

# **Clinical Graphs using ODS Graphics**

## **Sanjay Matange, SAS Institute, Cary, NC**

### **ABSTRACT**

Safety data for clinical trials is often collected in tabular form. The process of interpretation and analysis can be made more effective by the usage of graphical display of this data along with the generated statistics. Displays such as Liver Function Test Panels and Matrices, Hazard Function Plots, Patient Profiles, Adverse Event Plots, Lab Tests for Blood Chemistry and Hematology are commonly used in the Health and Life Sciences domain. Many of these graphs are easy to do using one of the many tools available in the ODS Graphics tool set. This paper will describe in detail how to create these standard graphs for the Health and Life Sciences industry using ODS Graphics.

### **INTRODUCTION**

High quality graphs are essential for analysis of data in the Health and Life Sciences domain. Large volumes of such data are generated in the duration of a clinical trial, and visualizing the data itself, along with the derived statistics is essential for the analysis of such data. Some of the key features of the graphs include:

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

The goal of this paper is to show how to create graphs commonly used in the Health and Life Sciences domain using ODS Graphics. In this paper, we will take sample graphs commonly used for analysis of safety data for Clinical Trials, Patient Profile Graphs, graphs for viewing laboratory tests, etc., and show the code needed to build each graph in detail. These graphs have been designed by professionals in the field, who have the necessary domain expertise to design the right graph. It is not the intent of this paper to evaluate the effectiveness of such graphs. So, the design of the graph itself is beyond the scope of this paper. In this paper, we will cover various techniques that can be used to build such graphs thus providing the information you need to customize them to your needs.

### **GRAPHS FOR HEALTH AND LIFE SCIENCES**

We will review and build some common graphs that have been frequently used in the Health and Life Sciences domain. We will start with some graphs discussed in the paper "Graphical Approaches to the Analysis of Safety Data from Clinical Trials" Ohad Amit, et. al., Pharmaceutical Statistics, 2008. Additionally, we will cover other popular graphs in this field. These are:

1. Empirical distribution function for maximum change in QTc.
2. Box Plot of change from baseline in QTc by time and treatment.
3. Mean change from baseline in QTc by time and treatment.
4. Distribution of ASAT by time and treatment.
5. Distribution of maximum LFT values by treatment.
6. Panel of LFT shift from baseline to maximum by treatment
7. Matrix display of Maximum LFT values.
8. LFT Patient profiles.
9. Most frequent on-therapy adverse events sorted by relative risk.
10. Cumulative distribution (with SEs) of time to first AE of special interest
11. Hazard functions for AEs of special interest (with SEs).
12. Adverse event timeline.
13. Lipid Profile Over Time.
14. Immunology Profile by Treatment.

Let us review each graph along with the code needed to create the graphs using the Statistical Graphics (SG) Procedures or the Graph Template Language (GTL).

## Example 1: Empirical distribution function for maximum change in QTc

Figure 1A shows the Empirical distribution function for maximum change in QTc as shown in Figure 1 in the paper by Amit et. al. Figure 1B shows a similar graph created using the SGPLOT procedure.

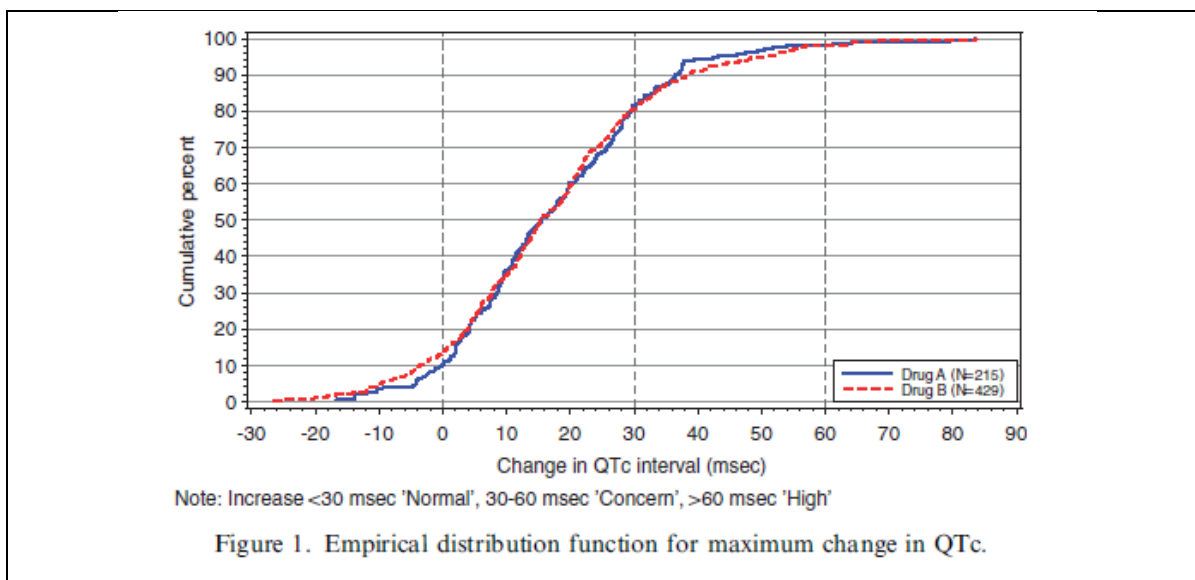


Figure 1A – From “Graphical Approaches to the Analysis of Safety Data from Clinical Trials”. Amit, et. al.

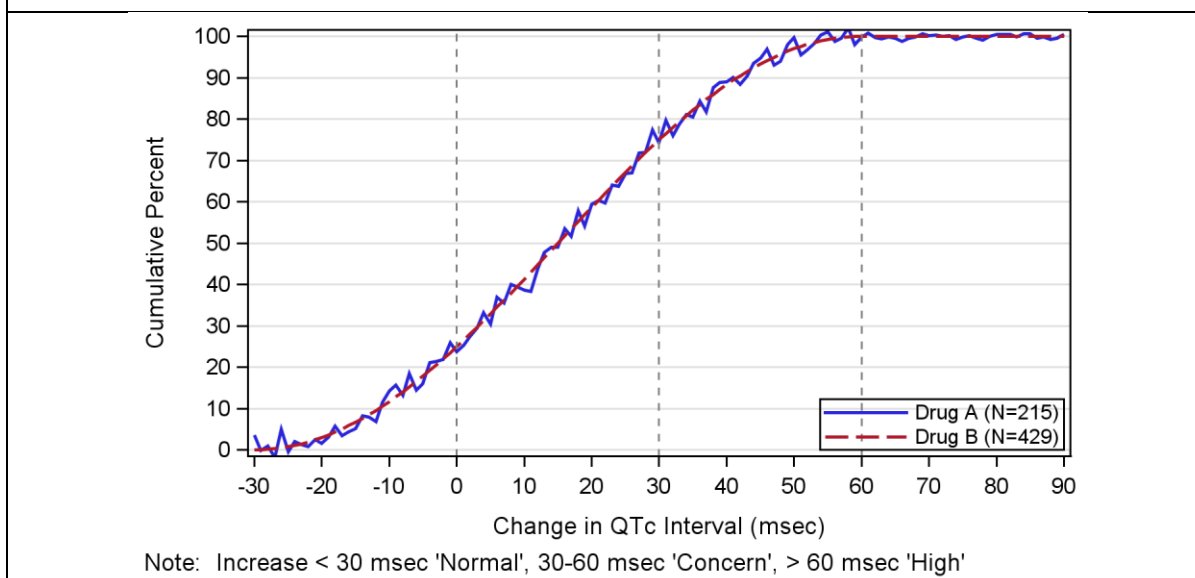


Figure 1B – Same graph as above created using the SGPLOT Procedure with simulated data.

```
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 1C – SAS program using SGPLOT Procedure

The example shown in Figure 1B illustrates a key feature of the Statistical Graphics (SG) Procedures and Graph Template Language (GTL). The syntax uses a building-block approach to creating complex graphs. The graph is made by overlaying two separate series plots in the same graph. In fact, any number of compatible plot statements can be combined in this way to create graphs. The data for the graph in Figure 1B is in the form of three separate columns, one each for qtc, cpa, and cpb. Here are the key features of this graph:

- A SERIES plot of cpa by qtc for drug A.
- A SERIES plot of cpb by qtc for drug B.
- The line thickness for each graph is increased using the lineattrs=(thickness=2) option.
- A legend label for each plot is set. This label will be shown in the legend.
- The KEYLEGEND statement is used to position the legend inside the data area.
- The CYCLEATTRS option automatically assigns different visual attributes to the multiple plot statements.

## Example 2: Boxplot of change from baseline in QTc by time and treatment

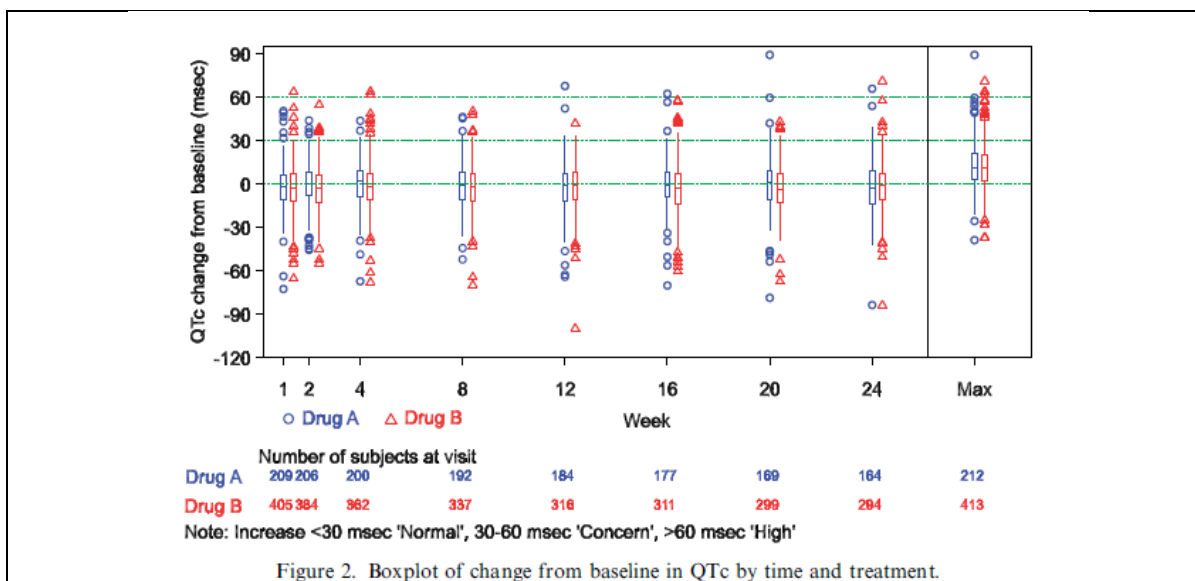


Figure 2A – From “Graphical Approaches to the Analysis of Safety Data from Clinical Trials”. Amit, et. al.

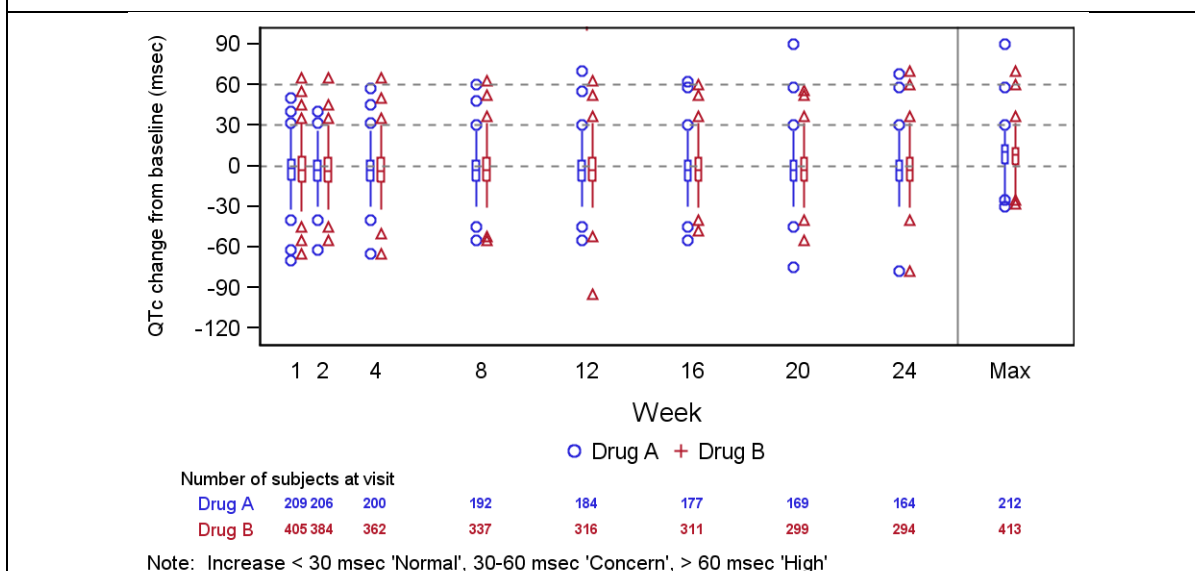


Figure 2B – Same graph as above created using GTL with simulated data.

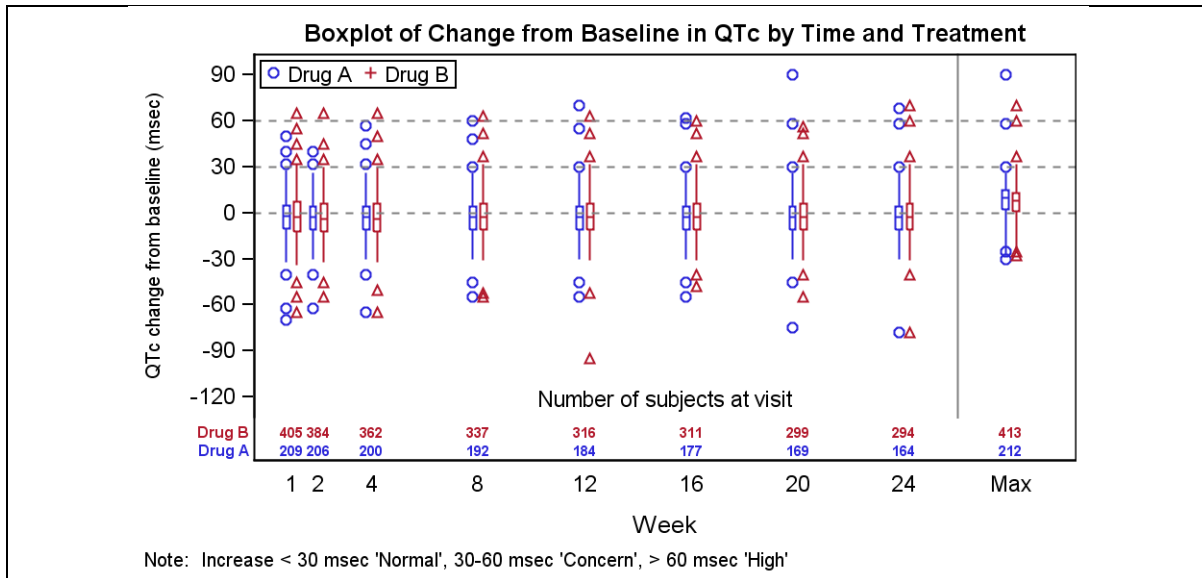


Figure 2C –This layout provides a tighter alignment of the data, the “At Risk” values and the X axis.

