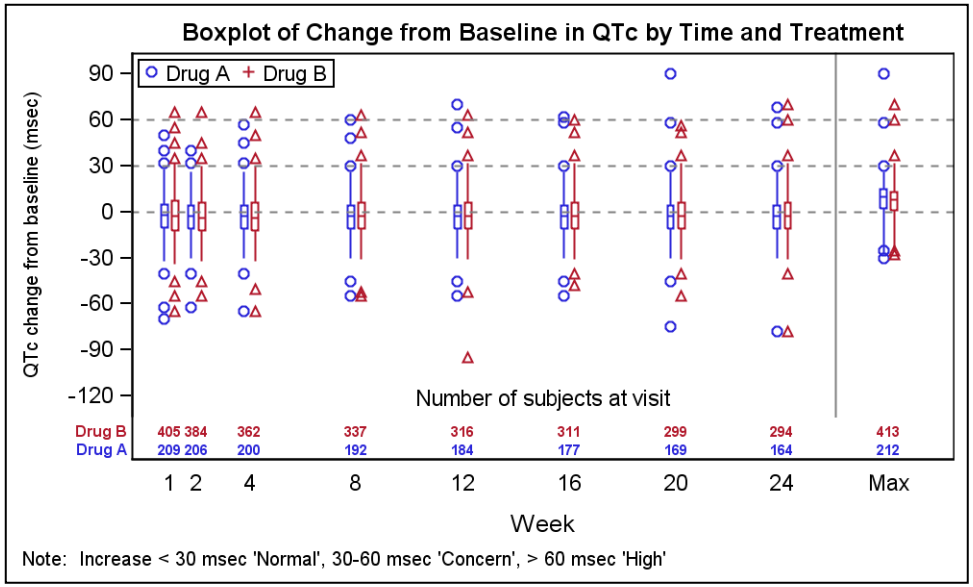

```
footnote1 j=l "For ALAT, ASAT and ALKPH, the Clinical Concern Level is 2 ULN;";
footnote2 j=l "For BILTOT, the CCL is 1.5 ULN: where ULN is the Upper Level of Normal Range";
proc sgpanel data=Safety6;
  panelby group / layout=panel columns=4 spacing=10 novarname;
  scatter x=pl y=sl / group=drug;
  refline 1 1.5 2 / axis=Y lineattrs=(pattern=dash);
  refline 1 1.5 2 / axis=X lineattrs=(pattern=dash);
  rowaxis integer min=0 max=4;
  colaxis integer min=0 max=4;
  keylegend / title="" noborder;
  run;
```

Figure 8 – LFT Patient Profile




```
footnote1 j=l "For ALAT, ASAT and ALKPH, the Clinical Concern Level is 2 ULN;";
footnote2 j=l "For BILTOT, the CCL is 1.5 ULN: where ULN is the Upper Level of Normal Range";
proc sgpanel data=Safety8;
  panelby patient / novarname;
  series x=days y=alat / markers lineattrs=(thickness=2px pattern=solid);
  series x=days y=asat / markers lineattrs=(thickness=2px pattern=solid);
  series x=days y=alkph / markers lineattrs=(thickness=2px pattern=solid);
  series x=days y=biltot / markers lineattrs=(thickness=2px pattern=solid);
  series x=days y=dval / lineattrs=(thickness=4px pattern=solid);
  refline 1 1.5 2 / axis=Y lineattrs=(pattern=dash);
  colaxis min=-50 max= 200; rowaxis label="LFT (/ULN)";
  keylegend / noborder;
  run;
```

Figure 2 – Boxplot of Change from Baseline



- We have used one Layout Overlay.
- Two Box Plots with DiscreteOffset, one each for Drug A and Drug B
- An InnerMargin with BlockPlots for the “At risk” table.
- An reference line on X axis with DiscreteOffset
- Multiple reference lines on Y axis.
- Inset Discrete Legend.
- Simulate interval axis.

Figure 2 – Boxplot of Change from Baseline

```
proc template;
  define statgraph QTcBox3;
    begingraph ;
      entrytitle 'Boxplot of Change from Baseline in QTc by Time and Treatment' / textattrs=(size=10);
      entryfootnote halign=left "Note: Increase < 30 msec 'Normal', 30-60 msec 'Concern', > 60 msec 'High' " ;
      layout overlay / xaxisopts=(display=(tickvalues line label) discreteopts=(tickvaluefitpolicy=truncate
            tickdisplaylist= ("1" "2" "" "4" "" "" "" "8" "" "" "" "12" "" "" "" "16" "" "" "" "20" "" "" "" "24" "" "" "" "Max")))
            yaxisopts=(label='QTc change from baseline (msec)' labelattrs=(size=8)
                        offsetmin=0.05 offsetmax=0.05linearopts=(viewmin=-120 viewmax=90
                    tickvaluesequence=(start=-120 end=90 increment=30)));
        < Plot Statements >
        entry halign=center "Number of subjects at visit" / valign=bottom;
        innermargin;
          blockplot x=week block=tota / label="Drug A" <options>;
          blockplot x=week block=totb / label="Drug B" <options>;
        endinnermargin;
      endlayout;
    endgraph;   end; run;
```

