

I Am Legend for PharmaSUG

Kriss Harris, SAS Specialists Ltd, Hertfordshire, United Kingdom

ABSTRACT

Have you ever produced a legend on a plot that was taking up too much space, hence making the actual graph too small? Have you ever removed a legend because it was taking up too much space? Have you ever wanted to just produce a legend? Have you ever wondered that there must be a more efficient way of producing a legend then using the exact same legend on every BY variable of your output? This paper will demonstrate solutions to the above problems using Graph Template Language (GTL) in SAS® 9.2, in particularly using the SERIES, VECTOR and SCATTERPLOT statements.

INTRODUCTION

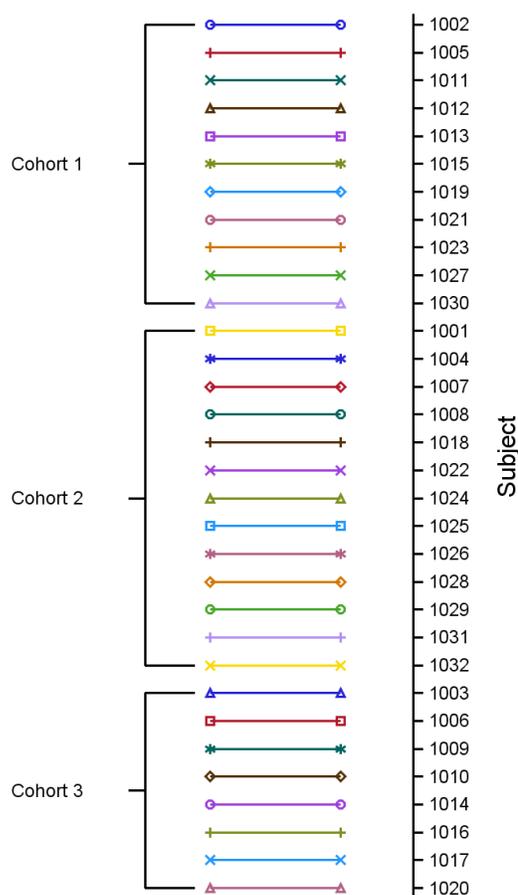
Who is your favorite legend? What qualities do you like about the legend? These are interesting questions, and most likely you have thought of a few legends now.

You are probably aware that legends have done extraordinary things and stand out from the crowd. When you view an output of a graph in SAS, typically the actual plot takes up most of the area on the output, and there is a small section at the bottom of the output that displays the legend. On this occasion the legend does not stand out on the output.

Similar to the stories you have remembered about legends, there are occasions when a legend should stand on the output, and be plotted on its own. That's right, the legend should not be in a small area at the bottom of the plot, the legend should be the main focus. The legend should be the actual plot. Making the legend the actual plot is beneficial in situations where the legend is large or where the legend is being repeated on every plot. There is no need to have a legend on a plot that is taking up too much space and therefore squashing the plot and making the plot difficult to see. Also there is no need for the same legend to be at the bottom of every plot, which typically happens when you use a BY variable. Instead of having the same legend on the bottom of each plot another alternative is to simply add a plot of the legend alone on to the last page if you're outputting using rtf, or as a reference in a separate output if you're outputting image files.

Figure 1 below is an example of how the output looks when the legend is made the main focus. Plotting the legend alone can be achieved using GTL, and the plot statements: `SERIES`, `VECTOR` and `SCATTER`.

Figure 1: Legend



WHY PLOT A LEGEND ALONE AGAIN

Figures 2, 3 and 4 show the laboratory results by relative day for Cohorts 1, 2 and 3 respectively. You can see that the legend area is a reasonable size in Figures 2, 3 and 4 and that the legend area takes up less than 20% of the total plot area.

Figure 5 is the combination of the data in Figures 2, 3 and 4, and exemplifies the motivation for plotting a legend alone. You will notice that the legend is using over 50% of the total plot area, and that the plot of the laboratory results are displayed in a smaller area compared to Figures 2, 3 and 4. The default maximum area a legend can occupy is 20% and therefore the legend is only being displayed because the MAXLEGENDAREA option in the ODS GRAPHICS specified that a larger legend could be displayed. If the MAXLEGENDAREA did not specify to plot a larger legend, then the following note would have been in the LOG.