```

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)));
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;
entry halign=center "Number of subjects at visit" / valign=bottom;
innermargin;
blockplot x=week block=tota / label="Drug A" display=(label values) valuehalign=start
repeatedvalues=true filltype=Alternate
valueattrs=graphdatatext(color=graphdata1:contrastcolor size=6 weight=bold)
labelattrs=graphvaluetext(color=graphdata1:contrastcolor size=7 weight=bold);
blockplot x=week block=totb / label="Drug B" display=(label values) valuehalign=start
repeatedvalues=true filltype=Alternate
valueattrs=graphdatatext(color=graphdata2:contrastcolor size=6 weight=bold)
labelattrs=graphvaluetext(color=graphdata2:contrastcolor size=7 weight=bold);
endinnermargin;
endlayout;
endgraph; end; run;

```

Figure 2D – SAS program using GTL

The example graph shown in Figure 2B is created using a Lattice of four rows. The top row contains the box plot, followed by a row for the “At Risk” label, and the “At Risk” table.

An alternative graph layout is shown in Figure 2C. This version brings the “At Risk” table closer to the rest of the data. The SAS code is shown in Figure 2D. Some notable items in this graph are:

- Only one LAYOUT OVERLAY container is needed.
- The “At Risk” information is placed in the bottom INNER MARGIN within the graph itself.
- A blank region is created at the upper end of the graph by setting YOFFSETMAX.
- The Legend is inserted in the upper left corner of the graph.
- By default, if a box plot is put in the legend, it is represented by a line. To get markers in the boxplot, we have to do the following:
  - Add two scatter plots in the graph, and include these in the legend.
  - The data for the scatter plots is forced outside the data range of the Y axis
  - The data for these scatter plots is not in the data set, but is being generated on the fly, using the expressions: `.eval (coln ( ) )`,
  - The Y axis range is set between -120 and 90. Hence the scatter plots are not shown in the plot, but do show up in the legend.

As can be seen in the program in Figure 2D, the Box plot X axis is discrete by default. Since the data is not equally spaced, we have to simulate the “scaling” by inserting data for the missing weeks. This issue will be addressed in the SAS 9.3 release which supports “Grouped” Box Plots on an interval axis. This is shown in Figure 2E below.

```
proc template;
define statgraph QTcBoxTest;
begingraph ;
  entryfootnote halign=left "Note: Increase < 30 msec 'Normal', 30-60 msec 'Concern', > 60 msec 'High' ";
  layout overlay / xaxisopts=(type=linear label='Week' linearopts=(tickvaluefitpolicy=none
    tickvaluelist=(1 2 4 8 12 16 20 24 28) tickdisplaylist=('1' '2' '4' '8' '12' '16' '20' '24' 'Max')))
    yaxisopts=(label='QTc change from baseline (msec)' labelattrs=(size=8) yoffsetmax=0.1
    linearopts=(viewmin=-120 viewmax=90 tickvaluesequence=(start=-120 end=90 increment=30)));
  boxplotparm x=eval(week-0.2) y=valuea stat=stat / name="A" legendlabel="Drug A" <options>;
  boxplotparm x=eval(week+0.2) y=valueb stat=stat / name="B" legendlabel="Drug A" <options>;
  referenceline x=26 / lineattrs=(pattern=solid);
  referenceline y=0 / lineattrs=(pattern=shortdash);
  referenceline y=30 / lineattrs=(pattern=shortdash);
  referenceline y=60 / lineattrs=(pattern=shortdash);
  entry halign=center "Number of subjects at visit" / valign=bottom textattrs=(size=7);
  discretelegend "A" "B" / border=true valign=top halign=left location=inside;

  innermargin;
    blockplot x=week block=tota / <options>;
    blockplot x=week block=totb / <options>;
  endinnermargin;

endlayout;
endgraph;
end;
run;
```

Figure 2E – SAS 9.3 sample code for same graph using interval box plot statement.

### Example 3: Mean Change from Baseline in QTc by time and treatment.

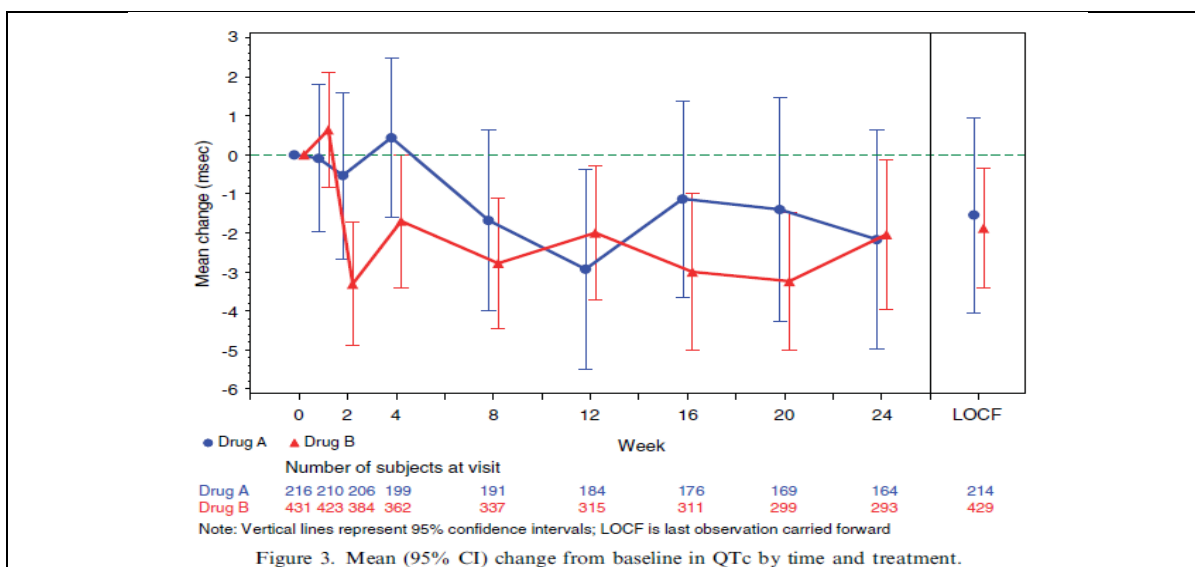
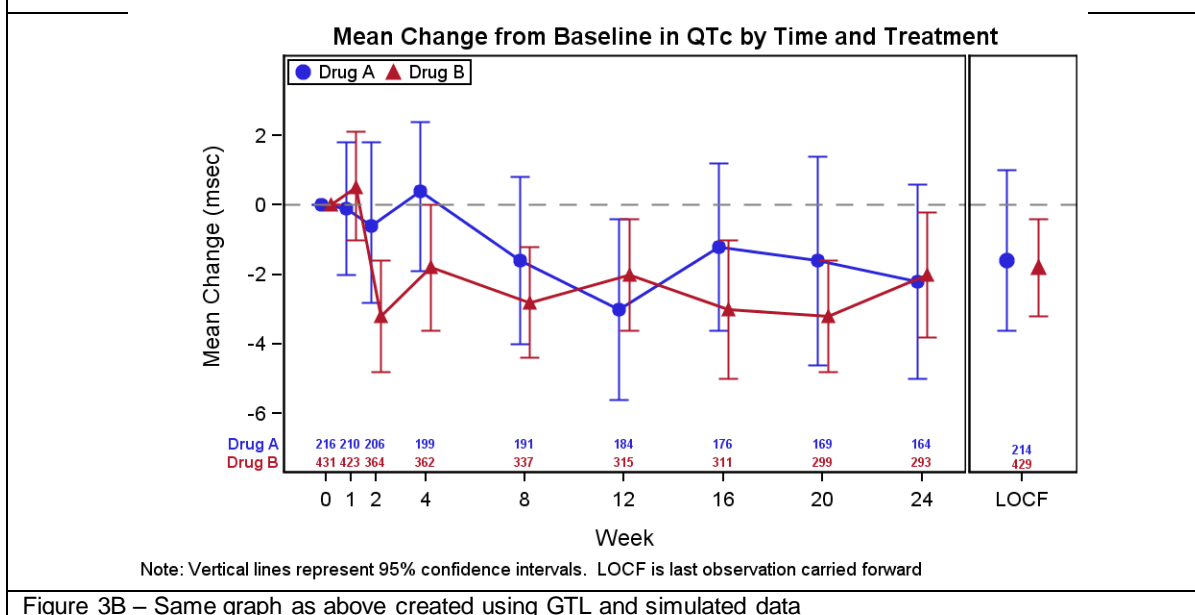


Figure 3A – From “Graphical Approaches to the Analysis of Safety Data from Clinical Trials”. Amit, et. al.



For the graph shown in Figure 3B the data for the LOCF value is to be plotted separately, without the series join. So, we have to plot that data point separately. To achieve this, we have used a LATTICE layout, with one row and two columns and set the column weights appropriately. The data is separated into two sets of columns, one for the first 9 observations, and one for the last observation. The graph uses the following structure as shown in Figure 3C.

- We use two SCATTERPLOTS, one for each drug, plotted with a separation of 0.2 weeks using the  $X=\text{eval}(\text{week} + 0.2)$  in the plot statements. We use GRAPHDATA1 and GRAPHDATA2 elements for each.
- We use two SERIESPLOTS, one for each drug, plotted with a separation of 0.2 weeks using the  $X=\text{eval}(\text{week} + 0.2)$  in the plot statements. We use GRAPHDATA1 and GRAPHDATA2 elements for each.
- We use two BLOCKPLOTS in the INNERMARGIN to display the “At Risk” data colored by treatment.
- Then, we repeat this process for the 2<sup>nd</sup> cell with the LOCF data. SERIESPLOT can be skipped.
- Once again we have moved the legend and the “At Risk” table inside the graph for better alignment.