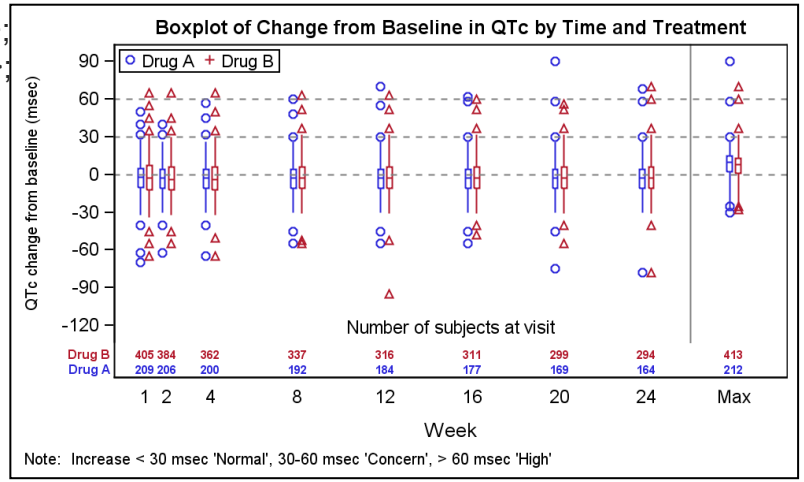


Figure 2 – Boxplot of Change from Baseline

```
layout overlay / <options>;
  boxplotparm x=week y=valuea stat=stat / outlineattrs=graphdata1
                discreteoffset=-0.2 boxwidth=0.25 capshape=bracket display=(outliers median)
                medianattrs=graphdata1 whiskerattrs=graphdata1 outlierattrs=graphdata1;
  boxplotparm x=week y=valueb stat=stat /
               discreteoffset=0.2 boxwidth=0.25 capshape=bracket display=(outliers median)
               medianattrs=graphdata2(pattern=solid) whiskerattrs=graphdata2(pattern=solid)
               outlineattrs=graphdata2(pattern=solid) outlierattrs=graphdata2(symbol=triangle);

  scatterplot x=eval(coln(1)) y=eval(coln(200)) / markerattrs=graphdata1 name="A" legendlabel="Drug A";
  scatterplot x=eval(coln(1)) y=eval(coln(200)) / markerattrs=graphdata2 name="B" legendlabel="Drug B";

  referenceline x=26 / lineattrs=(pattern=solid);
  referenceline y=0 / lineattrs=(pattern=shortdash);
  referenceline y=30 / lineattrs=(pattern=shortdash);
  referenceline y=60 / lineattrs=(pattern=shortdash);
  discretelegend "A" "B" / border=true location=inside
                halign=left valign=top;
  innermargin;
  endinnermargin;
endlayout;;
```

