PharmaSUG 2018 - Paper DV-23

A tale of a tail or how to make your graph more informative

Iryna Kotenko, Intego Group/Experis Clinical

ABSTRACT

The nature of the data undoubtedly determines its visual shape and quite often you have to look for workarounds to make a better visual representation of the data. This article discusses the features of PROC TEMPLATE & PROC SGRENDER for efficient data scaling and gives number of practical examples.

INTRODUCTION

A primary goal of data visualization is to communicate information clearly and efficiently. But the data you work with can always play tick on you; at first glance strait forward graph become absolutely none-informative and requires an individual approach and workarounds to the data pre-processing and presentation. Fortunately, modern clinical programmers have a huge arsenal of visualization tools, which can help realize even the most daring ideas. In this article, a visualization problem that was encountered in my working routine will be described along with three methods of its solving using standard SAS® procedures PROC TEMPLATE & PROC SGRENDER.

BPROC TEMPLATE & PROC SGRENDER OVERVIEW

The SGRENDER procedure produces that graphical output from templates that are created with the Graph Template Language (GTL). The GTL is a comprehensive language for creating statistical graphics, which can be used to create customized layouts and graphs that are beyond the scope of the Statistical Graphics procedures[1]. Programmer can create user-defined template through PROC TEMPLATE and standardized GTL condition statements. For detailed description of both procedures please refers to the section Recommended Readings at the end of the article.

BUSINESS CASE FOR SCALING A GRAPH

The analysis of laboratory findings takes key point in the analysis of clinical trial data. Blood tests are often used in health care to determine physiological and biochemical states, such as disease, mineral content, pharmaceutical drug effectiveness, and organ function. In laboratory blood assessments different diseases indicates themselves differently in the results of a blood test. For example, under some clinical conditions, measurements can show a deviation from the norm in tens or even hundreds of times. It is natural that the treatment is aimed return blood counts to normal range. What then can we see when we visualize such a data? Let's look at Figures 1a where the individual's plot of the blood test A assessments over time are shown and Figure 1b the plot of the mean value for the group at visit alongside with confidence limits to mean are shown over time. The PROC SGPLOT has been used to create these two plots:

```
/********************** Standard Graphs **********************************/
/************************** Figure 1a *************************************/
proc sgplot data=res;
  series x=visitnum y=aval /  group=usubjid ;
  xaxis values=(1 to 20 by 1) type=linear label="Study Visits";
  yaxis label="Laboratory Test A (IU/L)";
  keylegend ;
run;
```

```
/************************* Figure 1b ************************************/
proc sgplot data=resm noautolegend;
  scatter  x=visitnum y=mean / yerrorupper=high yerrorlower=low ;
  series x=visitnum y=mean2 ;
  xaxis values=(1 to 20 by 1) type=linear label="Study Visits";
  yaxis label="Laboratory Test A (IU/L)";
run;
```

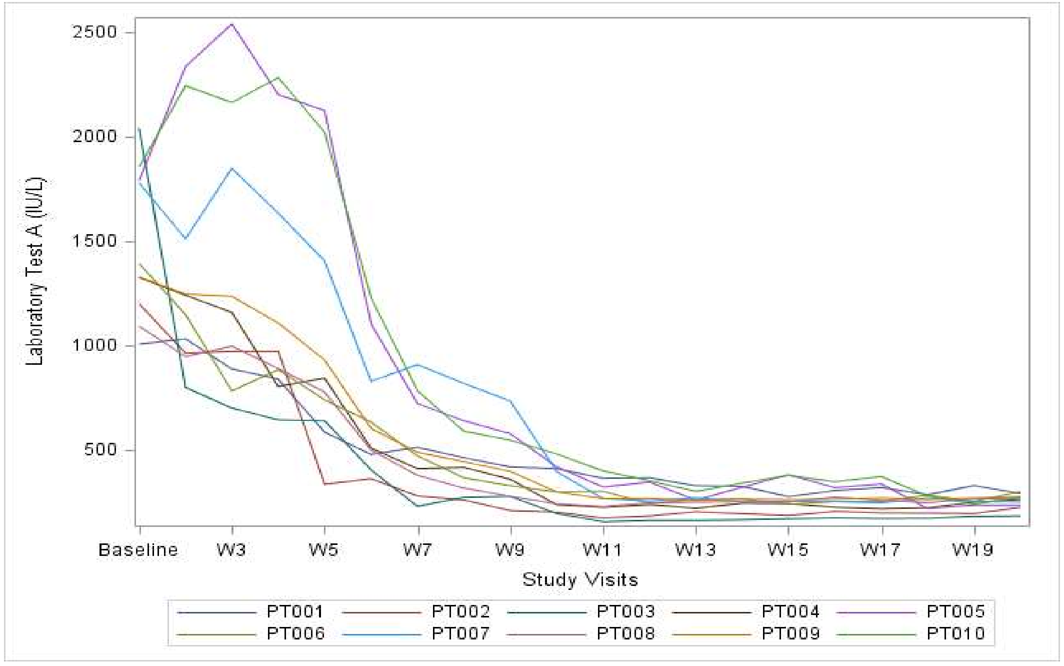

Figure 1a is a sample figure of individual laboratory values over time.

