Paper 1601-2015

Nesting Multiple Box Plots and BLOCKPLOTS using GTL and Lattice Overlay

SAS®

Greg Stanek MS

ABSTRACT

There are times when the objective is to provide a summary table and graph for several quality improvement measures on a single page to allow leadership to monitor the performance of measures over time. The challenges were to decide which SAS procedure(s) to use and how to integrate multiple SAS procedures to generate a set of plots and summary tables within one page and to determine whether to use box plots or series plots of means or medians. The use of SGPLOT and SGPANEL Procedures and Graph Template Language (GTL) were considered.

While SGPLOT and the SGPANEL procedures provide easier alternatives and better graphics with ODS than GPLOT and GREPLAY, at times given the nature of the request there is a need to utilize the power of Graph Template Language (GTL) and the SGRENDER Procedure, which is the focus. We develop a Macro called %BXPLOT2, where our initial effort, focuses on the use of the BOXPLOTPARM statement to display a series of box plots and the BLOCKPLOT statement for a summary table and then used the LAYOUT OVERLAY statement to combine the box plots and summary tables on one page. The secondary initiative is to illustrate the flexibility of the code to allow other plots, such as the SERIESPLOT and BANDPLOT respectfully, which allows the user to view different types of plots on the page.

The results display a summary table (BLOCKPLOT) above each box plot series for each measure on a single page. Within each plot of the series there is an option to overlay benchmark value(s) and a series line connecting the median values of each box plot. The BLOCKPLOT contains descriptive statistics per time period illustrated in the associated plot statement(s).

The discussion points will focus on techniques for nesting the lattice overlay with box plots and BLOCKPLOTS in GTL and some reasons for choosing box plots versus series plots of medians or means.

INTRODUCTION

SAS has provided some nice graphical options to summarize and illustrate the data from the SGPLOT Procedures to the GTL. Each of the procedures was explored to address our particular need and it led us to GTL given its flexibility.

GTL has proven to be flexible and after some investment of time it opens alternate avenues to generate multiple graphs on a single page through the OVERLAY statements. One of the many features of GTL is the flexibility it has to customize reports with the LAYOUT and PLOT statements with the variety of associated OPTIONS. One of the interesting features is the ability to generate multiple plots within a layout that is nested within a layout with sets of the nested plots on a page.

While there are many ways to provide summarized results on a single page, we focus on nesting plots within a series of LAYOUT OVERLAY statements in GTL and highlight some of the options used for the final product. We specifically focus on using the LAYOUT statements (LAYOUT OVERLAY and LAYOUT LATTICE) and the PLOT statements (BLOCKPLOT, BOXPLOTPARM, SERIESPLOT, BANDPLOT) respectfully. The BLOCKPLOT statement allows us to provide summary level statistics associated with other PLOT statements in GTL. For our initial effort

we use the BLOCKPLOT with BOXPLOTPARM for box plots. We then follow up to use the BLOCKPLOT with the SERIESPLOT and BANDPLOT statements. The intent of the paper is to provide an approach to generate plots of different measures on a single page where there is a need to have greater flexibility than some of the other graphic options available.

We embedded the GTL code and SGRENDER Procedure within %BXPLOT2. The macro %BXPLOT2 utilizes PROC SQL to generate the macro variables for the y-axis options for the plots. The orientation of the report is portrait, but the report can be generated in landscape as well by using the options orientation=landscape and modifying the ht= and wdt= options within %BXPLOT2. Please refer to the appendix section to see the full code for %BXPLOT2. The following are the input parameters for %BXPLOT2.

```
% BXPLOT2(
INDAT=Input Dataset,
Y1=Y-Axis Measure variable for top right quadrant,
Y2= Y-Axis Measure variable for top left quadrant,
Y3= Y-Axis Measure variable for bottom right quadrant,
Y4= Y-Axis Measure variable for bottom left quadrant,
PLOT_TYPE1=Type of Chart for Y1 (SERIES, BOXPLOT, etc.),
PLOT_TYPE2= Type of Chart for Y2 (SERIES, BOXPLOT, etc.),
PLOT_TYPE3= Type of Chart for Y3 (SERIES, BOXPLOT, etc.),
PLOT_TYPE4= Type of Chart for Y4 (SERIES, BOXPLOT, etc.),
x=X-Axis variable,
YR1=Y-Axis Reference Line 1,
YR2= Y-Axis Reference Line 2,
HT=Adjust the height of the report. Default 1000px,
WDT=Adjust the width of the report. Default 900px,
SZ=Adjusts the size of the font default 5,
RSTRCT=Restriction criteria for data= in SGRENDER,
ENTTL=Title Name for Report for Entry Title in GTL,
ENTFT=Footer Name for Report for Entry Footer in GTL,
DSPL=Display options for the BOXPLOTPARM Statement (Connect, Median, Means, CAPS, Fill)
LBL1= Label for Measure variable for top right quadrant,
LBL2= Label for  Measure variable for top left quadrant,
LBL3= Label for  Measure variable for bottom right quadrant,
LBL4= Label for  Measure variable for bottom left quadrant
NO_GRPS=Number of BINS for the Y-Axis);
```