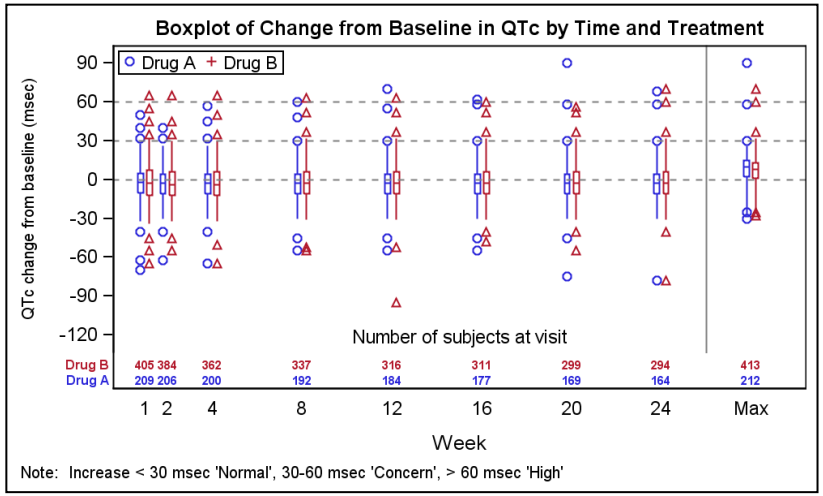
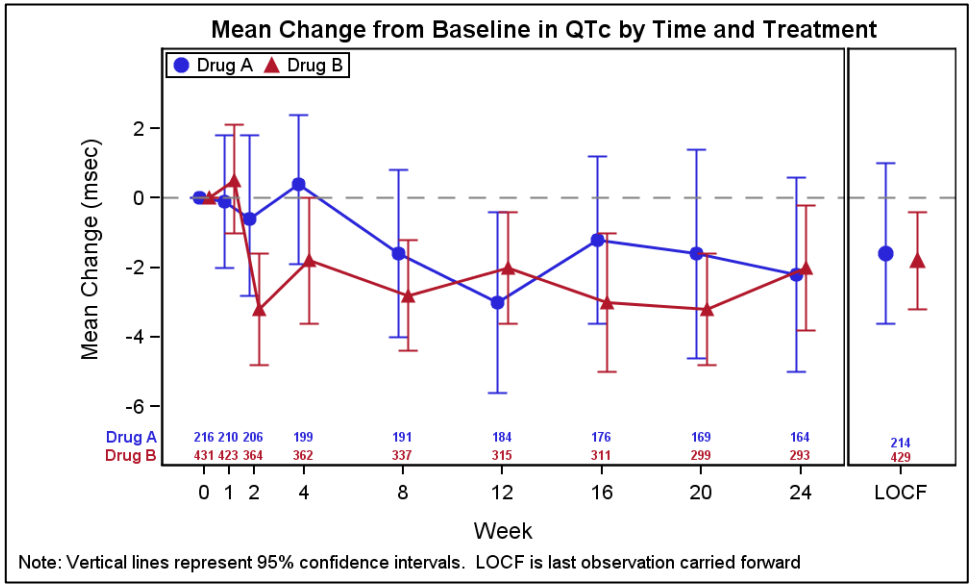


Figure 3 – Mean Change from Baseline for QTc



- We have used a LATTICE layout with two columns. The row weights are specified.
- We used the following plot statements in each cell:
  - We used two ScatterPlots jittered using the eval(week + 0.2) function.
  - We used two SeriesPlots jittered using the eval(week + 0.2) function.
  - We used an Inner Margin with two BLOCKPLOT statements to display the “At Risk” data.

Figure 3 – Mean Change from Baseline for QTc

```
proc  template;
  define statgraph safety.safety3;
    begingraph / designwidth=8in designheight=6in;
      entrytitle '                Mean Change from …"';
      layout lattice / columns=2 columnweights=(0.87 0.13)
                       rowdatarange=union;
        layout overlay / <options>;   /*--Left Cell--*/
        endlayout;
        layout overlay / <options>;  /*--Right Cell--*/
        endlayout;
      endlayout;
    entryfootnote halign=left "Note: Vertical …" ;
  endgraph;  end;   run;
```

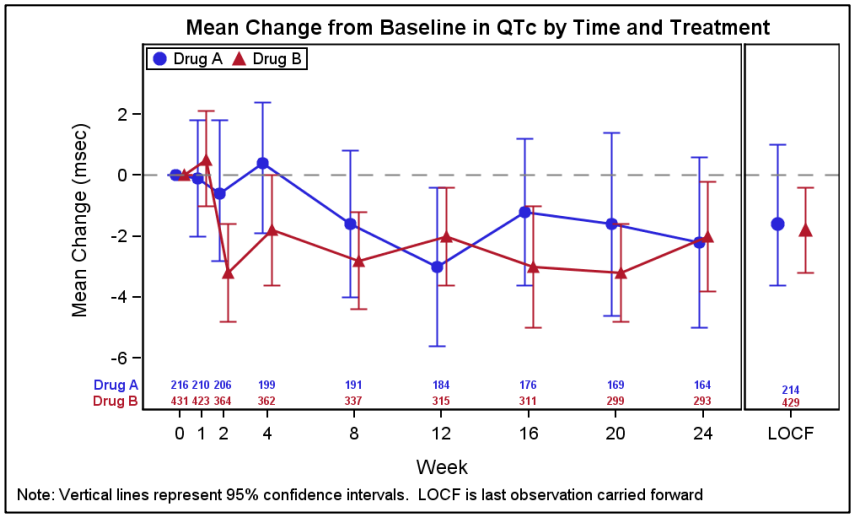
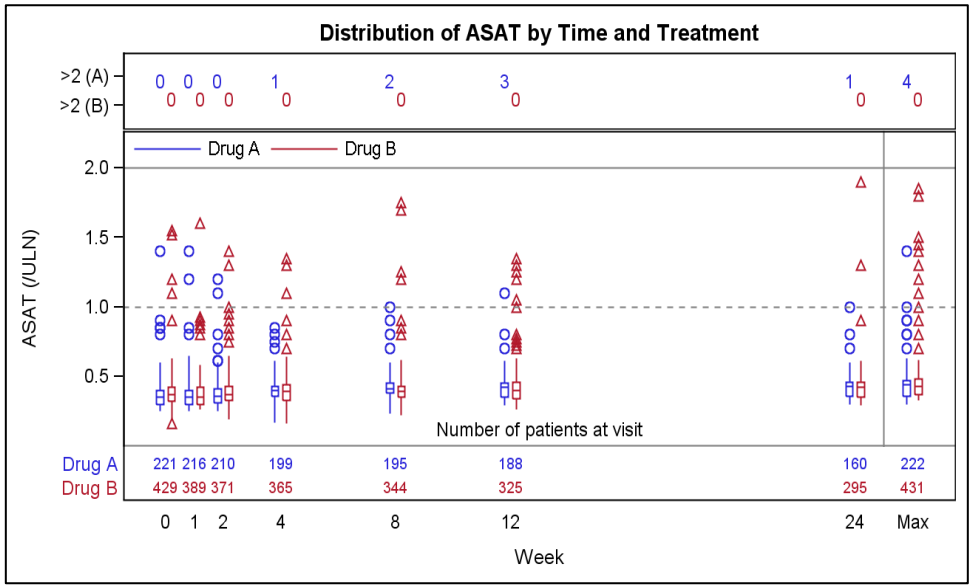