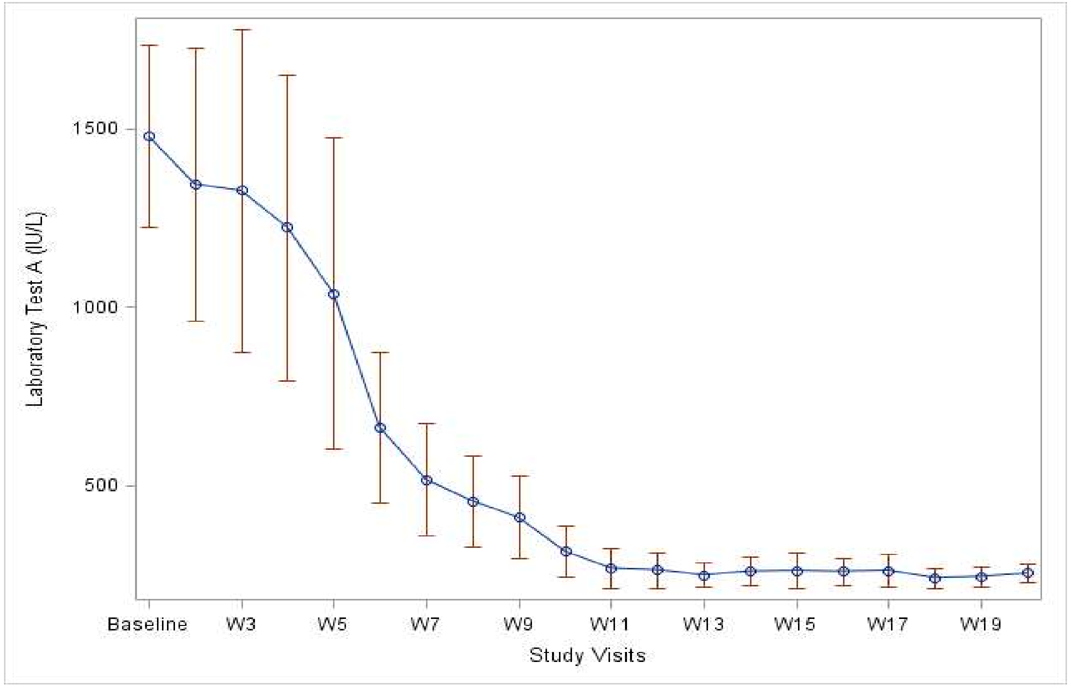

Figure 1b is a sample figure of mean value along with confidence intervals to mean for a laboratory test over time.

As it can be seen at the beginning of treatment, blood test A level shows extremely high results for all patients and, for some patients, even approaches value of 3000 units per liter. By the middle of the treatment the blood test results decrease and fluctuate within values of 100 to 500 units. The trends of the first part of the treatment are well traced on the Figure 1a and Figure 2a, although it is completely unclear what is happening in the "tail" of the graph representing in the second period of treatment.

A natural question arises: is it possible to find an approach in which both parts of the graph give us maximum information? After some consideration, three methods of solving this problem were proposed. Each of these methods will be explained in more detail in the sections below.

SOLUTION 1: TWO-SCALE PLOT

The first solution implies the division of a graph into two cells that will have different Y axis scaling. First part of the graph will have the same scale as initial plot (from 0 to 3000 for individual's plot and 0 to 2000 for means plot) and the second part Y axis scale will be changed to 100 to 450 and 200 to 400 for individual's and means plots respectively.

This solution uses the following features of PROC TEMPLATE:

LAYOUT LATTICE - defines a multi-cell grid of graphs like it shown on the figure below for the case 2 rows by 2 columns, but user can setup n by m cells for the graph:

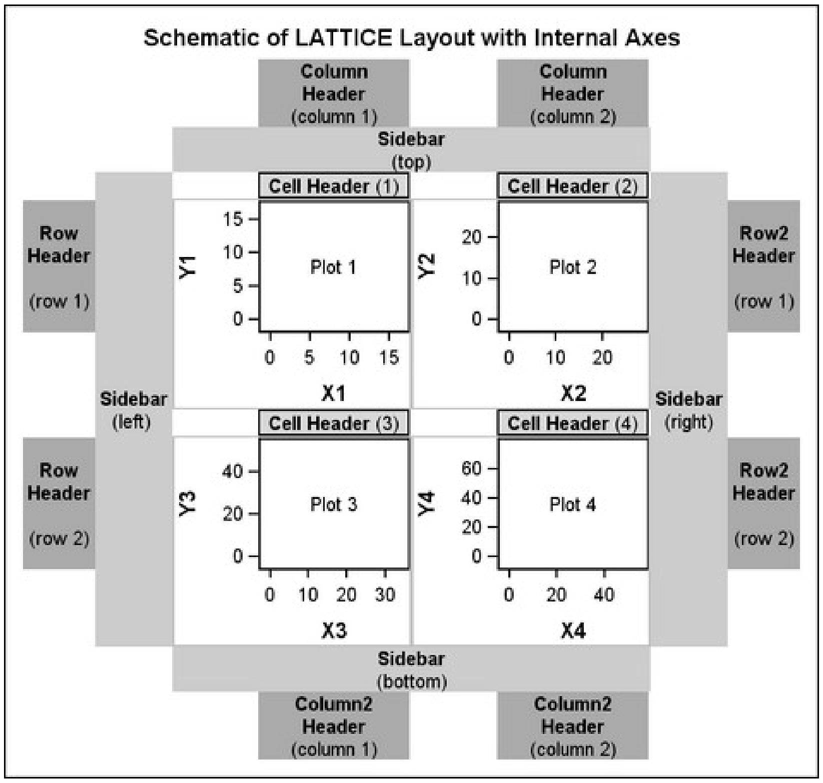

Figure 2 Schema of LATTICE layout for 2x2 scenario

LAYOUT OVERLAY allows to overlay the results of the statements that are contained in the layout block

The following SAS code produces the two-scale plot for individual's results over time:

```
/***********************    Example 1   **********************************/
/************************** Figure 3a ************************************/
proc template;
 define statgraph broken_pt1;
 begingraph;
    entrytitle 'Solution 1 Individual Plot';
    layout lattice / columns=2 columnweights=(.5 .5)
                 columndatarange=union columngutter=9;
     layout overlay / xaxisopts=(display=(tickvalues ticks)
                             linearopts=(viewmin=1 viewmax=10
                             tickvaluesequence=(start=1 end=10
                             increment=1)))
                 yaxisopts=(display=(tickvalues ticks)
                             linearopts=(viewmin=0 viewmax=3000
                             tickvaluesequence=(start=0 end=3000
                             increment=500)));
       seriesplot x=visitnum y=aval / group=usubjid index=usubjid name="pt";
     endlayout;
     layout overlay / xaxisopts=(display=(tickvalues ticks)
                             linearopts=(viewmin=11 viewmax=20
                             tickvaluesequence=(start=11 end=20
                             increment=1)))
                 yaxisopts=(display=(tickvalues ticks) offsetmax=0
                             linearopts=(viewmin=100 viewmax=450
                             tickvaluesequence=(start=100 end=450
                             increment=50)));
      seriesplot x=visitnum y=aval/ group=usubjid index=usubjid name="ptt";
     endlayout;
     sidebar / align=left;
        entry 'Laboratory Test A (IU/L)' / rotate=90;
       endsidebar;
     sidebar / align=bottom;
        discretelegend "pt" / title="Patients:" displayclipped=true
                             autoalign=(BOTTOM) location=OUTSIDE border=off;
       endsidebar;
   endlayout;
 endgraph;
end;
run;

proc sgrender data=res template=broken_pt1;
run;
```

The same approach has been implemented to means plot and the code can be found in the appendix to this article. The result graphs (Figures 3a and 3b) are way more informative compering to Figures 1. The user might break the graph into that many pieces it is necessary for clear visualization of the results.