```

proc template;
define statgraph safety.safety3;
beginningraph / designwidth=8in designheight=6in;
entrytitle '          Mean Change from Baseline in QTc by Time and Treatment';
layout lattice / columns=2 columnweights=(0.87 0.13) rowdatarange=union;
rowaxes;
rowaxis / label='Mean Change (msec)' offsetmax=0.1 linearopts=(viewmin=-6 viewmax=3);
endrowaxes;
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
markerattrs=graphdata1(symbol=circlefilled size=9) errorbarattrs=graphdata1
name='a' legendlabel='Drug A';
seriesplot x=eval(week+0.2) y=BMean / discreteoffset=0.1
lineattrs=graphdata2(pattern=solid thickness=2);
scatterplot x=eval(week+0.2) y=BMean / YErrorLower=BLow YErrorUpper=BHigh discreteoffset=0.1
markerattrs=graphdata2(symbol=trianglefilled size=9) errorbarattrs=graphdata2(pattern=solid)
name='b' legendlabel='Drug B';
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
repeatedvalues=true filltype=Alternate labelattrs=graphdata2
valueattrs=graphdatatext(color=graphdata2:contrastcolor size=6 weight=bold)
labelattrs=graphvaluertext(color=graphdata2:contrastcolor size=7 weight=bold);
blockplot x=week block=AN / display=(label values) valuehalign=start
repeatedvalues=true filltype=Alternate labelattrs=graphdata1
valueattrs=graphdatatext(color=graphdata1:contrastcolor size=6 weight=bold)
labelattrs=graphvaluertext(color=graphdata1:contrastcolor size=7 weight=bold);
endinnermargin;
endlayout;

layout overlay / yaxisopts=(linearopts=(viewmin=-6 viewmax=3) label='Mean Change (msec)')
xaxisopts=(linearopts=(tickvaluelist=(1) tickdisplaylist=('LOCF')) display=(line ticks tickvalues));
seriesplot x=eval(week2-1) y=AMean2 / discreteoffset=-0.1 lineattrs=graphdata1(thickness=2);
scatterplot x=eval(week2-1) y=AMean2 / YErrorLower=ALow2 YErrorUpper=AHigh2
discreteoffset=-0.1 markerattrs=graphdata1(symbol=circlefilled size=15)
errorbarattrs=graphdata1;
seriesplot x=eval(week2+1) y=BMean2 / discreteoffset=0.1
lineattrs=graphdata2(pattern=solid thickness=2);
scatterplot x=eval(week2+1) y=BMean2 / YErrorLower=BLow2 YErrorUpper=BHigh2
discreteoffset=0.1 markerattrs=graphdata2(symbol=trianglefilled size=15)
errorbarattrs=graphdata2(pattern=solid);
referenceline y=0 / lineattrs=(pattern=dash);
innermargin;
blockplot x=week2 block=BN2 / display=( values) valuehalign=start
repeatedvalues=true filltype=Alternate labelattrs=graphdata2
valueattrs=graphdatatext(color=graphdata2:contrastcolor size=6 weight=bold)
labelattrs=graphvaluertext(color=graphdata2:contrastcolor size=7 weight=bold);
blockplot x=week2 block=AN2 / display=( values) valuehalign=start
repeatedvalues=true filltype=Alternate labelattrs=graphdata1
valueattrs=graphdatatext(color=graphdata1:contrastcolor size=6 weight=bold)
labelattrs=graphvaluertext(color=graphdata1:contrastcolor size=7 weight=bold);
endinnermargin;
endlayout;
endlayout;
entryfootnote halign=left "Note: Vertical lines represent 95% confidence intervals. LOCF is ...";
endgraph; end; run;

```

Figure 3C – SAS 9.3 sample code for same graph using interval box plot statement.

#### Example 4: Distribution of ASAT by time and treatment

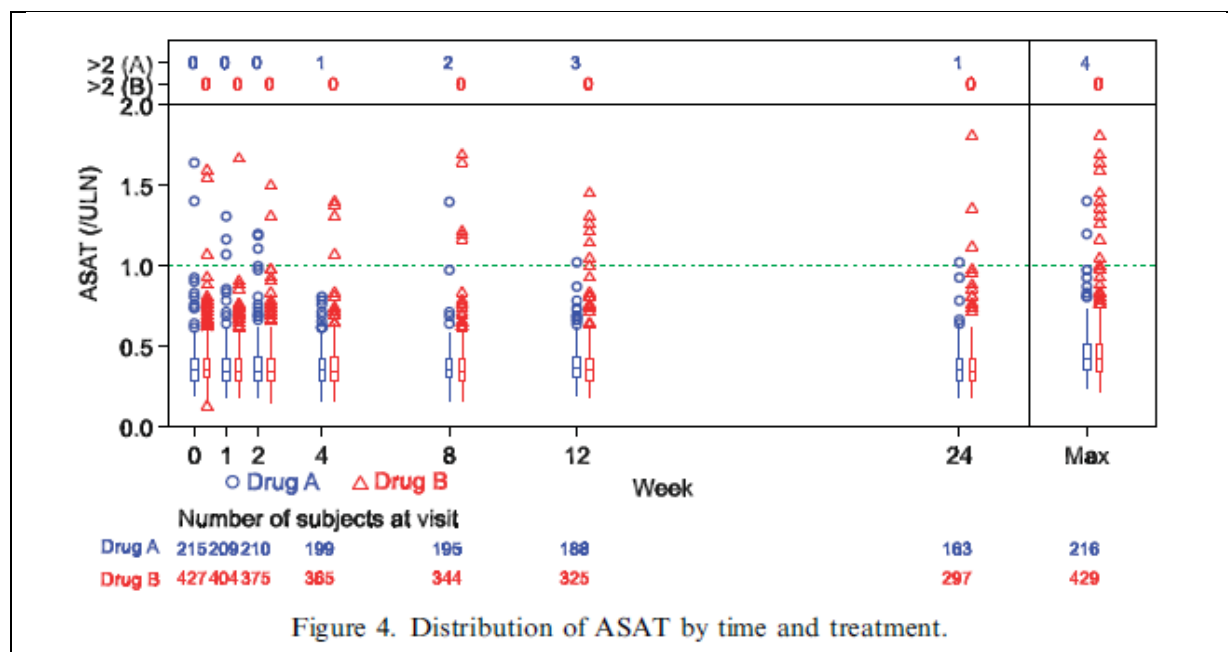
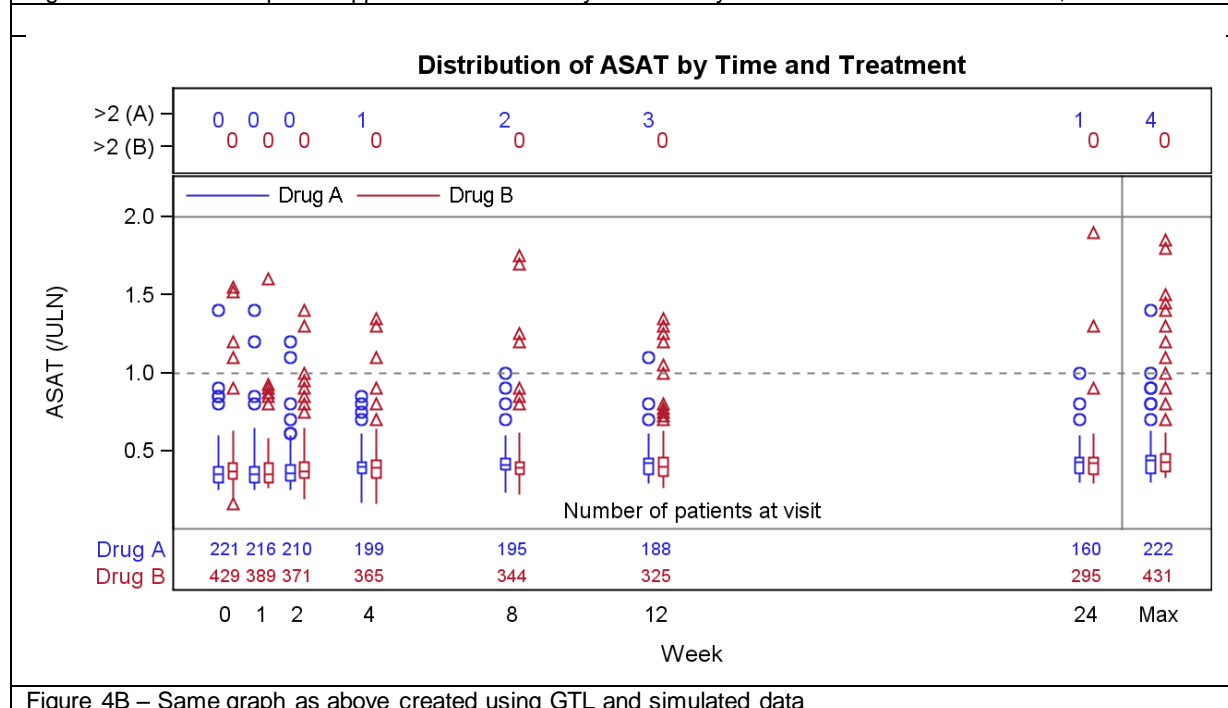


Figure 4A – From “Graphical Approaches to the Analysis of Safety Data from Clinical Trials”. Amit, et. al.



The graph shown in Figure 4B, the count of values above  $2 \times \text{ULN}$  are shown in the strip above the box plots. The code for this graph template is shown in Figure 4C. The key aspects of the graph are:

- We have used a LATTICE layout with two rows. The row weights are specified.
- The 1<sup>st</sup> 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 2<sup>nd</sup> row contains two BOXPLOT statements in a LAYOUT OVERLAY, one for each drug.
- The 2<sup>nd</sup> row has an inner margin with two BLOCKPLOT statements to display the “At Risk” data.



- A BOXPLOTARM draws the boxes based on summarized data. We have used that to simulate the graph. In a real case, we will have detailed data and we can use the BOXPLOT statement to get the same results.
- The BOXPLOT supports a discrete axis only for SAS 9.2. The data is not at even date intervals, so we have to introduce the additional data observations to get the right scaling on the X axis.
- As mentioned in Figure 2E, A BOXPLOT with support for interval data will be released with SAS 9.3. Usage of that statement in the future will further simplify the code for this graph.