Figure 3 – Mean Change from Baseline for QTc

```
layout overlay / xaxisopts=(label='Week' linearopts=(tickvaluelist=(0 1 2 4 8 12 16 20 24)
                    tickdisplaylist=('0' '1' '2' '4' '8' '12' '16' '20' '24')))
                  yaxisopts=(label='Mean Change (msec)' offsetmax=0.1
                    linearopts=(viewmin=-6 viewmax=3));

  seriesplot x=eval(week-0.2) y=AMean / discreteoffset=-0.1 lineattrs=graphdata1(thickness=2);
  scatterplot x=eval(week-0.2) y=AMean / YErrorLower=ALow YErrorUpper=AHigh
              discreteoffset=-0.1 name='a' legendlabel='Drug A‘ <options>;
   seriesplot x=eval(week+0.2) y=BMean / discreteoffset=0.1 <options>;
   scatterplot x=eval(week+0.2) y=BMean / YErrorLower=Blow YErrorUpper=BHigh
              discreteoffset=0.1 name='b' legendlabel='Drug B‘<options>;
;
   referenceline y=0 / lineattrs=(pattern=dash);
   discretelegend 'a' 'b' / location=inside halign=left valign=top;

   innermargin;
      blockplot x=week block=BN / display=(label values) valuehalign=start  <options>;
      blockplot x=week block=AN / display=(label values) valuehalign=start  <options>;
   endinnermargin;

 endlayout;
```

Figure 4 – Distribution of ASAT



- We have used a LATTICE layout with two rows. The row weights are specified.
- The 1st row has two SCATTER plots in a LAYOUT OVERLAY, one for each drug.
  - The MARKERCHARACTER option is used to display the counts in place of the marker.
- The 2nd row contains two BOXPLOTPARM statements in a LAYOUT OVERLAY.
  - Use DiscreteOffset to jitter each box plot.
  - Use an Inner Margin with two BLOCKPLOT statements to display the “At Risk” data.
- Simulate interval axis.