NOTE: Some graph legends have been dropped due to size constraints. Try adjusting the MAXLEGENDAREA=, WIDTH= and HEIGHT= options in the ODS GRAPHICS statement.

Given the fact that the default maximum legend is 20%, you can argue that if the legend area is more than 20% of the total plot area, then consideration should be made to plot the initial figure without the legend such as Figure 6, and then plot the legend on its own such as in Figure 1. The other argument is that if the legend takes up more than 20% of the data, then you should reduce the groups in a plot, so that the legend takes up less than 20% of the plot. Similar to Figures 2, 3 and 4.

Figure 2: Lab results of Cohort 1

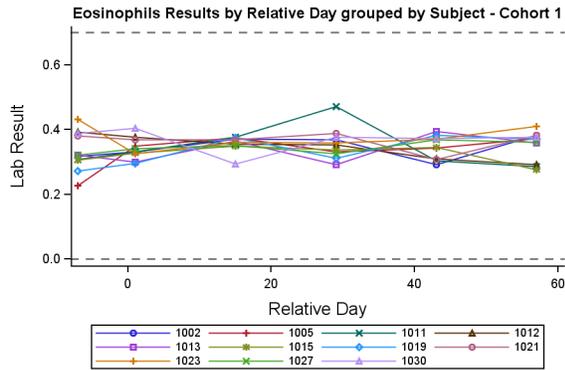


Figure 3: Lab results of Cohort 2

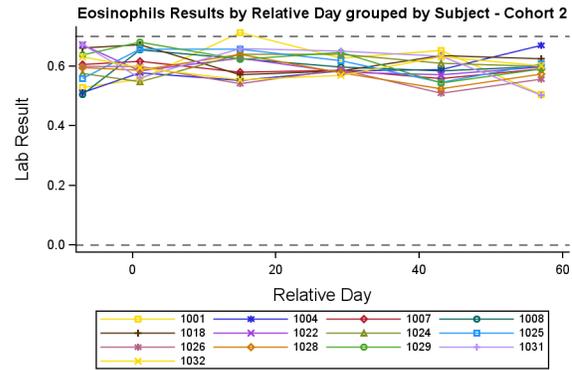


Figure 4: Lab results of Cohort 3

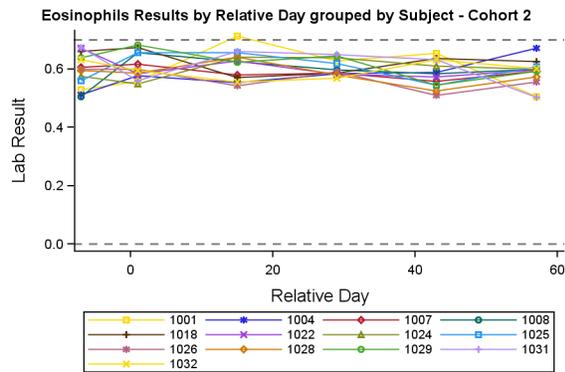


Figure 5: Lab results of all Cohorts

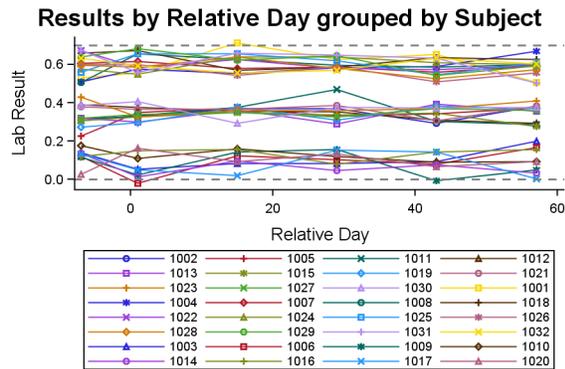
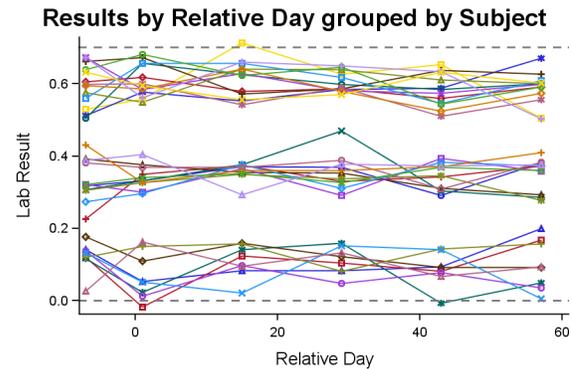


Figure 6: Lab results of all Cohorts and no legend



It is apparent that the absence of the legend in Figure 6 makes the line plot clearer than in Figure 5.