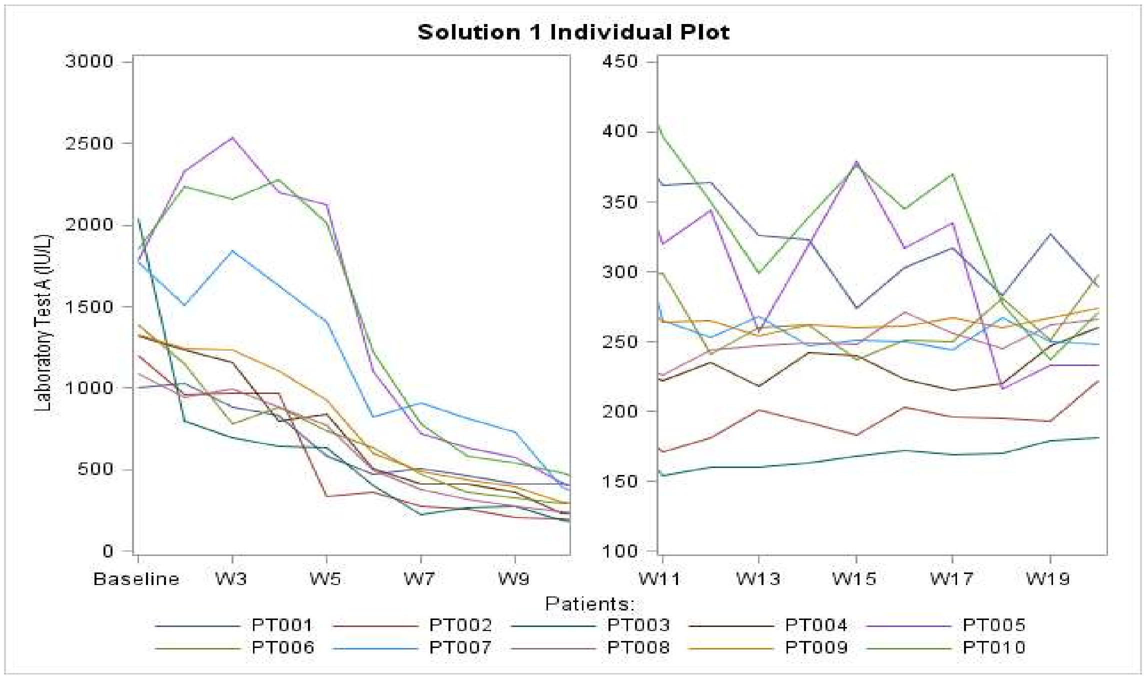

Figure 3a Individual laboratory values over time on two-scale plot

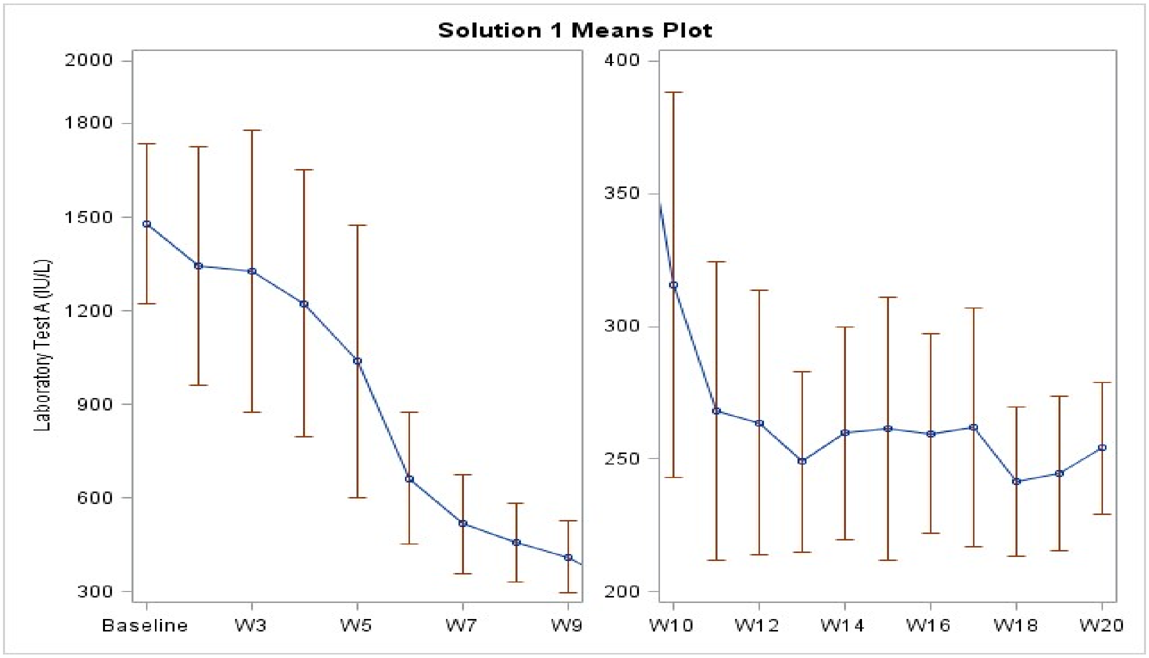

Figure 3b Mean value along with confidence intervals to mean for a laboratory test over time on two-scale plot.

SOLUTION 2: BREAKING AXIS

The second solution is built around breaking vertical Y axis into two pieces that have different increment for the scaling. For the top of the graph (ranges from 500 to 3000 for individual plot and 500 to 2000 for means plot) the increment is set to 500 although for the bottom it is set to 100 and 50 for individual and means plots respectively. The SAS code that produces the plot for individual's blood test results over time is shown below:

```
proc template;
   define statgraph broken_pt2;
   begingraph;
      Entrytitle "Solution 2 Broken Axis Individual Plot";
      layout lattice / rowweights=(.48 .04 .48)
                       columndatarange=union
                       rowdatarange=union;
         rowaxes;
            rowaxis / display=(line ticks tickvalues);
            rowaxis;
            rowaxis / display=(line ticks tickvalues);
         endrowaxes;
         layout overlay / xaxisopts=(display=none)
                          walldisplay=(fill);
            seriesplot x=visitnum y=aval / group=usubjid index=usubjid;
         endlayout;
         layout overlay / pad=0;
            entry halign=left
                  textattrs=(size=10pt) "W" / rotate=90 pad=0;
         endlayout;
         layout overlay / walldisplay=(fill) xaxisopts=(label="Visit");
            seriesplot x=visitnum y=avalb/ group=usubjid
                                          index=usubjid
                                          name="pt";
         endlayout;
         sidebar / align=left;
              entry 'Laboratory Test A (IU/L)' / rotate=90;
           endsidebar;
         sidebar / align=bottom;
              discretelegend "pt" / title="Patients:"
                                    displayclipped=true
                                    autoalign=(BOTTOM)
                                    location=OUTSIDE
                                    border=off;
           endsidebar;
      endlayout;
   endgraph;
   end;
run;
proc sgrender data=res2 template=broken_pt2;
run;
```

This approach is similar to solution 1 and uses the same methods LAYOUT LATTICE and LAYOUT OVERLAY however now that entire space of the graph is splitted into 3 parts: first and third encapsulate 96% of the graph and the second one narrow part is playing a role of a break. The results you can see on the figures below. Although this approach presents the data in more informative effective way the results doesn't look nice and tide and require some preprocessing to the data. SAS code for preprocessing and the code for means plot over time with break in vertical axis could be found below in the appendix section.