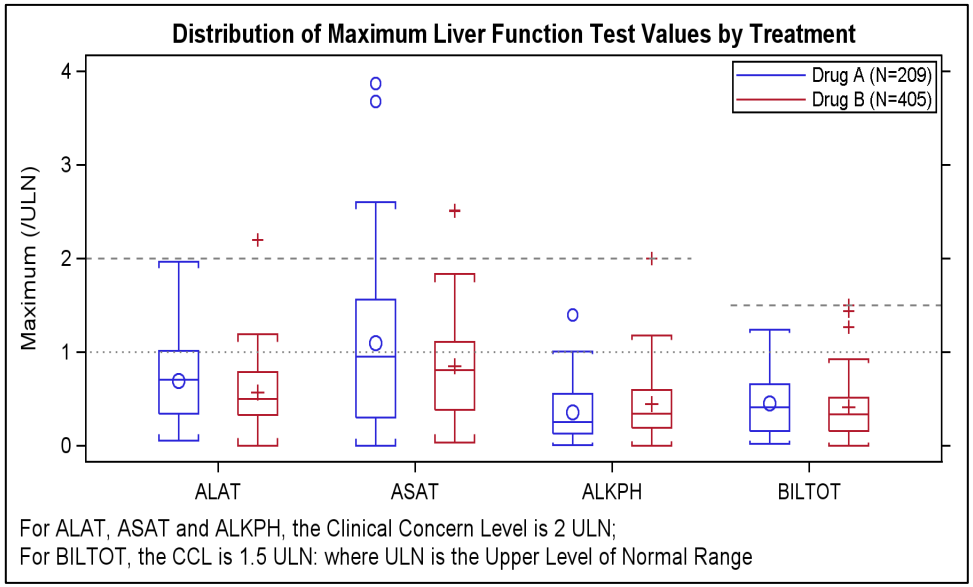
Figure 4 – Distribution of ASAT

```
proc template;
  define statgraph dist;
    begingraph;
    entrytitle 'Distribution of ASAT by Time and Treatment';
    layout lattice / columns=1 columndatarange=union rowweights=(0.15 0.85) rowgutter=1px;
       layout overlay / xaxisopts=(type=discrete display=none) yaxisopts=(label=' ');
          scatterplot x=week y=labelb / discreteoffset= 0.2 markercharacter=countb
               markercharacterattrs=graphdata2(size=9);
          scatterplot x=week y=labela / discreteoffset=-0.2 markercharacter=counta
               markercharacterattrs=graphdata1(size=9);
    endlayout;
    layout overlay / yaxisopts=(label='ASAT (/ULN)')
      yaxisopts=(linearopts=(tickvaluelist=(0.5 1.0 1.5 2.0)) offsetmin=0.0 offsetmax=0.1)
      xaxisopts = (label="Week" discreteopts=(tickdisplaylist=("0" "1" "2" "" "4" "" "" ""
               "8" "" "" "" "12" "" "" "" "" "" "" "" "" "" "" "" "24" "" "Max")) display=(tickvalues label));
      boxplotparm x=week y=valuea stat=stat / name="Drug A"
            discreteoffset=-0.2 boxwidth=0.25 capshape=bracket display=(outliers median)
            medianattrs=graphdata1 whiskerattrs=graphdata1(thickness=1)
            outlierattrs=graphdata1(size=8 weight=normal) outlineattrs=graphdata1(thickness=1) ;
      boxplotparm x=week y=valueb stat=stat / name="Drug B"
         discreteoffset=0.2 boxwidth=0.25 capshape=bracket display=(outliers median)
         medianattrs=graphdata2(pattern=solid) whiskerattrs=graphdata2(pattern=solid thickness=1)
         outlineattrs=graphdata2(pattern=solid thickness=1) outlierattrs=graphdata2(size=8
         symbol=triangle);
```

Figure 4 – Distribution of ASAT

```
innermargin;
  blockplot x=week block=totb / label="Drug B" display=(label values) valuehalign=start
        repeatedvalues=true valueattrs=graphdatatext(color=graphdata2:contrastcolor)
        filltype=Alternate labelattrs=graphvaluetext(color=graphdata2:contrastcolor);
  blockplot x=week block=tota / label="Drug A" display=(label values) valuehalign=start
        repeatedvalues=true valueattrs=graphdatatext(color=graphdata1:contrastcolor)
        filltype=Alternate labelattrs=graphvaluetext(color=graphdata1:contrastcolor);
endinnermargin;
```

Figure 5 – Distribution of Maximum LFT

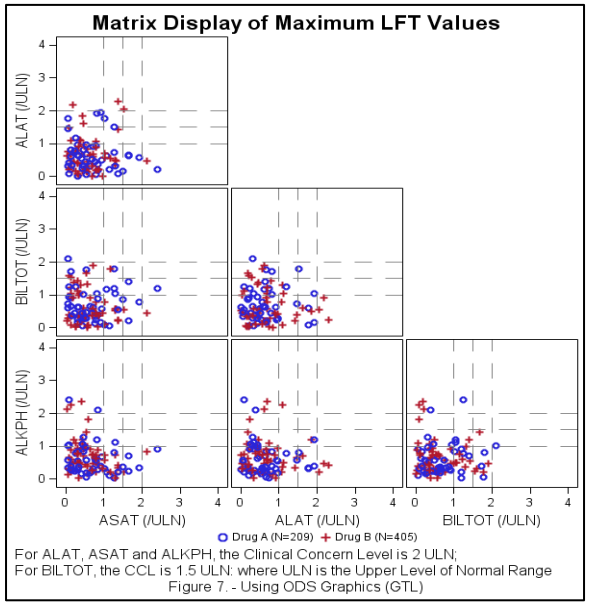


- We have used two BoxPlot statements, one for Drug A, one for Drug B.
- We have used DiscreteOffset to place the boxes side by side.
  - Reduce boxwidth for each box
- Add DropLines with DiscreteOffset to position the starting point in between mid point values.