THE DATA AND CODE

Table 1 below shows the subset of the data used to produce the plot of the Legend. For your information there were 11 subjects in Cohort 1, 13 subjects in Cohort 2 and 8 subjects in Cohort 3. It is useful to be aware of these numbers so that the columns prefixed with the letter “y” can be understood better.

The data is sorted by the columns COHORT and SUBJID and there were 32 subjects in total. The column INDEX_FINAL represents this sorted order, and hence has values 1 to 32. The columns Y1, Y10, and Y12 are the mean, minimum and maximum of the INDEX_FINAL values in Cohort 1, and as mentioned before there are 11 subjects in Cohort 1. Therefore the minimum value is 1, the maximum value is 11, and the mean is 6. Similarly the columns Y2, Y20 and Y22 are the mean, minimum and maximum values in Cohort 2, and the same logic applies in

Cohort 3. The columns prefixed with “x” are values on the values that will be plotted on the x-axis which range from 1 to 10. There are 2 rows per subject so that the lines identifying the subjects can range from 6 to 10 as in the XAX column.

Table 1: Subset of the data

Subject Identifier	cohort	index_final	xax	x	x1	x2	x3	y1	y10	y12	y2	y20	y22	y3
1002	Cohort 1	1	6	1	3.5	4	5.5	6	1	11
1002	Cohort 1	1	10	1	3.5	4	5.5	6	1	11
1005	Cohort 1	2	6	1	3.5	4	5.5	6	1	11
1005	Cohort 1	2	10	1	3.5	4	5.5	6	1	11
1011	Cohort 1	3	6	1	3.5	4	5.5	6	1	11
1011	Cohort 1	3	10	1	3.5	4	5.5	6	1	11
1012	Cohort 1	4	6	1	3.5	4	5.5	6	1	11
1012	Cohort 1	4	10	1	3.5	4	5.5	6	1	11
1013	Cohort 1	5	6	1	3.5	4	5.5	6	1	11
1013	Cohort 1	5	10	1	3.5	4	5.5	6	1	11
1015	Cohort 1	6	6	1	3.5	4	5.5	6	1	11
1015	Cohort 1	6	10	1	3.5	4	5.5	6	1	11
1019	Cohort 1	7	6	1	3.5	4	5.5	6	1	11
1019	Cohort 1	7	10	1	3.5	4	5.5	6	1	11
1021	Cohort 1	8	6	1	3.5	4	5.5	6	1	11
1021	Cohort 1	8	10	1	3.5	4	5.5	6	1	11
1023	Cohort 1	9	6	1	3.5	4	5.5	6	1	11
1023	Cohort 1	9	10	1	3.5	4	5.5	6	1	11
1027	Cohort 1	10	6	1	3.5	4	5.5	6	1	11
1027	Cohort 1	10	10	1	3.5	4	5.5	6	1	11
1030	Cohort 1	11	6	1	3.5	4	5.5	6	1	11
1030	Cohort 1	11	10	1	3.5	4	5.5	6	1	11
1001	Cohort 2	12	6	1	3.5	4	5.5	.	.	.	18	12	24	.
1001	Cohort 2	12	10	1	3.5	4	5.5	.	.	.	18	12	24	.

LAYOUT OVERLAY

The LAYOUT OVERLAY statement below was used to set up the plot space. As you can see on Figure 7, the Y2 axis was used to display the subject numbers, and there were 32 subjects, hence the TICKVALUESEQUENCE options within the Y2AXISOPTS statement.