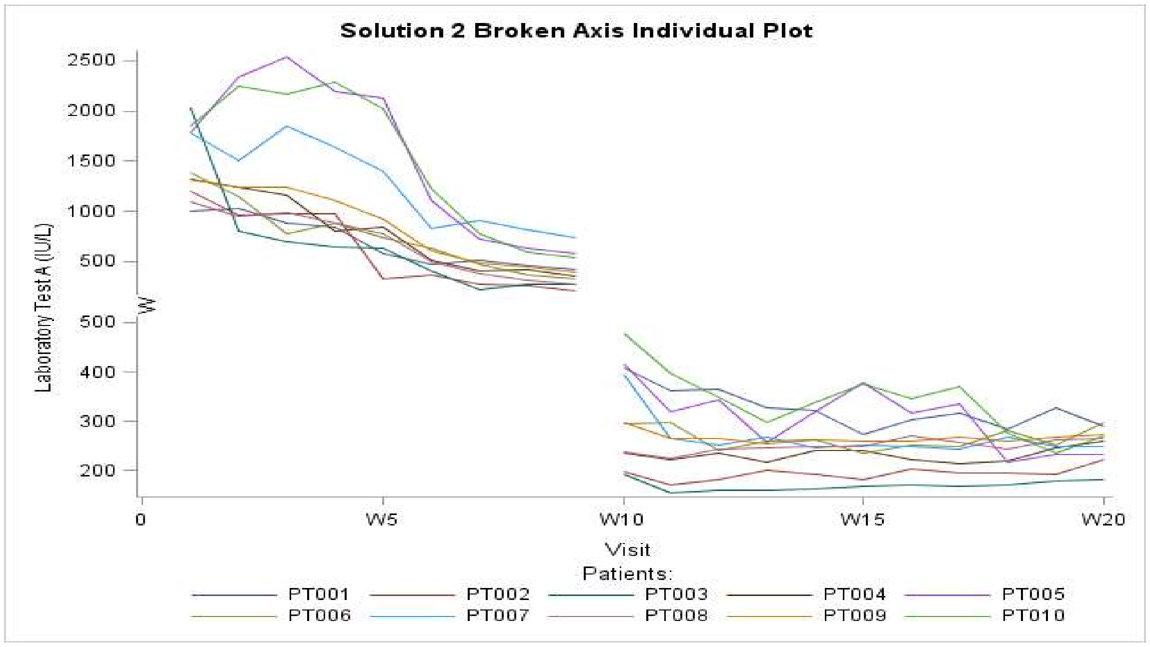

Figure 4a Individual laboratory values over time plot with break in vertical axes