```

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=counttb
markercharacterattrs=graphdata2(size=9);
scatterplot x=week y=labela / discreteoffset=-0.2 markercharacter=countta
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);

referenceline y=1.0 / lineattrs=(pattern=shortdash);
referenceline x=25 / lineattrs=(pattern=solid);
referenceline y=2.0 / lineattrs=(pattern=solid);
referenceline y=0 / lineattrs=(pattern=solid);
discretelegend "Drug A" "Drug B" / location=inside valign=top halign=left border=false;
entry halign=center "Number of patients at visit" / valign=bottom;

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;

endlayout;

endLayout;
endgraph;
end;
run;

```

Figure 4C – SAS program using GTL Template

**Example 5: Distribution of maximum LFT values by treatment.**

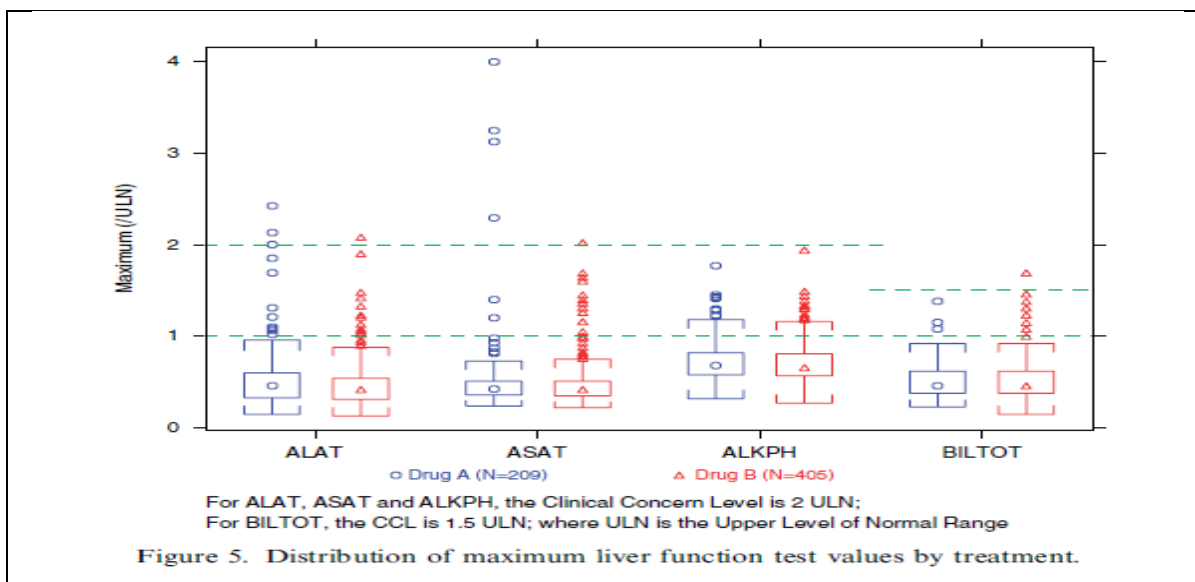
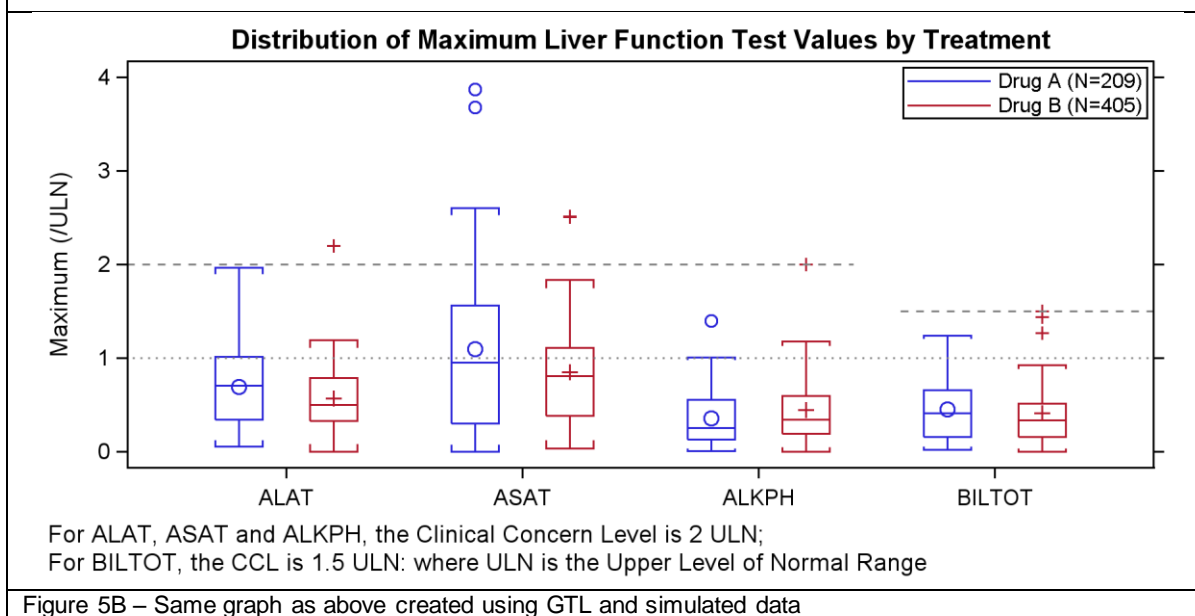


Figure 5A – From “Graphical Approaches to the Analysis of Safety Data from Clinical Trials”. Amit, et. al.



The graph shown in Figure 5B, uses a LAYOUT LATTICE container and multiple plot statements as shown here:

- The data for drug A is plotted by the test using the 1<sup>st</sup> BOXPLOT statement.
- The data for drug B is plotted by the test using the 2<sup>nd</sup> BOXPLOT statement.
- Since the X axis is DISCRETE, normally, the 2<sup>nd</sup> plot would be drawn directly on top of 1<sup>st</sup>, with each box centered on the tick value on the axis.
- To make the two boxes draw side by side, we have used the DISCRETEOFFSET option. This option shifts the center of each box by a fraction of the midpoint spacing. In this case, we have shifted the 1<sup>st</sup> box plot to the left by 20% and the 2<sup>nd</sup> boxplot to the right by 20%.
- For this to work well, we also need to shrink the width of the boxes to a smaller value. By default, the box width is 80% of the midpoint spacing. We have set the box width of each to 20% to get the correct result.
- The 1<sup>st</sup> box plot is assigned the GRAPHDATA1 style element and the 2<sup>nd</sup> boxplot is assigned GRAPHDATA2 element for visual properties.

- A DROPLINE statement can be used to create a partial reference line to one of the axes. If the axis is DISCRETE, then the value is from the midpoint location for the value. If you want the drop line to start between two mid points, you can use the DISCRETEOFFSET option and offset it by the fraction needed.
- In this case, we have used two DROPLINE statements, one from BILTOT with a -40% discrete offset, and one from ALKPH a +40% discrete offset.
- The BOXPLOT statements allows for customization of various elements of the box. Visual elements can be set separately for Box Outline, Mean, Median, Outliers, Whiskers, etc. So, attributes for each have to be set to obtain the right effect.
- For the linear Y axis, the graph will automatically determine the data extent of the axis from the data values. To ensure we get exactly the axis we want, the Y axis options are set accordingly.

```

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)'
outlineattrs=graphdata1(thickness=1) outlierattrs=graphdata1 meanattrs=graphdata1
medianattrs=graphdata1(thickness=1) whiskerattrs=graphdata1
whiskerattrs=graphdata1(thickness=1);
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)'
outlineattrs=graphdata2(pattern=solid thickness=1) outlierattrs=graphdata2
meanattrs=graphdata2 medianattrs=graphdata2(pattern=solid thickness=1)
whiskerattrs=graphdata2(pattern=solid thickness=1);
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 5C – SAS program using GTL Template

# **Example 6: Panel of LFT shift from baseline to maximum by treatment**

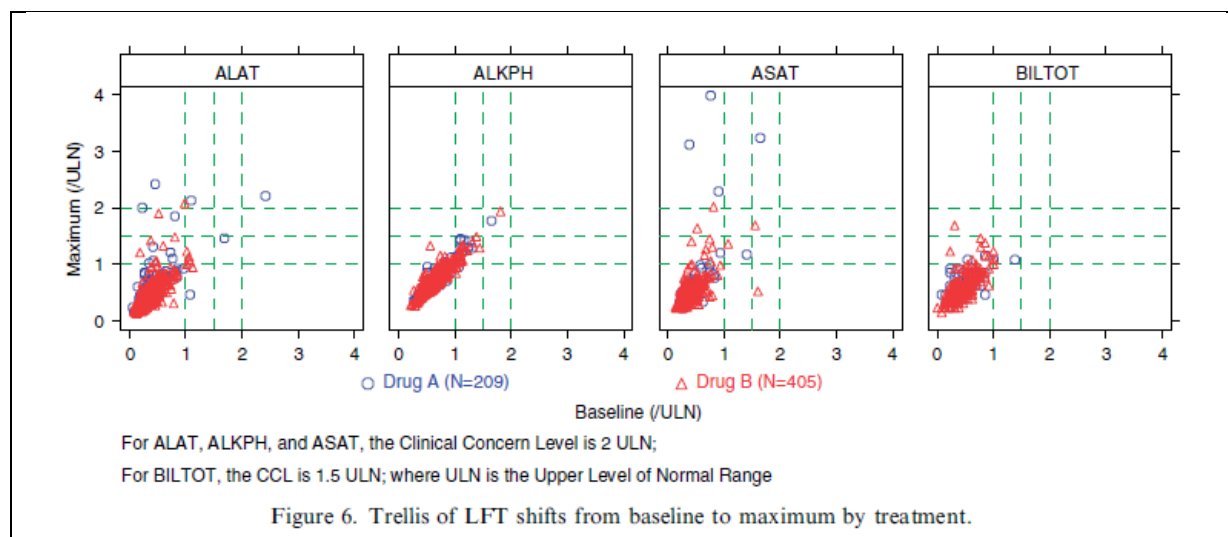
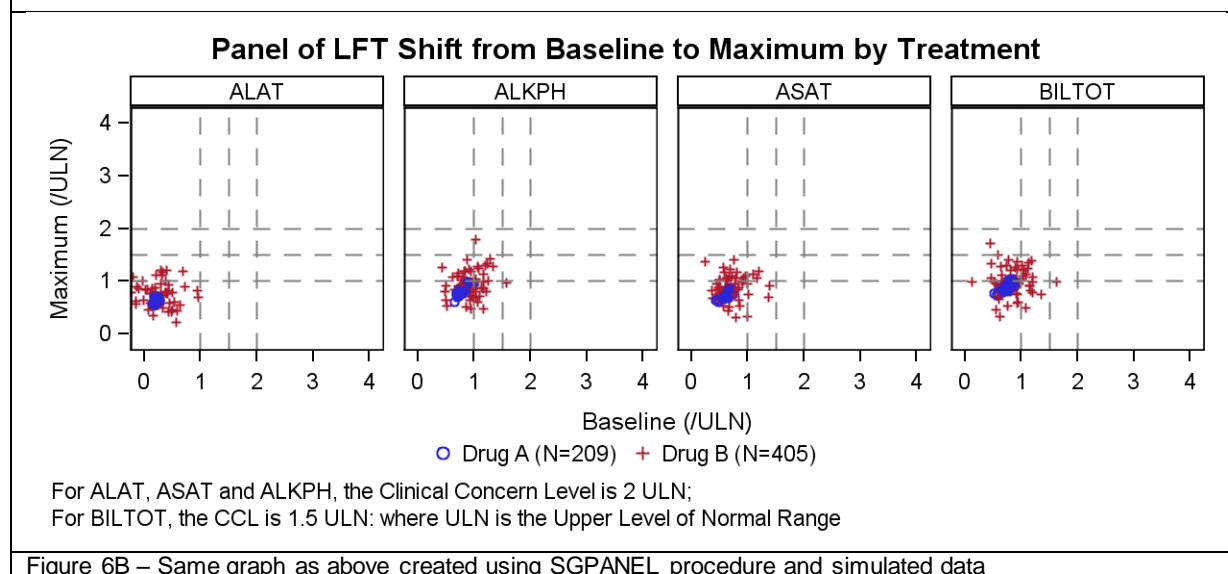


Figure 6A – From “Graphical Approaches to the Analysis of Safety Data from Clinical Trials”. Amit, et. al.



This graph is very straightforward, using the SGPPANEL procedure with PANELBY=group.