THE MAIN PLOT (SERIES PLOT)

```
layout overlay/xaxisopts=(display = NONE) yaxisopts = (display = NONE)
      y2axisopts = ( display = STANDARD reverse = true label = "Subject"
      linearopts= ( tickvaluesequence=(start=1 end=32 increment=1)));
```

```
/* Main plot */
seriesplot x=xax y= index_final / yaxis = y2 display = all group=subjid
index = index_final lineattrs = (pattern =1 thickness = 1) markerattrs =
(size = 1pt);
```

The SERIES PLOT statement was used to draw the legend, i.e. the line colors and symbols that identifies the subjects across all plots. It is vital that the column in the option **INDEX =** is consistent in all datasets so that the correct subject can be identified by the legend plot. So for example, in these datasets subject 1002, should always have the value 1 as its index, and similarly subject 1020 should always have the value 32 as its index. Pattern 1 has been chosen in the LINEATTRS option so that all of the lines are solid. Again the pattern used to identify the patients, should match the pattern in the other graphs.

COHORT LABEL INFORMATION (VECTOR PLOT AND SCATTERPLOT)

The VECTOR combined with the SCATTERPLOT statement adds the finishing touches to the legend and nicely details on the plot the subjects that are assigned to each cohort. The VECTOR statement draws the vertical and horizontal lines on the plot. For example the first VECTOR statement has XORIGIN and YORIGIN equal to the columns X1 and Y1 respectively. These columns contain the values 3.5 and 6. The statement then has X and Y values equal to 4 and 6 respectively. Therefore the first statement draws a line from the coordinates (3.5, 6) to (4, 6).