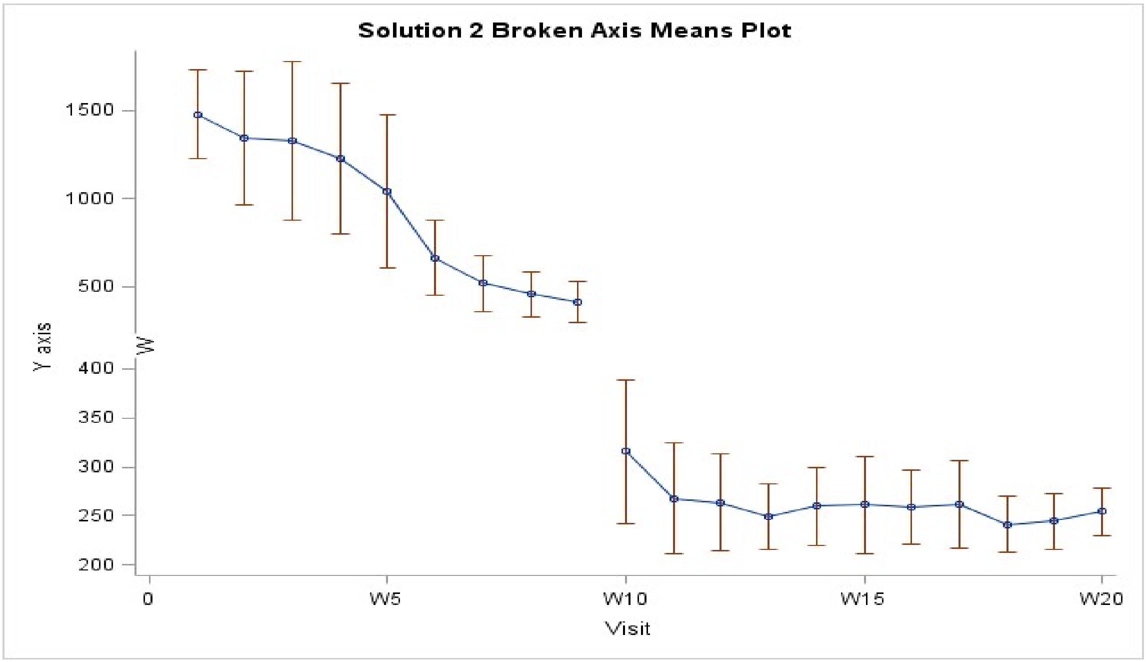

Figure 4b Mean value along with confidence intervals to mean for a laboratory test over time plot with break in vertical axes

SOLUTION 2: A GRAPH INSIDE A GRAPH

This solution was inspired by many interactive tools that allows you to enlarge certain area of a picture in an overlaying panel as if you use a magnifying glass. Exploring given example it can be seen that left part of Figure 1a,b are pretty occupied with visual information although the right part has a lot of white space on the top. Thus the decision was made to overlay this white space with a smaller graph that would show the ”tail” of the Figure 1a,b graph in different y-axis scale.

This solution uses the following feature of PROC TEMPLATE:

LAYOUT GRIDDED – that offers the best way to nest a table of information or a graph inside another layout. HALIGN, VALIGN, WIDTH and HEIGHT LAYOUT GRIDDED allows to place subgraph to the area with white space and use it in efficient manner. So, the SAS code for the solution 3 looks like the following:

```
proc template;
   define statgraph broken_pt3;
   begingraph;
      entrytitle 'Solution 3 Individual Plot';
      layout overlay;
         layout overlay / xaxisopts=(label="Visits"
                                     display=(tickvalues ticks label)
                                     linearopts=(viewmin=1 viewmax=21
                                     tickvaluesequence=(start=1 end=20
                                     increment=1)))
                          yaxisopts=(label="Laboratory Test A (IU/L)"
                                     display=(tickvalues ticks label)
                                     linearopts=(viewmin=0 viewmax=3000
                                     tickvaluesequence=(start=0 end=3000
                                     increment=500)));
            seriesplot x=visitnum y=aval / group=usubjid index=usubjid
                                           name="pt";
             discretelegend "pt" / title="Patients:" displayclipped=true
                                   autoalign=(BOTTOM) location=OUTSIDE
                                   border=off;
         endlayout;
         layout gridded / halign=right valign=top width=330px height=220px ;
         layout overlay / xaxisopts=(display=(tickvalues ticks)
                                     linearopts=(viewmin=11 viewmax=20
                                     tickvaluesequence=(start=11 end=21
                                     increment=1)))
                          yaxisopts=(display=(tickvalues ticks) offsetmax=0
                                     linearopts=(viewmin=100 viewmax=410
                                     tickvaluesequence=(start=100 end=400
                                     increment=100)));
            seriesplot x=visitnum y=aval/ group=usubjid index=usubjid
                                          name="pt";
         endlayout;
             endlayout;
      endlayout;
   endgraph;
end;
run;

proc sgrender data=res template=broken_pt2;
run;
```

The results for this solution are shown on the Figures 5a and 5b. Graphs look clear, informative, a bit unusual but presentable at the same time.