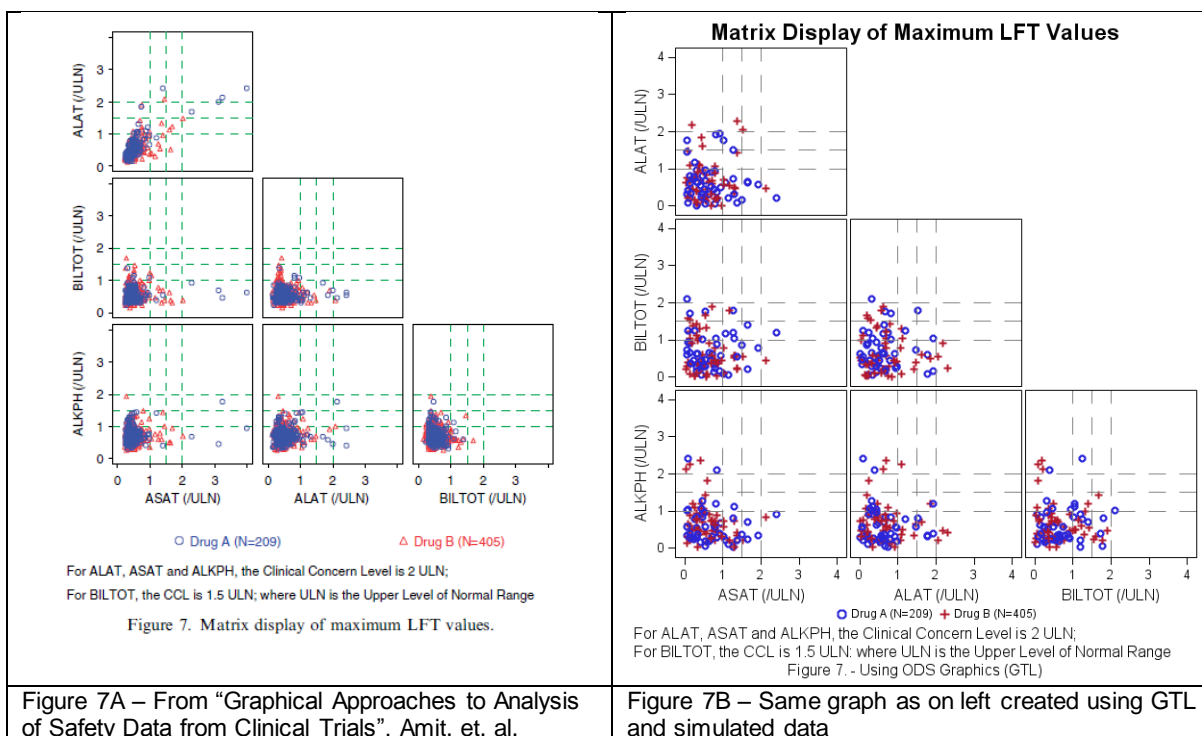
```

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 6C – SAS program using GTL Template

## Example 7: Matrix display of Maximum LFT values



The template code for this graph looks lengthy, but the structure is very simple. It uses a 3x3 LATTICE. Each cell of the layout uses a similar structure for the scatter plot and reference lines as shown in Figure 7C and 7D.

```
proc template;
define statgraph Safety.Safety7;
beginninggraph / designwidth=8in designheight=12in;
  layout lattice / columns=3 order=rowmajor columndata=union rowdata=union < options>;
  sidebar;
  layout overlay;
  discretelegend 'scat' / across=2 border=false;
  endlayout;
endsidebar;
columnaxes;
  columnaxis / lineartype=(thresholdmax=1 viewmax=4);
  columnaxis / lineartype=(thresholdmax=1 viewmax=4);
  columnaxis / lineartype=(thresholdmax=1 viewmax=4);
endcolumnaxes;
rowaxes;
  rowaxis / lineartype=(thresholdmax=1 viewmax=4);
  rowaxis / lineartype=(thresholdmax=1 viewmax=4);
  rowaxis / lineartype=(thresholdmax=1 viewmax=4);
endrowaxes;

/*--See Figure 7D for this block of code --*/
layout overlay;
plot statements;
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 Level of ...";
entryfootnote halign=center "Figure 7. - Using ODS Graphics (GTL)";
endgraph;
end;
run;
```

Figure 7C – SAS program using GTL Template

/\*--The following block of code is inserted at "See Figure 7D" in Figure 7C--\*/

```

/*--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;

/*--Row 2 --*/
layout overlay / yaxisopts=(linearopts=(thresholdmax=1));
scatterplot x=asat y=biltot / group=drug name='scat';
referenceline x=1 / lineattrs=(pattern=dash); /*-- Repeat for X=1.5 and X=2--*/
referenceline y=1 / lineattrs=(pattern=dash); /*-- Repeat for Y=1.5 and Y=2--*/
endlayout;

layout overlay / yaxisopts=(linearopts=(thresholdmax=1));
scatterplot x=alat y=biltot / group=drug name='scat';
referenceline x=1 / lineattrs=(pattern=dash); /*-- Repeat for X=1.5 and X=2--*/
referenceline y=1 / lineattrs=(pattern=dash); /*-- Repeat for Y=1.5 and Y=2--*/
endlayout;

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

/*--Row 3 --*/
layout overlay / yaxisopts=(linearopts=(thresholdmax=1));
scatterplot x=asat y=alkph / group=drug name='scat';
referenceline x=1 / lineattrs=(pattern=dash); /*-- Repeat for X=1.5 and X=2--*/
referenceline y=1 / lineattrs=(pattern=dash); /*-- Repeat for Y=1.5 and Y=2--*/
endlayout;

layout overlay / yaxisopts=(linearopts=(thresholdmax=1));
scatterplot x=alat y=alkph / group=drug name='scat';
referenceline x=1 / lineattrs=(pattern=dash); /*-- Repeat for X=1.5 and X=2--*/
referenceline y=1 / lineattrs=(pattern=dash); /*-- Repeat for Y=1.5 and Y=2--*/
endlayout;

layout overlay / yaxisopts=(linearopts=(thresholdmax=1));
scatterplot x=biltot y=alkph / group=drug name='scat';
referenceline x=1 / lineattrs=(pattern=dash); /*-- Repeat for X=1.5 and X=2--*/
referenceline y=1 / lineattrs=(pattern=dash); /*-- Repeat for Y=1.5 and Y=2--*/
endlayout;

```

Figure 7D – SAS program using GTL Template

## Example 8: LFT Patient profiles

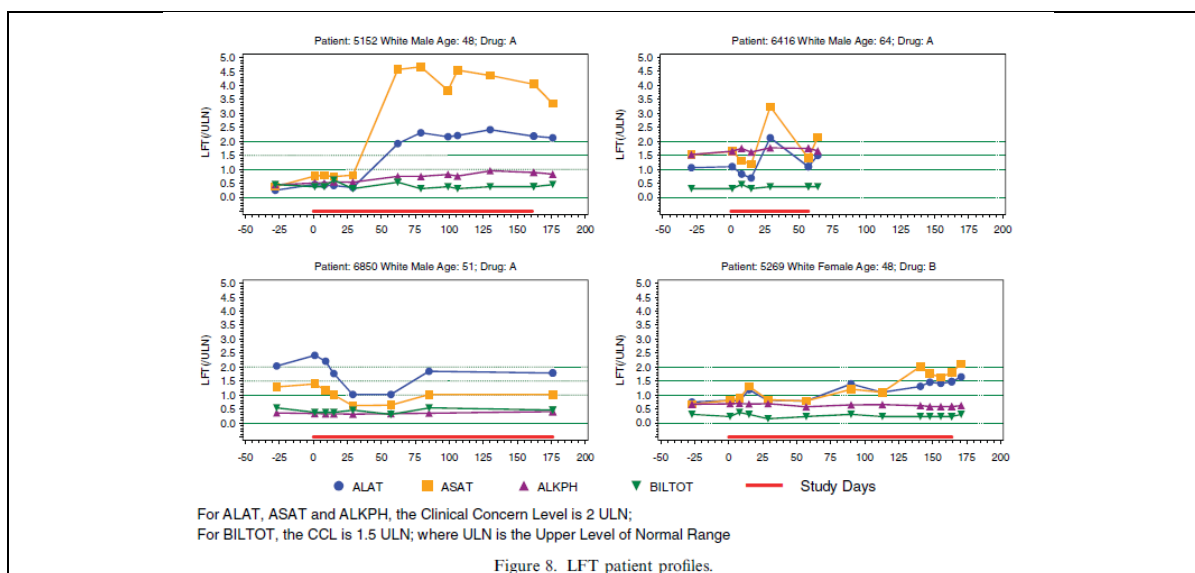


Figure 8A – From “Graphical Approaches to the Analysis of Safety Data from Clinical Trials”. Amit, et. al.

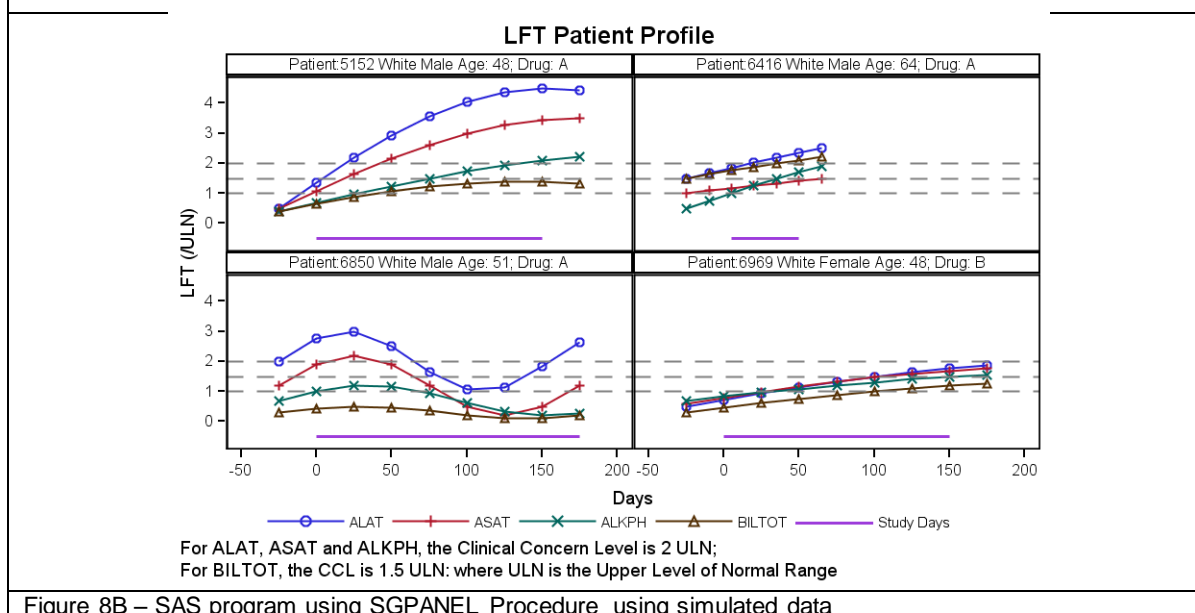


Figure 8B – SAS program using SGPanel Procedure using simulated data

```
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 8C – SAS program using SGPanel Procedure

## Example 9: Most Frequent On Therapy Adverse Events

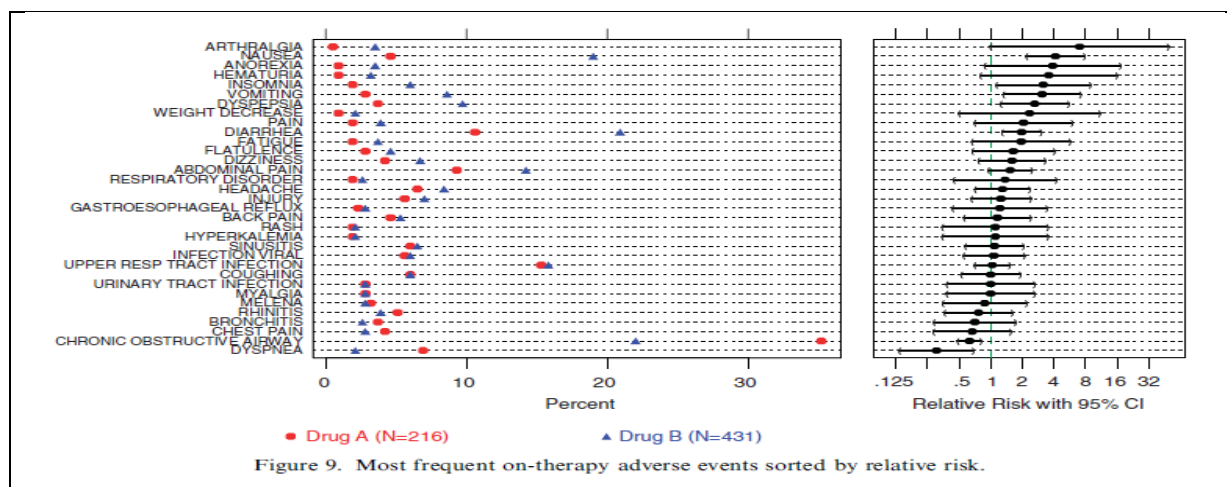


Figure 9A – From “Graphical Approaches to the Analysis of Safety Data from Clinical Trials”. Amit, et. al.