The SCATTERPLOT statement plots the label "Cohort 1" at the coordinates (1, 6). As mentioned before the value 6 is the mean value of Cohort 1 and that is why it was calculated.

```

/* Cohort 1 */
vectorplot xorigin=x1 yorigin=y1 x=x2 y=y1 / yaxis = y2 ARROWHEADS=false;
vectorplot xorigin=x2 yorigin=y1 x=x2 y=y10 / yaxis = y2 ARROWHEADS=false;
vectorplot xorigin=x2 yorigin=y10 x=x3 y=y10 / yaxis = y2 ARROWHEADS=false;
vectorplot xorigin=x2 yorigin=y1 x=x2 y=y12 / yaxis = y2 ARROWHEADS=false;
vectorplot xorigin=x2 yorigin=y12 x=x3 y=y12 / yaxis = y2 ARROWHEADS=false;

scatterplot x = x y = y1 / markercharacter = c1 yaxis = y2;

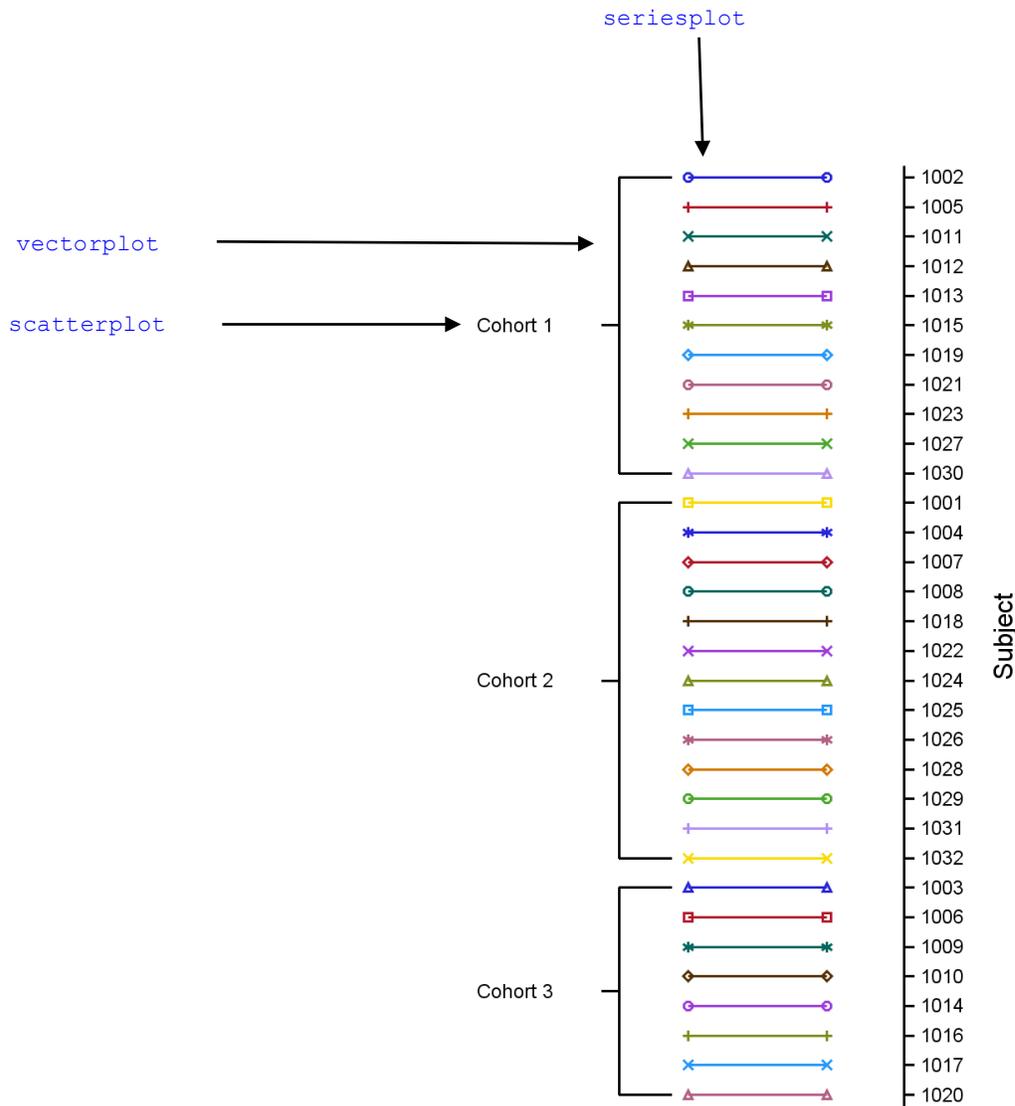
/* Cohort 2 */
vectorplot xorigin=x1 yorigin=y2 x=x2 y=y2 / yaxis = y2 ARROWHEADS=false;
vectorplot xorigin=x2 yorigin=y2 x=x2 y=y20 / yaxis = y2 ARROWHEADS=false;
vectorplot xorigin=x2 yorigin=y20 x=x3 y=y20 / yaxis = y2 ARROWHEADS=false;
vectorplot xorigin=x2 yorigin=y2 x=x2 y=y22 / yaxis = y2 ARROWHEADS=false;
vectorplot xorigin=x2 yorigin=y22 x=x3 y=y22 / yaxis = y2 ARROWHEADS=false;

scatterplot x = x y = y2 / markercharacter = c2 yaxis = y2;

/* Cohort 3 */

```

Figure 7: Legend plot with descriptive annotation



CONCLUSION

Plotting the legend alone is useful when the legend is either taking up too much space on the plot or when the same legend is plotted on every output, which essentially takes up valuable space of the actual figure you want to analyze. Adding the VECTORPLOT statement makes the legend appear more glamorous, although if you wanted to you could plot a basic legend alone by only using the SERIESPLOT statement.

For an even simpler legend, which is potentially quicker to produce, please see Sanjay Matange's article on legends that is mentioned in the references section below.

On occasions it may be better to avoid using a legend altogether and use direct labeling instead. An advantage of using direct labeling is that it minimizes the eye movement that occurs when trying to decode the colors on the plot with the legend.

REFERENCES

Kuhfeld, Warren. February 2010. *Statistical Graphics in SAS: An Introduction to the Graph Template Language and the Statistical Graphics Procedures*. Cary, North Carolina. SAS Institute.

Matange, Sanjay. *Make a Good Graph*. Available at <http://support.sas.com/resources/papers/proceedings13/361-2013.pdf>

Graphically Speaking: Just a legend, please. (2014, January). Available at <http://blogs.sas.com/content/graphicallyspeaking/2012/04/20/just-a-legend-please/>

ACKNOWLEDGMENTS

I would like to thank Sanjay Matange of SAS Institute for reviewing this paper and offering valuable suggestions. Additionally, I would like to thank Sharon Carroll for the support she gave me whilst preparing the paper.