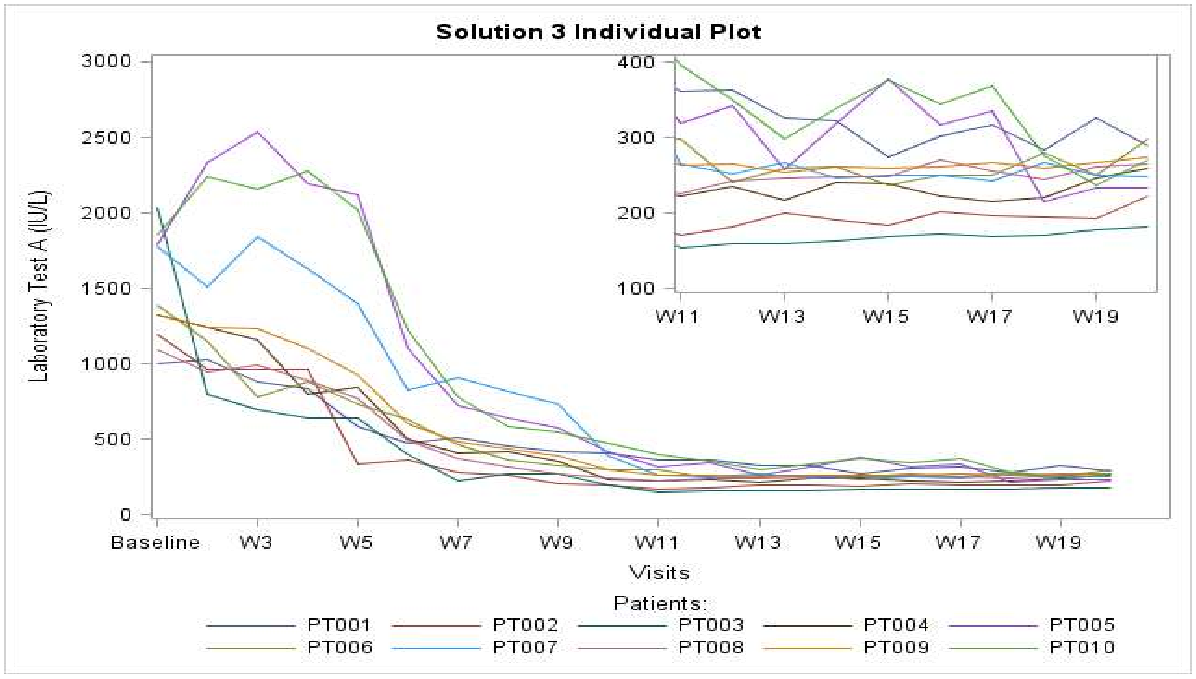

Figure 5a Individual laboratory values over time plot in style a graph inside a graph

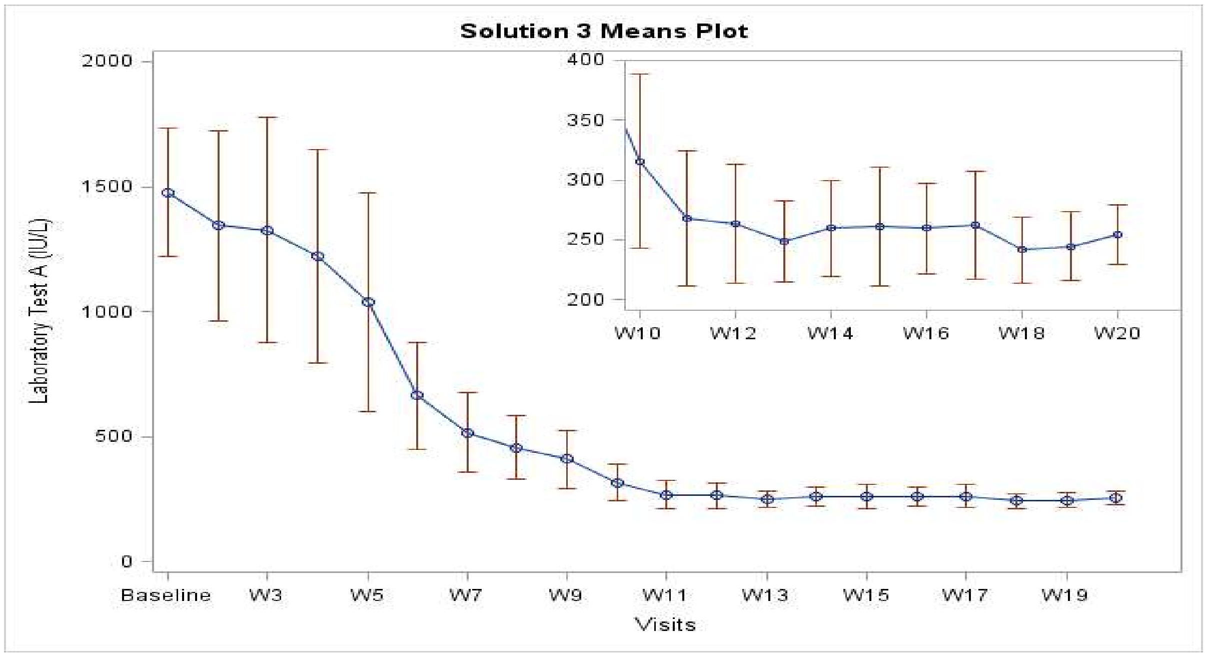

Figure 5b Mean value along with confidence intervals to mean for a laboratory test over time plot in style a graph inside a graph

CONCLUSION

In conclusion, it worth to say that it does not matter how unimaginable the visualization task looked at first sight, the GTL procedure can help to solved it easily in elegant way, moreover it usually can give more than one solution to this problem. The GTL procedure gives the programmer a powerful and at the same time easy-to-understand and distinguished tool for producing high-quality graphics.

RECOMMENDED READING

- *SAS® 9.4 Graph Template Language: User's Guide, Fifth Edition (http://documentation.sas.com/?docsetId=grstatug&docsetTarget=titlepage.htm&docsetVersion=9.4&locale=en)*
- *SAS Data Visualization Blog (https://blogs.sas.com/content/topic/data-visualization/)*
- *SAS Support Samples and Notes (https://support.sas.com/en/search.html?q=*%3A*&fq=siteArea%3A%22Samples%20%26%20SAS%20Notes%22&fq=siteAreaChild%3ASamples)*