Figure 5 – Distribution of Maximum LFT

```
proc template;
  define statgraph LFT_MaxByTrt;
    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));
          boxplot x=Test y=A / display=(mean median outliers caps) capshape=bracket
                              discreteoffset=-0.2 boxwidth=.2 name='a' legendlabel='Drug A (N=209)‘ <options>;
          boxplot x=Test y=B / display=(mean median outliers caps) capshape=bracket yaxis=y2
                              discreteoffset=0.2 boxwidth=.2 name='b' legendlabel='Drug B (N=405)‘<options>;
          referenceline y=1 / lineattrs=(pattern=dot);
          dropline x='BILTOT' y=1.5 / dropto=Y yaxis=y2 discreteoffset=-.4 lineattrs=(pattern=shortdash);
          dropline x='ALKPH' y=2.0 / dropto=Y yaxis=y discreteoffset=.4 lineattrs=(pattern=shortdash);
          discretelegend 'a' 'b' / location=inside valign=top halign=right across=1;
        endlayout;
        entryfootnote halign=left "For ALAT, ASAT and ALKPH, the Clinical Concern Level is 2 ULN;";
        entryfootnote halign=left "For BILTOT, the CCL is 1.5 ULN: where ULN is the Upper ….";
    endgraph;
end;
```

Figure 7 – Matrix Display of Maximum LFT

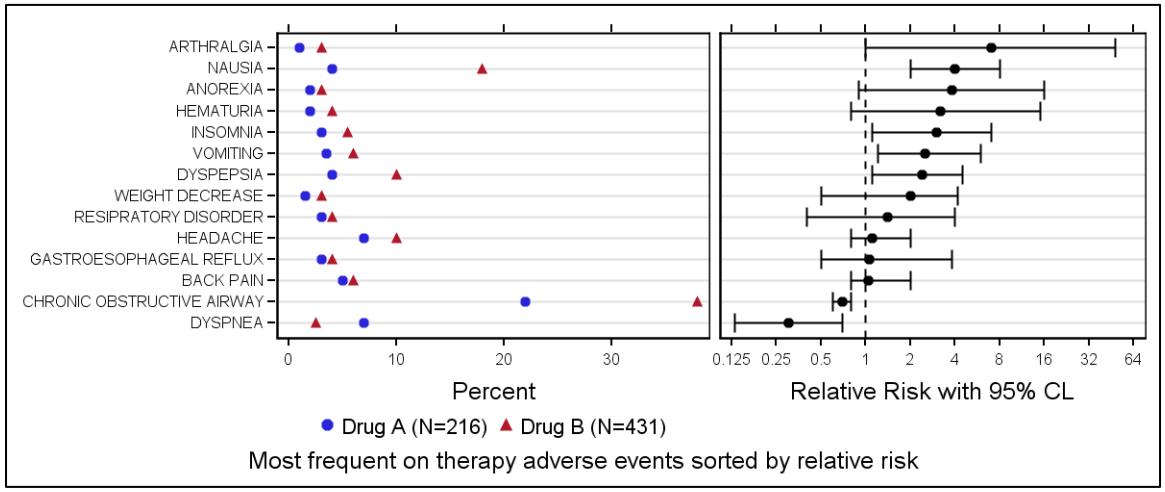


- Use a 3 x 3 Lattice layout.
- Populate each cell with a group ScatterPlot with the appropriate variables.
- Add reference lines for 1.0, 1.5 and 2.0.

Figure 7 – Matrix Display of Maximum LFT

```
/*--Row 1 --*/
    layout overlay / yaxisopts=(linearopts=(thresholdmax=1));
      scatterplot x=asat y=alat / group=drug name='scat';
      referenceline x=1 / lineattrs=(pattern=dash);
      referenceline x=1.5 / lineattrs=(pattern=dash);
      referenceline x=2 / lineattrs=(pattern=dash);
      referenceline y=1 / lineattrs=(pattern=dash);
      referenceline y=1.5 / lineattrs=(pattern=dash);
      referenceline y=2 / lineattrs=(pattern=dash);
    endlayout;

     layout overlay / yaxisopts=(linearopts=(thresholdmax=1));
    endlayout;

    layout overlay / yaxisopts=(linearopts=(thresholdmax=1));
    endlayout;
```