RECOMMENDED READING

- Matange, Sanjay. November 2013. *Getting Started with the Graph Template Language in SAS: Examples, Tips, and Techniques for Creating Custom Graphs*. Cary, North Carolina. SAS Institute.

CONTACT INFORMATION

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

Name: Kriss Harris
Enterprise: SAS Specialists Ltd.
E-mail: italjet125@yahoo.com
Web: <http://www.krissharris.co.uk>
Twitter: <https://twitter.com/krissharris>

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.

APPENDIX

```
/* Creating the Format for Cohort */  
  
proc format;  
value cohort 1 = "Cohort 1"  
            2 = "Cohort 2"  
            3 = "Cohort 3";  
run;  
  
/* Creating the Cohort and Subject alignment */  
data leg_prepare;  
call streaminit(4001);  
do subjid = 1001 to 1032;  
    cohort = ceil(RAND('UNIFORM') * 3) ;  
    subjidc = strip(put(subjid, best.));  
    format cohort cohort.;  
    output;  
end;  
run;  
  
proc sort data = leg_prepare;  
by cohort subjid;  
run;  
  
/* Creating the random data for the line plots */  
  
data random_lab_data;  
set leg_prepare;  
call streaminit(400);  
if cohort = 1 then do lbdy = -7, 1 to 57 by 14;  
    lbtest = "Eosinophils";  
    mean = 0.35;  
    sd = 0.05;  
    lbstresn = rand("normal", mean, sd);  
    lbstnrlo = 0;  
    lbstnrhi = 0.7;
```

```
output;
end;

if cohort = 2 then do lbdy = -7, 1 to 57 by 14;
    lbtest = "Eosinophils";
    mean = 0.6;
    sd = 0.05;
    lbstresn = rand("normal", mean, sd);
    lbstnrlo = 0;
    lbstnrhi = 0.7;
output;
end;

if cohort = 3 then do lbdy = -7, 1 to 57 by 14;
    lbtest = "Eosinophils";
    mean = 0.1;
    sd = 0.05;
    lbstresn = rand("normal", mean, sd);
    lbstnrlo = 0;
    lbstnrhi = 0.7;
output;
end;

label subjid = "Subject Identifier";
label lbtest = "Lab Test or Examination Name";
label lbstresn = "Numeric Result/Finding in Standard Units";
label lbstnrlo = "Reference Range Lower Limit-Std Unit";
label lbstnrhi = "Reference Range Upper Limit-Std Unit";
label lbdy = "Study Day of Specimen Collection";

drop mean sd;
run;

/** Back to creating the legend */

proc sql;
create table distinct_subject_cohort as
select distinct subjid, cohort, subjidc
from random_lab_data
order by cohort, subjid;
quit;

data distinct_subject_cohort_index;
set distinct_subject_cohort;
index_final = _n_;
run;

proc sql;
create table format_dataset as
select distinct index_final as start, subjidc as label, "subjid" as fmtname
from distinct_subject_cohort_index;
quit;

proc format cntlin = format_dataset;
run;

proc means data = distinct_subject_cohort_index;
class cohort;
var index_final;
output out = summary min = min mean = mean max = max;
run;

proc sql;
create table leg prepare merge as
select a.*, b.min, b.mean, b.max
from distinct_subject_cohort_index as a left join summary as b
on a.cohort = b.cohort
order by index_final;
quit;
```

```

data legend_pre;
set Leg_prepare_merge;
x = 1;
x1 = 3.5;
x2 = 4;
x3 = 5.5;

/* For Vector plots - denoting cohort group*/

if cohort = 1 then do;
    y1 = mean;
    y10 = min;
    y12 = max;
end;

else if cohort = 2 then do;
    y2 = mean;
    y20 = min;
    y22 = max;
end;

else if cohort = 3 then do;
    y3 = mean;
    y30 = min;
    y32 = max;
end;

/* For patient legend */
do i = 6, 10;
    xax = i;
output;
end;
drop i;

run;

proc sort data = legend_pre;
by cohort index_final;
run;

data legend;
set legend_pre;
by cohort index_final;
if cohort = 1 then do;
    if first.cohort then c1 = "Cohort 1";
end;
if cohort = 2 then do;
    if first.cohort then c2 = "Cohort 2";
end;
if cohort = 3 then do;
    if first.cohort then c3 = "Cohort 3";
end;
drop subjidc min mean max;
run;

proc sort data = legend;
by index_final;
run;

/* Legend Plot */

proc template;
    define style styles.KJHPlot;
        parent = Styles.listing;
        style GraphFonts from GraphFonts /
            'GraphValueFont' = ("<sans-serif>, <MTsans-serif>",6pt)
            'GraphLabelFont'=( "<sans-serif>, <MTsans-serif>",8pt)
            'GraphDataFont'=( "<sans-serif>, <MTsans-serif>",6pt);
        CLASS graphWalls / FRAMEBORDER=off;
    enddefine;