CONTACT INFORMATION

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

Iryna Kotenko
Irina.Kotenko@intego-group.com

APPENDIX

```
/*************************************************************************/
/********************    Solution 1 Means Plot   *************************/
/*************************************************************************/
proc template;
 define statgraph broken_m1;
 begingraph;
   entrytitle 'Solution 1 Means Plot';
   layout lattice / columns=2 columnweights=(.5 .5) columndatarange=union
                    columngutter=9;
    layout overlay / xaxisopts=(display=(tickvalues ticks)
                               linearopts=(viewmin=1 viewmax=9
                               tickvaluesequence=(start=1 end=9
                               increment=1)))
                     yaxisopts=(display=(tickvalues ticks)
                               linearopts=(viewmin=300 viewmax=2000
                               tickvaluesequence=(start=300 end=2000
                               increment=300)));
       seriesplot x=visitnum y=mean2;
       scatterplot  x=visitnum y=mean / yerrorupper=high yerrorlower=low ;
     endlayout;
     layout overlay / xaxisopts=(display=(tickvalues ticks)
                                     linearopts=(viewmin=10 viewmax=20
                                     tickvaluesequence=(start=10 end=20
                                     increment=1)))
                           yaxisopts=(display=(tickvalues ticks) offsetmax=0
                                     linearopts=(viewmin=200 viewmax=400
                                     tickvaluesequence=(start=200 end=400
                                     increment=50)));
        seriesplot x=visitnum y=mean2;
        scatterplot  x=visitnum y=mean / yerrorupper=high yerrorlower=low ;
         endlayout;
         sidebar / align=left;
           entry 'Laboratory Test A (IU/L)' / rotate=90;
         endsidebar;
      endlayout;
   endgraph;
end;
run;
```

```
proc sgrender data=resm template=broken_m1;
run;

/*************************************************************************/
/********************          Solution 2        *************************/
/********************      Data Preprocessing    *************************/
/*************************************************************************/
```

If you about to break your vertical axes only in one place the following code should be performed.

```
data resm2;
  set resm;
  if visitnum>=10 then do; /*if the visit is after the break*/
    /*create new set of variables to be visualized*/
      meanb=mean;
      meanb2=mean2;
      lowb=low;
      highb=high;
    /*set original variables to missing for the graph part after the break*/
      call missing(mean, mean2, low,high);
  end;
run;

/*************************************************************************/
/********************    Solution 2 Means Plot   *************************/
/*************************************************************************/
proc template;
   define statgraph broken_m2;
   begingraph;
      Entrytitle "Solution 2 Broken Axis Means Plot";
      layout lattice / rowweights=(.48 .04 .48) columndatarange=union
                        rowdatarange=union;
         rowaxes;
            rowaxis / display=(line ticks tickvalues);
            rowaxis;
            rowaxis / display=(line ticks tickvalues);
         endrowaxes;
         layout overlay / xaxisopts=(display=none)walldisplay=(fill);
            seriesplot x=visitnum y=mean2;
            scatterplot x=visitnum y=mean / yerrorupper=high yerrorlower=low;
         endlayout;
         layout overlay / pad=0;
            entry halign=left textattrs=(size=10pt) "W" / rotate=90 pad=0;
         endlayout;
         layout overlay / walldisplay=(fill) xaxisopts=(label="Visit");
           seriesplot x=visitnum y=meanb2 ;
           scatterplot x=visitnum y=meanb/yerrorupper=highb yerrorlower=lowb;
         endlayout;
         sidebar / align=left;
            entry "Y axis" / rotate=90 textattrs=GraphLabelText;
         endsidebar;
      endlayout;
   endgraph;
   end;
run;

proc sgrender data=resm2 template=broken_m2;
run;
```

```
/*************************************************************************/
/********************    Solution 3 Means Plot    ************************/
/*************************************************************************/
proc template;
   define statgraph broken_m3;
   begingraph;
      entrytitle 'Solution 3 Means Plot';
      layout overlay;
         layout overlay / xaxisopts=(label="Visits"
                            display=(tickvalues ticks label)
                            linearopts=(viewmin=1 viewmax=21
                            tickvaluesequence=(start=1 end=20
                            increment=1)))
                          yaxisopts=(label="Laboratory Test A (IU/L)"
                            display=(tickvalues ticks label)
                            linearopts=(viewmin=0 viewmax=2000
                            tickvaluesequence=(start=0 end=3000
                            increment=500)));
           seriesplot x=visitnum y=mean2;
           scatterplot x=visitnum y=mean / yerrorupper=high yerrorlower=low;
         endlayout;
         layout gridded / halign=right valign=top width=330px height=220px ;
         layout overlay / xaxisopts=(display=(tickvalues ticks)
                             linearopts=(viewmin=10 viewmax=21
                             tickvaluesequence=(start=10 end=20
                             increment=1)))
                          yaxisopts=(display=(tickvalues ticks) offsetmax=0
                             linearopts=(viewmin=200 viewmax=400
                             tickvaluesequence=(start=200 end=400
                             increment=50)));
            seriesplot x=visitnum y=mean2;
            scatterplot x=visitnum y=mean / yerrorupper=high yerrorlower=low;
         endlayout;
        endlayout;
      endlayout;
   endgraph;
end;
run;

proc sgrender data=resm template=broken_m3;
run;
```