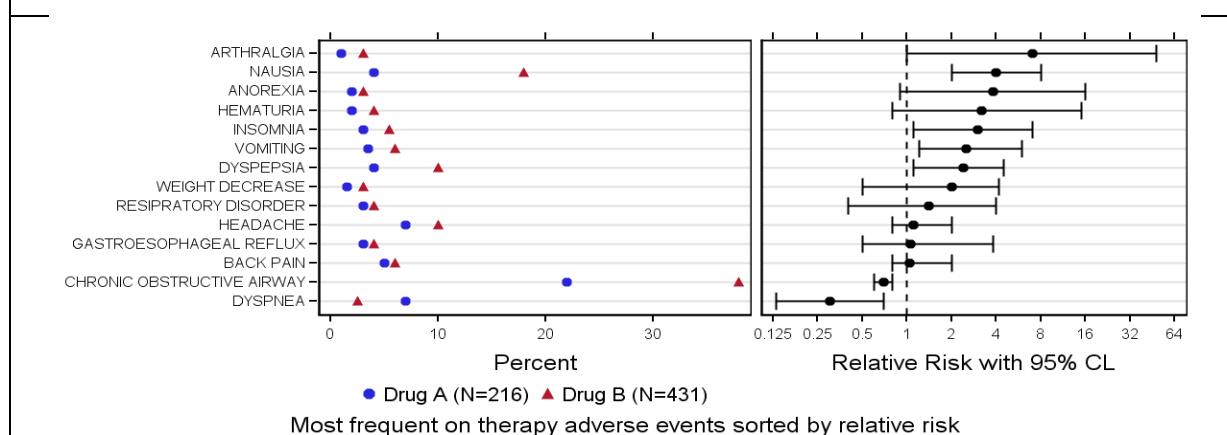


Figure 9B – Similar graph as above created using GTL and partial simulated data

```
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;
endrowaxes;
layout overlay / axisopts=(displaysecondary=(ticks) linearopts=(thresholdmax=0));
scatterplot x=a y=ae / markerattrs=graphdata1(symbol=circlefilled) name='a' legendlabel='Drug A (N=)';
scatterplot x=b y=ae / markerattrs=graphdata2(symbol=trianglefilled) name='b' legendlabel='Drug B (N=)';
discretelegend 'a' 'b' / border=false;
endlayout;
layout overlay / axisopts=(label='Relative Risk with 95% CL' 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 9C – SAS program using GTL



**Example 10: Cumulative distribution (with SEs) of time to first AE of special interest.**

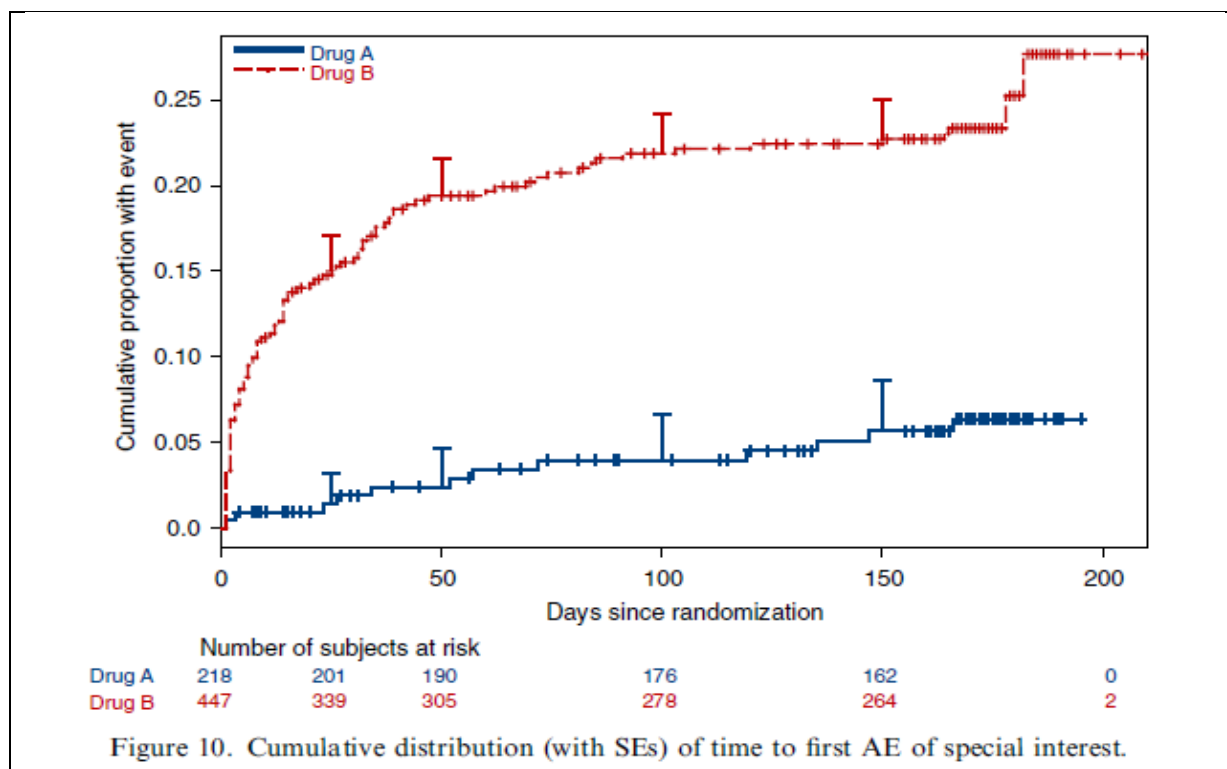


Figure 10A – From “Graphical Approaches to the Analysis of Safety Data from Clinical Trials”. Amit, et. al.

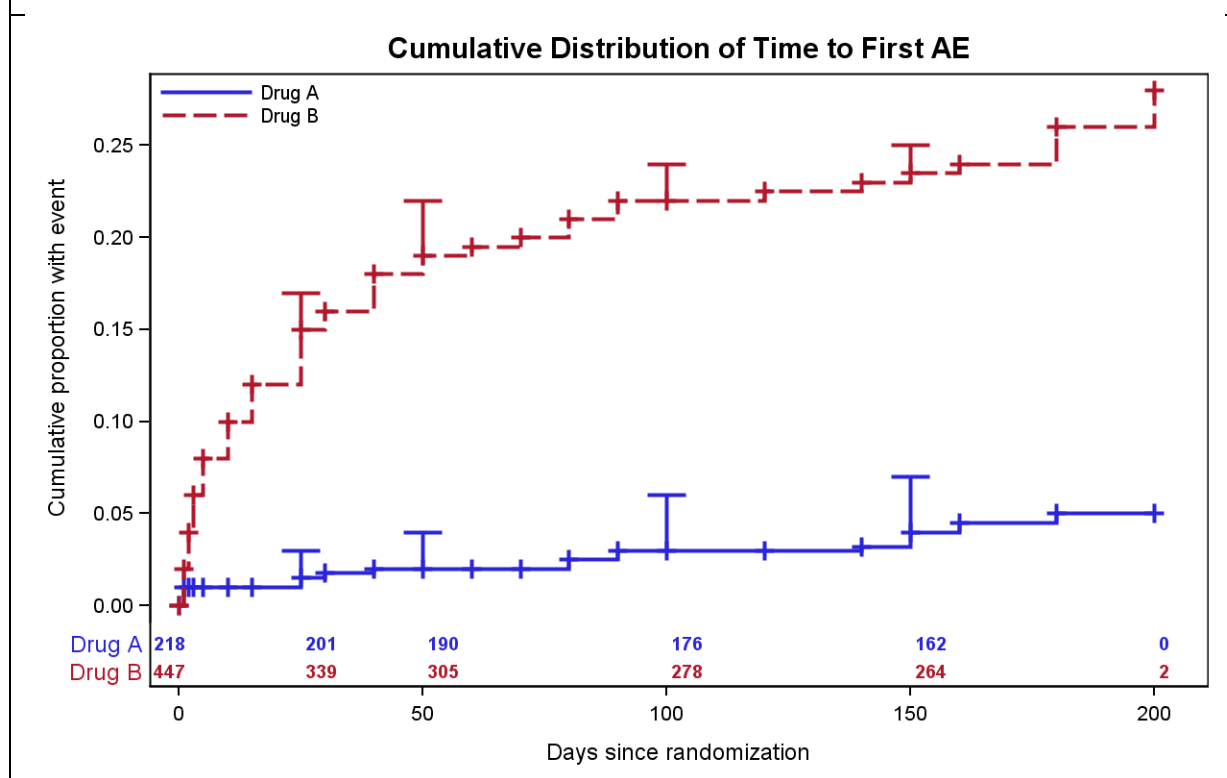


Figure 10B – Same graph as above created using GTL and simulated data

```

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 10C – SAS program using GTL

The graph shown in Figure 10B is created using a combination of multiple plot statements. The main features of the code in Figure 10C are as follows:

- This plot is created using GTL, using one LAYOUT OVERLAY container
- Use one STEPLOT x=day y=drugA to plot the data for drug A, including error bars.
- Use a SCATTERPLOT x=day y=drugA to see the points on the curve.
- Overlay another STEPLOT x=day y=drugB to plot the data for drug B, including error bars.
- Use a SCATTERPLOT x=day y=drugB to see the points on the curve.
- Set the visual attributes for drug A plots to GRAPHDATA1 and drug B plots to GRAPHDATA2.
- Name these plots A and B, and include them in the DISCRETELEGEND statement.
- The Legend is positioned in the upper left corner of the graph.
- Include an INNERMARGIN in the overlay container at the bottom (by default).
- This creates a region inside the data area for inclusion of the BLOCKPLOT statements.
- Add two BLOCKPLOTS in the INNERMARGIN to display the “At Risk” values.
- The BLOCKPLOTS are assigned the GRAPHDATA1 and GRAPHDATA2 elements to match the STEP plots.

In this use case the X axis is of type “interval” and not “discrete”. The BLOCKPLOT creates blocks for all values of X where the BLOCK variable has the same or missing value. In this case, the blocks start at the values in the data. Since the “At Risk” value is drawn in the block, it gets offset to the right of the marker in the Step Plot.

The graph in Figure 11B shows some techniques that can be used here to get better alignment between the data points and the “At Risk” values.