```

```

    end;
run;

ods graphics on / reset = all border = off imagefmt = png width = 2.5in height = 4in imagename
= "grlegend";

goptions reset = all;

ods listing style = styles.KJHPlot image_dpi = 300 gpath = "C:\Kriss\PharmaSUG 2014";

proc template;
  define statgraph order;
    begingraph;

    layout overlay / xaxisopts =(display = NONE) yaxisopts = (display = NONE)
                    y2axisopts = ( display = STANDARD reverse = true label = "Subject"
                    linearopts= ( tickvaluesequence=(start=1 end=32 increment=1)));

    /* Main plot */
    seriesplot x=xax y= index_final / yaxis = y2 display = all group=subjid
    index = index_final lineattrs = (pattern =1 thickness = 1) markerattrs = (size = 1pt);

    /* Cohort 1 */
    vectorplot xorigin=x1 yorigin=y1 x=x2 y=y1 / yaxis = y2 ARROWHEADS=false;
    vectorplot xorigin=x2 yorigin=y1 x=x2 y=y10 / yaxis = y2 ARROWHEADS=false;
    vectorplot xorigin=x2 yorigin=y10 x=x3 y=y10 / yaxis = y2 ARROWHEADS=false;
    vectorplot xorigin=x2 yorigin=y1 x=x2 y=y12 / yaxis = y2 ARROWHEADS=false;
    vectorplot xorigin=x2 yorigin=y12 x=x3 y=y12 / yaxis = y2 ARROWHEADS=false;
    scatterplot x = x y = y1 / markercharacter = c1 yaxis = y2;

    /* Cohort 2 */
    vectorplot xorigin=x1 yorigin=y2 x=x2 y=y2 / yaxis = y2 ARROWHEADS=false;
    vectorplot xorigin=x2 yorigin=y2 x=x2 y=y20 / yaxis = y2 ARROWHEADS=false;
    vectorplot xorigin=x2 yorigin=y20 x=x3 y=y20 / yaxis = y2 ARROWHEADS=false;
    vectorplot xorigin=x2 yorigin=y2 x=x2 y=y22 / yaxis = y2 ARROWHEADS=false;
    vectorplot xorigin=x2 yorigin=y22 x=x3 y=y22 / yaxis = y2 ARROWHEADS=false;

    scatterplot x = x y = y2 / markercharacter = c2 yaxis = y2;

    /* Cohort 3 */
    vectorplot xorigin=x1 yorigin=y3 x=x2 y=y3 / yaxis = y2 ARROWHEADS=false;
    vectorplot xorigin=x2 yorigin=y3 x=x2 y=y30 / yaxis = y2 ARROWHEADS=false;
    vectorplot xorigin=x2 yorigin=y30 x=x3 y=y30 / yaxis = y2 ARROWHEADS=false;
    vectorplot xorigin=x2 yorigin=y3 x=x2 y=y32 / yaxis = y2 ARROWHEADS=false;
    vectorplot xorigin=x2 yorigin=y32 x=x3 y=y32 / yaxis = y2 ARROWHEADS=false;

    scatterplot x = x y = y3 / markercharacter = c3 yaxis = y2;

    endlayout;

  endgraph;
end;
run;

proc sgrender data= legend template=order;
format index_final subjid.;
run;

ods listing close;

/* Line Plots */

proc template;
  define style styles.KJHPlot;
    parent = Styles.listing;
    style GraphFonts from GraphFonts /
      'GraphValueFont' = (", <MTsans-serif>", 6pt)
      'GraphLabelFont' = (", <MTsans-serif>", 8pt)
      'GraphDataFont' = (", <MTsans-serif>", 6pt)
  end;
endrun;