Figure 9 – Most Frequent On-Therapy AE

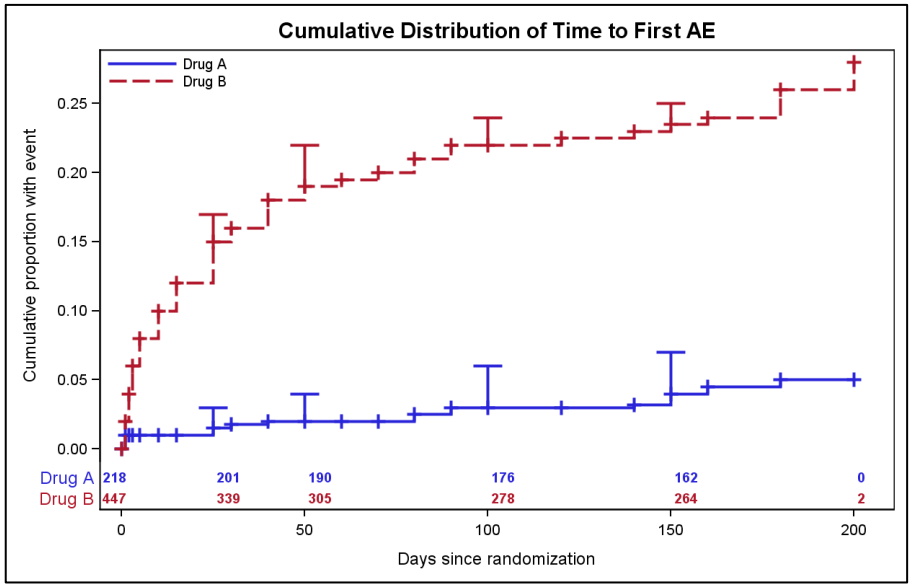


- We have used a Lattice Layout with 2 columns.
- Add two ScatterPlots in the left cell for Drug A and B
- Add a ScatterPlot with Error Bars in the right Cell.
- Add a reference line for X=1.

Figure 9 – Most Frequent On-Therapy AE

```
proc template;
  define statgraph safety.safety9;
    begingraph / designwidth=8in designheight=4in;
      entryfootnote "Most frequent on therapy adverse events sorted by relative risk";
      layout lattice / columns=2 rowdatarange=union;
          rowaxes;
              rowaxis / griddisplay=on display=(ticks tickvalues line) labelattrs=(size=9);
          endrowaxes;
          layout overlay / xaxisopts=(displaysecondary=(ticks) linearopts=(thresholdmax=0));
            scatterplot x=a y=ae / name='a' legendlabel='Drug A (N=)‘ <options>;
            scatterplot x=b y=ae / name='b' legendlabel='Drug B (N=)‘<options>;
            discretelegend 'a' 'b' / border=false;
        endlayout;
        layout overlay / xaxisopts=(label='Relative Risk wit…'  displaysecondary=(ticks)  type=log
                                logopts=(base=2 viewmin=0.125 viewmax=64  tickintervalstyle=logexpand));
            scatterplot x=mean y=ae / xerrorlower=low xerrorupper=high markerattrs=(symbol=circlefilled);
            referenceline x=1 / lineattrs=(pattern=shortdash);
       endlayout;
      endlayout;
    endgraph;
 end;
run;
```