**Example 11: Hazard functions for AEs of special interest (with SEs).**

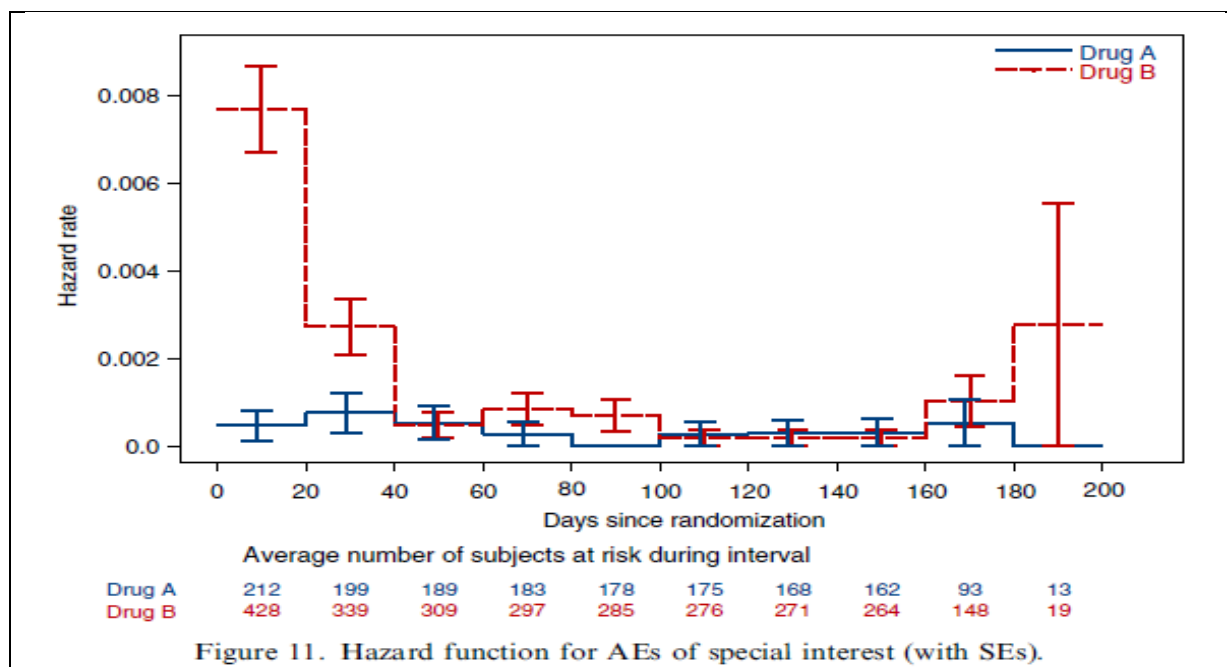
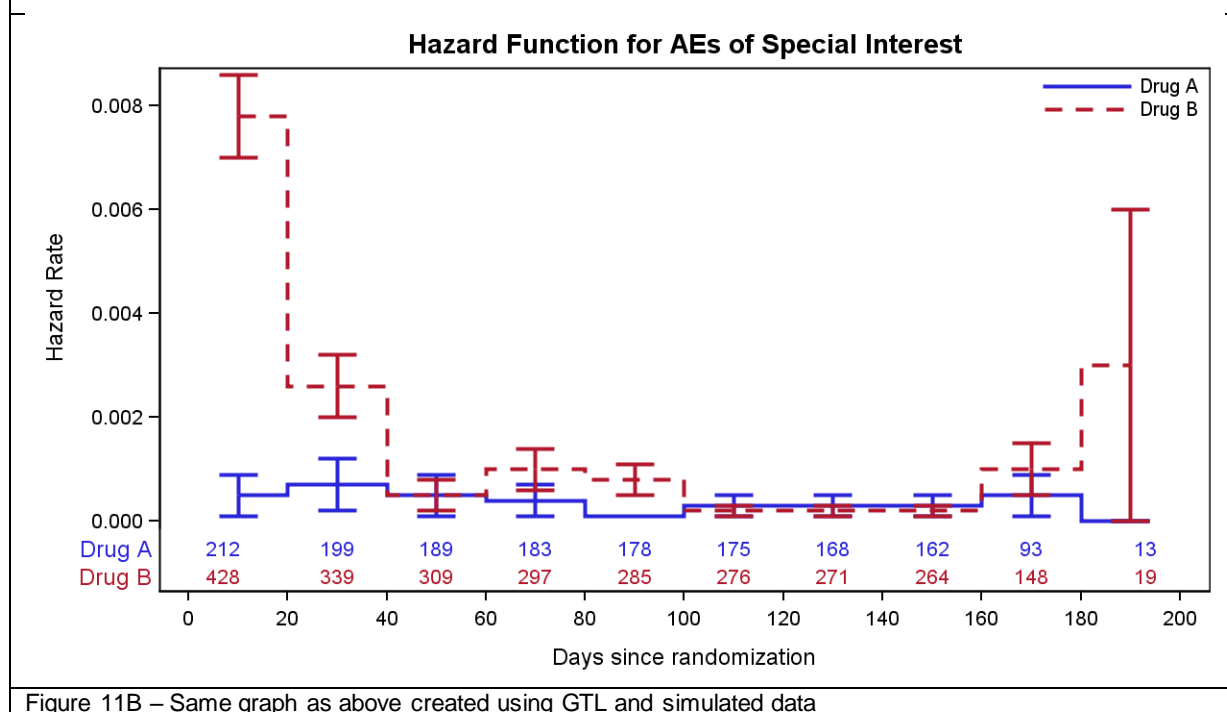


Figure 11A – From “Graphical Approaches to the Analysis of Safety Data from Clinical Trials”. Amit, et. al.



```

proc template;
define statgraph safety.safety11;
beginningraph ;
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
                 lineattrs=graphdata1(pattern=solid thickness=2)
                 errorbarattrs=graphdata1(pattern=solid thickness=2) name='a';
stepplot x=day y=B / errorupper=BHIGH errorlower=BLow justify=center
                 lineattrs=graphdata2(pattern=dash thickness=2)
                 errorbarattrs=graphdata2(pattern=solid thickness=2) name='b';
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;

```

Figure 11C – SAS program using GTL

The graph shown in Figure 11B is created by using a combination of multiple plot statements. The main features of the code shown in Figure 11C are as follows:

- This plot is created using GTL, using one LAYOUT OVERLAY container
- Use one STEPLOT x=day y=A to plot the data for drug A, including error bars.
- Overlay another STEPLOT x=day y=B to plot the data for drug B, including error bars.
- Set the visual attributes for 1<sup>st</sup> STEP plot to GRAPHDATA1 and 2<sup>nd</sup> STEP plot to GRAPHDATA2.
- Name these plots A and B, and include them in the DISCRETELEGEND statement.
- The Legend is positioned in the upper right corner of the graph.
- Include an INNERMARGIN in the overlay container at the bottom (by default).
- This creates a region inside the data area for inclusion of the BLOCKPLOT statements.
- Add two BLOCKPLOTS in the INNERMARGIN to display the “At Risk” values.
- The BLOCKPLOTS are assigned the GRAPHDATA1 and GRAPHDATA2 elements to match the STEP plots.

In this use case the X axis is of type “interval” and not “discrete”. The BLOCKPLOT creates blocks for all values of X where the BLOCK variable has the same or missing value. In this case, the blocks start at the values in the data. Since the “At Risk” value is drawn in the block, it gets offset to the right of the marker in the Step Plot.

To overcome this, we have shifted the X value for the BLOCKPLOT to the left by ½ the data interval by using BLOCKPLOT X=eval(day-10). Now, the blocks are effectively centered under each data value.

An alternative way is to set the X axis TYPE=DISCRETE. This forces the blocks to be centered at each of the data values. The result here is OK, except the axis values are now the exact ones in the data, and not from 0-200 by 20. Also the first and last intervals in the data are 10, while the other intervals are 20. However, setting the axis TYPE=DISCRETE makes all intervals equal, which is not desirable here.

## Example 12: Adverse Event Timeline

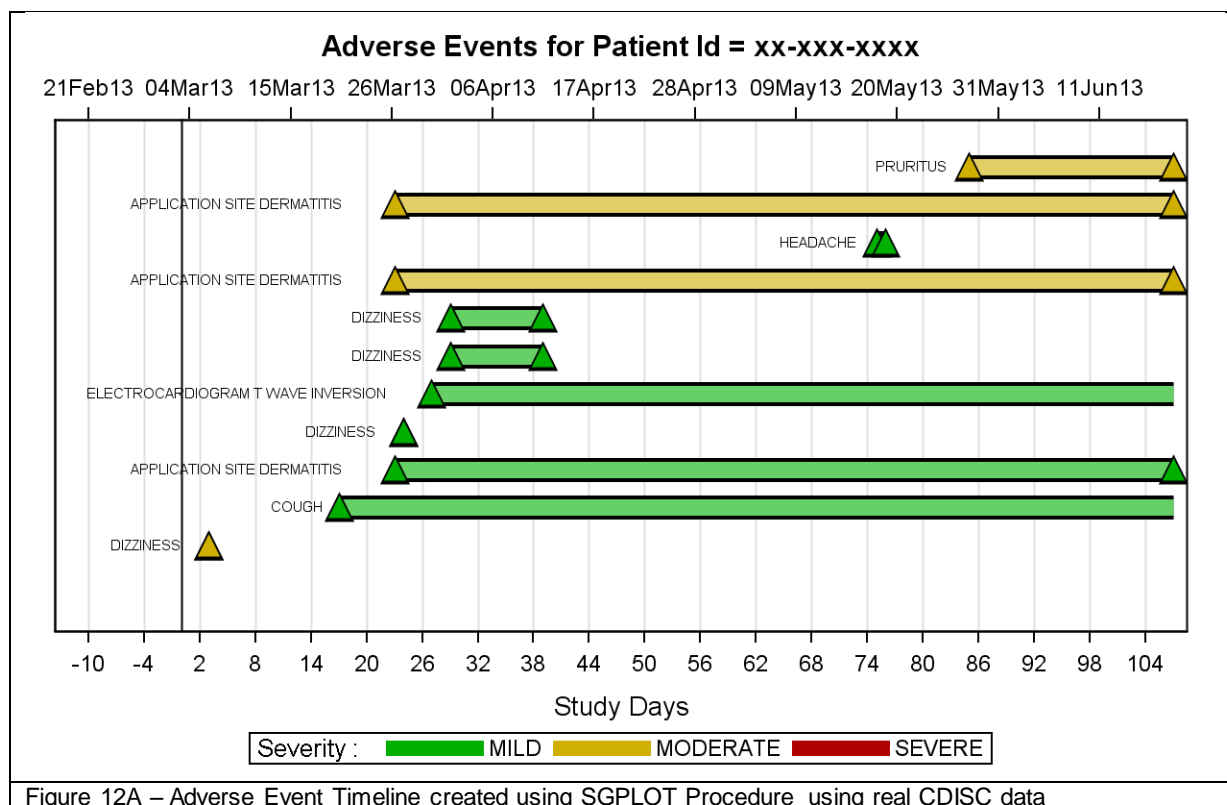


Figure 12A – Adverse Event Timeline created using SGPLOT Procedure using real CDISC data

```

title "Adverse Events for Patient Id = xx-xxx-xxxx";
proc sgplot data=ae noautolegend nocycleattrs;
  reffline 0 / axis=x lineattrs=(thickness=1 color=black);
  vector x=xs y=ys / xorigin=-110 yorigin = -110 group=aesev /*--Used to populate the Legend--*/
  lineattrs=(thickness=9px pattern=solid) name='sev';
  vector x=lastday y=y / xorigin=aestdy yorigin=y noarrowheads /*--Used to draw the black borders--*/
  lineattrs=(thickness=13px pattern=solid color=black) group=aesev;
  vector x=lastday y=y / xorigin=aestdy yorigin=y noarrowheads /*--Used to clear the inner region--*/
  lineattrs=(thickness=9px pattern=solid color=white) group=aesev;
  vector x=lastday y=y / xorigin=aestdy yorigin=y noarrowheads /*--Used to draw the event interval--*/
  lineattrs=(thickness=9px pattern=solid) transparency=0.4 group=aesev;

  scatter x=aestdy y=y / markerattrs=graphdatadefault(size=15px symbol=trianglefilled color=black);
  scatter x=aestdy y=y / markerattrs=(size=11px symbol=trianglefilled) group=aesev;
  scatter x=aeendy y=y / markerattrs=graphdatadefault(size=15px symbol=trianglefilled color=black);
  scatter x=aeendy y=y / markerattrs=(size=11px symbol=trianglefilled) group=aesev;
  scatter x=xc y=y / markerchar=aedecod markercharattrs=(size=6);
  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=0.02 values=(&minday2 to &maxday by 2);
  x2axis notimesplit display=(nolabel) tickvalueformat=date7. offsetmin=0.03 offsetmax=0.02
    values=(&mindate2 to &maxdate);
  keylegend 'sev' / title='Severity .';
run;

```

Figure 12B – SAS program using SGPLOT Procedure

Figure 12A shows an “Adverse Event Timeline” graph that shows the occurrence and duration of adverse events for a patient in the group by severity over time. Each event is labeled with a marker for start and end. The dictionary term for the event is also displayed. A Legend shows the severity levels.

The data for this graph is in the standard CDISC form. Some of the key features of the SGPLOT procedure used in the code shown in Figure 12B are as follows:

- The main graphical plot element used here is the VECTOR plot statement to draw the bar that represents the duration of the adverse event.
- Really, only one VECTOR plot statement is needed, but we have used 4 to get a border effect, and to blend the color to a softer shade.
- Also, to ensure we get all severity levels in the legend, the first VECTORPLOT uses columns that have all three values. This vector plot is used in the legend, but plotted off the graph region.
- SCATTER plot statements are used to show the start and end event marker. Again, multiple statements are used to get a marker with a border.
- Most the plots statements are plotted with X=study day and associated with the bottom “X” axis.
- One plot is plotted with X=study date, and associated with the top “X2” axis.
- X axis and X2 axis min and max values are set to ensure the X and X2 axis are properly aligned.