```

```

        'GraphFootnoteFont' = ("<sans-serif>, <MTsans-serif>",6pt,italic)
        'GraphTitleFont' = ("<sans-serif>, <MTsans-serif>",7pt,bold);

CLASS graphWalls / FRAMEBORDER=off;

    end;
run;

ods graphics on / reset = all border = off imagefmt = png width = 3in height = 2in imagename =
"grlabplot";

goptions reset = all;

ods listing style = styles.KJHPlot image_dpi = 300 gpath = "C:\Kriss\PharmaSUG 2014";

proc template;
define statgraph labplot;
dynamic _BYVAL_ _BYVAL2_;
begingraph;
    entrytitle _BYVAL_ " Results by Relative Day grouped by Subject - " _BYVAL2_;

    layout overlay / xaxisopts = (label = "Relative Day")
        yaxisopts = (label = "Lab Result");
    seriesplot x=lbdy y=lbstresn / display = all name="serieslabplot" group=subjid
index = index_final lineattrs = (pattern =1) markerattrs = (size = 4);

    discretelegend "serieslabplot";
    referenceline y = lbstnrlo / lineattrs = (pattern = 2);
    referenceline y = lbstnrhi / lineattrs = (pattern = 2);

    endlayout;

endgraph;
end;
run;

proc sgrender data= random_lab_data template=labplot;
by lbtest cohort;
format index_final subjid.;
run;

ods listing close;

/* All cohort plot with Legend */

ods graphics on / reset = all border = off MAXLEGENDAREA= 60 imagefmt = png width = 3in height
= 2in imagename = "grlabplot_wl";

goptions reset = all;

ods listing style = styles.KJHPlot image_dpi = 300 gpath = "C:\Kriss\PharmaSUG 2014";

proc template;
define statgraph labplot;
dynamic _BYVAL_ _BYVAL2_;
begingraph;
    entrytitle _BYVAL_ " Results by Relative Day grouped by Subject" _BYVAL2_;

    layout overlay / xaxisopts = (label = "Relative Day")
        yaxisopts = (label = "Lab Result");
    seriesplot x=lbdy y=lbstresn / display = all name="serieslabplot" group=subjid
index = index_final lineattrs = (pattern =1) markerattrs = (size = 4);

    discretelegend "serieslabplot";
    referenceline y = lbstnrlo / lineattrs = (pattern = 2);
    referenceline y = lbstnrhi / lineattrs = (pattern = 2);

    endlayout;

endgraph;

```

```
end;
run;

proc sgrender data= random_lab_data template=labplot;
format index_final subjid.;
run;

ods listing close;

/* All cohort plot without Legend */

ods graphics on / reset = all border = off MAXLEGENDAREA= 60 imagefmt = png width = 3in height
= 2in imagename = "grlabplot_n1";

goptions reset = all;

ods listing style = styles.KJHPlot image_dpi = 300 gpath = "C:\Kriss\PharmaSUG 2014";

proc template;
define statgraph labplot;
dynamic _BYVAL_ _BYVAL2_;
begingraph;
entrytitle _BYVAL_ " Results by Relative Day grouped by Subject" _BYVAL2_;

layout overlay / xaxisopts = (label = "Relative Day")
yaxisopts = (label = "Lab Result");

seriesplot x=lbdy y=lbstresn / display = all group=subjid index = index_final
lineattrs = (pattern =1) markerattrs = (size = 4);

referenceline y = lbstnrlo / lineattrs = (pattern = 2);
referenceline y = lbstnrhi / lineattrs = (pattern = 2);

endlayout;

endgraph;
end;
run;

proc sgrender data= random_lab_data template=labplot;
format index_final subjid.;
run;

ods listing close;
```