Figure 10 – Cumulative Distribution to First AE

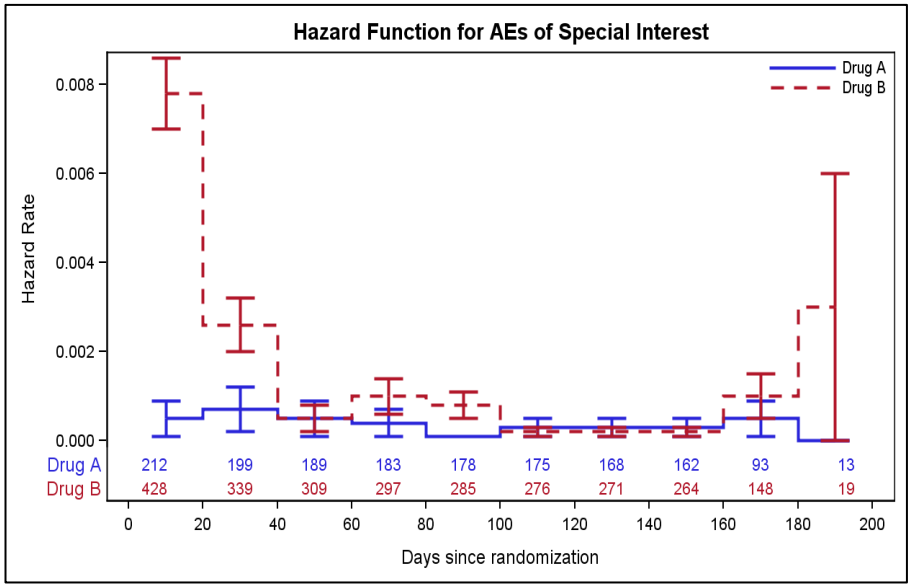


- Use a single Layout Overlay
- Use a ScatterPlot and StepPlot for Drug A.
- Use a ScatterPlot and StepPlot for Drug B.
- Use an InnerMargin with BlockPlots to show the “At Risk” numbers.

Figure 10 – Cumulative Distribution to First AE

```
proc template;
  define statgraph safety.safety10;
    begingraph;

      layout overlay / yaxisopts=(label='Cumulative proportion with event') xaxisopts=(offsetmax=0.05);
         stepplot x=day y=drugA / errorupper=upperA lineattrs=graphdata1(pattern=solid thickness=2)
                  errorbarattrs=graphdata1(pattern=solid thickness=2) name='a';
         scatterplot x=day y=drugA / markerattrs=graphdata1(symbol=plus size=9 weight=bold);
         stepplot x=day y=drugB / errorupper=upperB lineattrs=graphdata2(thickness=2)
                   errorbarattrs=graphdata2(pattern=solid thickness=2) name='b';
         scatterplot x=day y=drugB / markerattrs=graphdata2(symbol=plus size=9 weight=bold);
         discretelegend 'a' 'b' / location=inside halign=left valign=top across=1 border=false;

         innermargin;
            blockplot x=day block=riskB / display=(values label) labelattrs=GraphData2
                      valueattrs=graphdatatext(color=graphdata2:contrastcolor);
            blockplot x=day block=riskA / display=(values label) labelattrs=GraphData1
                      valueattrs=graphdatatext(color=graphdata1:contrastcolor) ;
          endinnermargin;

       endlayout;

       entryfootnote halign=center 'Figure 10. - Using ODS Graphics (GTL)';
    endgraph;
end;  run;
```

Figure 11 – Hazard Function for AE



- Use a single Layout Overlay
- Use a ScatterPlot and StepPlot for Drug A.
- Use a ScatterPlot and StepPlot for Drug B.
- Use an InnerMargin with BlockPlots to show the “At Risk” numbers.

Figure 11 – Hazard Function for AE

```
proc template;
  define statgraph safety.safety11;
    begingraph ;
       entrytitle  'Hazard Function for AEs of Special Interest';

       layout overlay / yaxisopts=(label='Hazard Rate')
                        xaxisopts=(linearopts=(viewmin=0 viewmax=200
                                               tickvaluelist=(0 20 40 60 80 100 120 140 160 180 200)));
         stepplot x=day y=A / errorupper=AHigh errorlower=ALow justify=center
                              name='a <options>';
         stepplot x=day y=B / errorupper=BHigh errorlower=BLow justify=center
                             l name='b'<options>'; ;
          discretelegend 'a' 'b' / location=inside halign=right valign=top across=1 border=false;

          innermargin;
             blockplot x=eval(day-10) block=BRisk / display=(values label)  valuehalign=center
                       labelattrs=GraphData2 valueattrs=GraphData2;
             blockplot x=eval(day-10) block=ARisk / display=(values label)  valuehalign=center
                      labelattrs=GraphData1 valueattrs=GraphData1;
          endinnermargin;
        endlayout;
     endgraph;
  end;
run;
```

Conclusions

- All of the above graphs recommended by domain experts can be done using ODS Graphics.
- Some graphs are very easy and natural. Some are not.
- The following missing features in SAS 9.2 ODS Graphics were identified to make such graphs easier.
  - Box Plots on Interval axis.
  - Box Plots with cluster grouping.
- These features have been added for SAS 9.3.
- We would love to hear from you as you create new graphs for analysis of Clinical Data so we can make sure the right features are in place.

Thanks!

The work presented here was done by

Sanjay Matange

Sr. Development Manager for ODS Graphics.

Sanjay.Matange@sas.com

The work is detailed in http://www.wuss.org/proceedings10/analy/3022_3_ANL-Matange.pdf

We look forward to hear from you about graphs you are working on that are difficult to do using ODS graphics.

Contact information

Jan René Larsen

Advisory Industry Consultant

SAS Institute A/S, Denmark

sdkjrl@sdk.sas.com

www.sas.com