### Example 13: Lipid Profile over Time

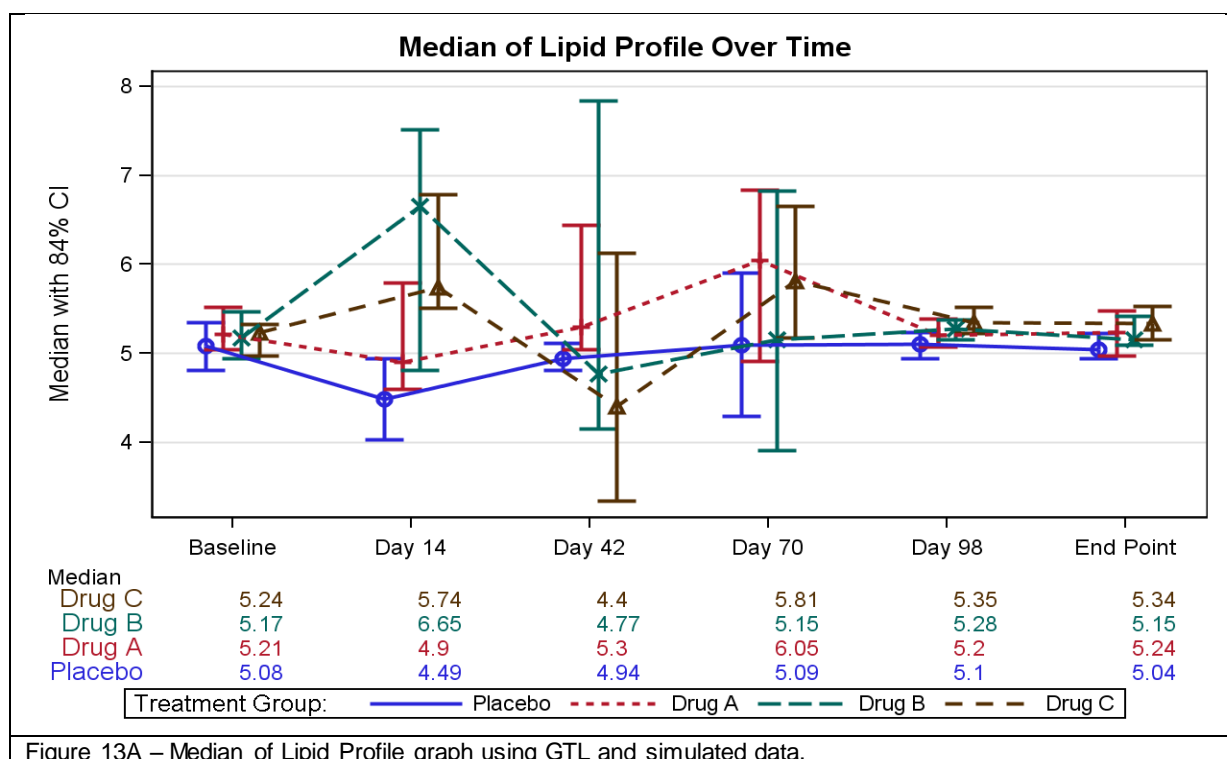


Figure 13A shows the graph for Median of Lipid Profile over time by Treatment Group. The main features of the graph are as follows:

- The graph is built using GTL, with a LAYOUT LATTICE of six rows.
- The first row includes the main graph using SERIESPLOTS and SCATTERPLOTS
- The 2<sup>nd</sup> row is used to display the term “Median”.
- The last 4 rows are used to include the external table showing the median values by day for each treatment group.
- The X axis has interval data, with a user defined format for the labels on the axis.
- Normally, each plot for the same value on the X axis would draw the marker directly in line with the tick value on the X axis.
- In this graph, we have “jittered” the X values a bit by using the X=eval(xc-delta) expression. The value of delta is set differently for each plot to create the effect. The same is done for the series plots.

```

proc template;
define statgraph lipid_profile;
  dynamic title;
  begingraph / designwidth=8in designheight=4.5in;
    entrytitle "Median of Lipid Profile over Time";

    layout lattice / columndatarange=union rowweights=(0.8 .04 .04 .04 .04 .04);

    layout overlay / yaxisopts=(label='Median with 84% CI' griddisplay=on)
      xaxisopts=(offsetmin=0.05 offsetmax=0.05 display=(line ticks tickvalues)
        linearopts=(tickvaluelist=(1 2 3 4 5 6)
          tickdisplaylist=('Baseline' 'Day 14' 'Day 42' 'Day 70' 'Day 98' 'End Point')));
      scatterplot x=eval(xc-0.15) y=p_med / yerrorlower=p_lcl yerrorupper=p_ucl name='ps'
        markerattrs=graphdata1(size=9px weight=bold)
        errorbarattrs=graphdata1(pattern=solid thickness=2);
      scatterplot x=eval(xc-0.05) y=a_med / yerrorlower=a_lcl yerrorupper=a_ucl name='as'
        markerattrs=graphdata2(size=9px weight=bold)
        errorbarattrs=graphdata2(pattern=solid thickness=2);
      scatterplot x=eval(xc+0.05) y=b_med / yerrorlower=b_lcl yerrorupper=b_ucl name='bs'
        markerattrs=graphdata3(size=9px weight=bold)
        errorbarattrs=graphdata3(pattern=solid thickness=2);
      scatterplot x=eval(xc+0.15) y=c_med / yerrorlower=c_lcl yerrorupper=c_ucl name='cs'
        markerattrs=graphdata4(size=9px weight=bold)
        errorbarattrs=graphdata4(pattern=solid thickness=2);
      seriesplot x=eval(xc-0.15) y=p_med / lineattrs=graphdata1(pattern=solid thickness=2px) name='pl';
      seriesplot x=eval(xc-0.05) y=a_med / lineattrs=graphdata2(pattern=shortdash thickness=2px)
        name='al';
      seriesplot x=eval(xc+0.05) y=b_med / lineattrs=graphdata3(pattern=mediumdash thickness=2px)
        name='bl';
      seriesplot x=eval(xc+0.15) y=c_med / lineattrs=graphdata4(pattern=dash thickness=2px) name='cl';
    endlayout;

    layout overlay;
      entry halign=left 'Median';
    endlayout;

    blockplot x=xc block=c_med / display=(values label) valuehalign=center label='Drug C'
      repeatedvalues=true valueattrs=graphdata4 labelattrs=graphdata4;

    blockplot x=xc block=b_med / display=(values label) valuehalign=center label='Drug B'
      repeatedvalues=true valueattrs=graphdata3 labelattrs=graphdata3;

    blockplot x=xc block=a_med / display=(values label) valuehalign=center label='Drug A'
      repeatedvalues=true valueattrs=graphdata2 labelattrs=graphdata2;

    blockplot x=xc block=p_med / display=(values label) valuehalign=center label='Placebo'
      repeatedvalues=true valueattrs=graphdata1 labelattrs=graphdata1;

    sidebar / spacefill=false;
      discretelegend 'pl' 'al' 'bl' 'cl' / title='Treatment Group: ' across=4;
    endsidebar;

  endlayout;

endgraph;
end;
run;

```

Figure 13B – SAS program using SGPLOT Procedure

## Example 14: Immunology profile by Treatment

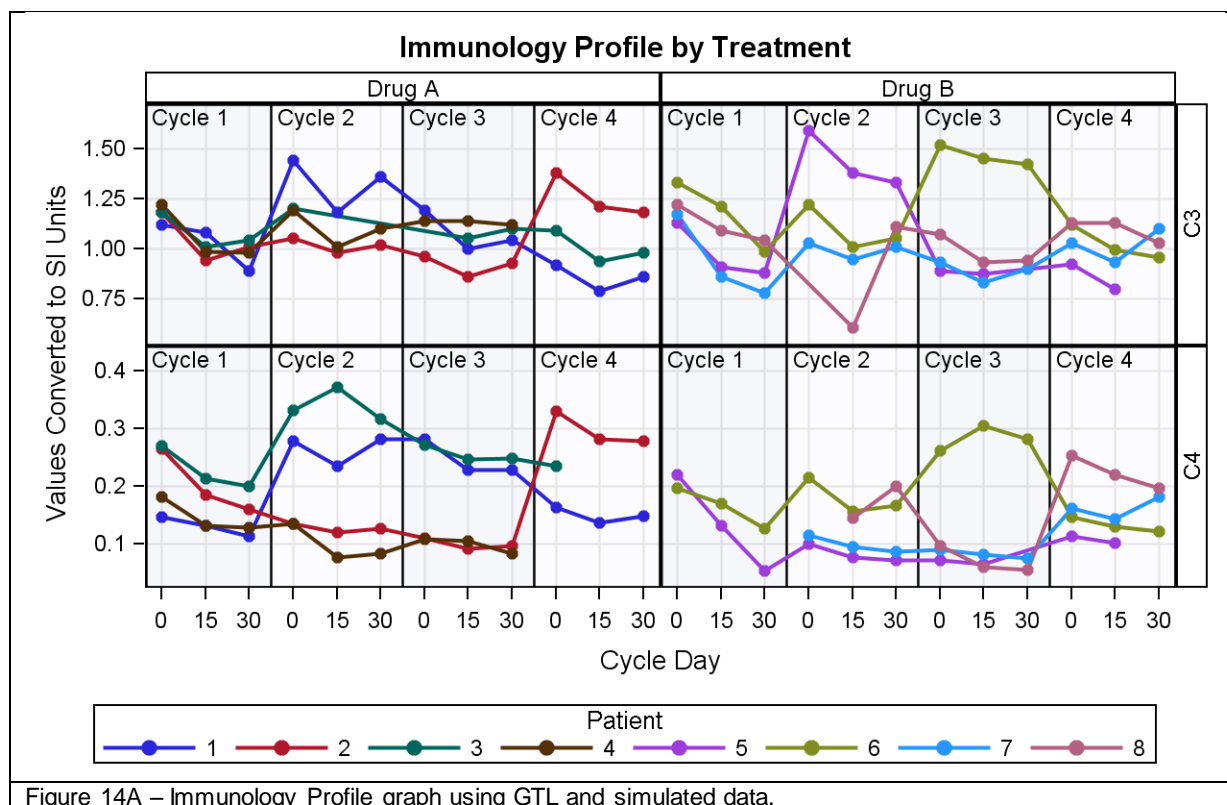


Figure 14A shows Immunology panel over time by cycle and treatment. The main features of the graph are as follows:

- This uses a LAYOUT DATALATTICE to create the classification panel of treatment by lab parameter.
- Each cell contains a Series plot of the lab values over time.
- A BLOCKPLOT is added to show the different cycles over the time duration.
- A DISCRETELEGEND is added in the bottom of the GRIDDED layout.

```
proc template;
define statgraph immune;
beginning / designwidth=6in designheight=4.5in;
entrytitle 'Immunology Profile by Treatment';
layout gridded;
layout datalattice columnvar=trt rowvar=lbparm / headerlabeldisplay=value includemissingclass=false
columnaxisopts=(label="Cycle Day" griddisplay=on type=discrete offsetmin = .03 offsetmax = .03
discreteopts= (tickdisplaylist= ("0" "15" "30" "0" "15" "30" "0" "15" "30" "0" "15" "30"))) )
rowaxisopts=(offsetmax=.1 label="Values Converted to SI Units " griddisplay=on)
rowdatarange=union;
layout prototype;
blockplot x=xval block=cyc / datatransparency = .75
display=(outline fill values ) valuealign=top labelposition=top name="block" filltype=alternate;
seriesplot x=xval y=sival / group=pt index=pt name='a'
display=all lineattrs=(pattern=1 thickness=2) markerattrs=(symbol=circlefilled size=8);
endlayout;
endlayout;
discretelegend 'a' / title ='Patient' across=8;
endlayout;
endgraph;
end;
```

Figure 14B – SAS program using GTL



## CONCLUSION

Analysis and interpretation of the data for clinical graphs is improved when the data and the statistics can be viewed together. The graphs shown above are commonly used in the Health and Life Sciences domain. These graphs can be easily created using SG procedures and Graph Template Language (GTL). Most of the graphs shown above can be created using no more than a page of code. Even where the program is a few lines long, the structure of the graph is very clear and easy to understand. Most of the work is done by the statements supported in the syntax, and no special coding or annotation is required.

## REFERENCES

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

## RECOMMENDED READING

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

Heath, Dan. 2007. "New SAS/GRAPH Procedures for Creating Statistical Graphics in Data Analysis." *Proceedings of the SAS Global Forum*. 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  
SAS Campus Dr  
Cary, NC 27512  
(919) 531-6753  
Fax (919) 677-4444  
[Sanjay.Matange@sas.com](mailto:Sanjay.Matange@sas.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.