OVERVIEW:

GRAPH TEMPLATE LANGUAGE (GTL):

The SAS 9.3 GTL reference guide provides a nice overview of GTL and we defer the reader to the SAS documentation website to review. http://support.sas.com/documentation/

BLOCKPLOT STATEMENT:

The BLOCKPLOT statement is a plot statement within GTL which allows us to provide a summary table with other graphical plot statements within GTL. The BLOCKPLOT statement also has a variety of options that allows us to shape the table format and with some time investment it produces a viable option to address reporting needs. For our efforts we focused on options to our specific need, but the example should help provide an avenue of exploration beyond our need[2,3,4].

For starters, the following code and example (Table 1) is an excerpt of the BLOCKPLOT statement with options we used within GTL. The CLASS=CLASS option creates a separate block plot for each unique categorical value[2,4]. In our case we had a categorical variable of summary statistics, i.e., the count (# of Cases), Median, 1st Quartile (25th percentile), and 3rd Quartile (75th percentile) and Mean, but we can include other statistics such as confidence limits, min and max values. The INCLUDEMISSINGCLASS = (TRUE/FALSE) provides the option to either include or exclude missing values of the class variable. The REPEATEDVALUES=(TRUE/FALSE) provides an option to either combine or segment identical block values along the x-axis[3,4]. Another option available is VALUEFITPOLICY= TRUNCATE or SHRINK, but it did not uniformly shrink the values in each of the blocks for our examples, so we decided to use The LABELATTRS= and VALUEATTRS= options to adjust the size of the text uniformly. Finally the VALUEHALIGN=LEFT|CENTER|RIGHT|START and the OUTLINEATTERS= (color=) options were used to adjust the color of the outline of the tables and alignment of the values in the cells[3].

```
%MACRO BLOCKPLOT(INDAT=,Y1=,X=,HT=100PX,WDT=9000PX,SZ=5,RSTRCT=,ENTFT=);
   PROC TEMPLATE;
      DEFINE STATGRAPH BLOCKPLOT;
         BEGINGRAPH;
               LAYOUT OVERLAY;
                  LAYOUT LATTICE/COLUMNS=1;  COLUMNAXES; COLUMNAXIS/GRIDDISPLAY=ON  LABEL=' '; ENDCOLUMNAXES;
                     BLOCKPLOT X=BL_&X. BLOCK=BL_&Y1. /CLASS=BL_STATS INCLUDEMISSINGCLASS=FALSE REPEATEDVALUES=TRUE
                     DISPLAY=(VALUES LABEL OUTLINE ) OUTLINEATTRS=(COLOR=LIGHTGRAY) VALUEHALIGN=CENTER
                     LABELATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+1)PT) VALUEATTRS=GRAPHDATATEXT(SIZE=&SZ.PT);
                  ENDLAYOUT;
               ENDLAYOUT;
            ENTRYFOOTNOTE HALIGN=LEFT "&ENTFT." /TEXTATTRS=(SIZE=%EVAL(&SZ.+1)PT STYLE=ITALIC);
         ENDGRAPH;
   END;
   RUN;
   ODS GRAPHICS ON / BORDER=OFF NOBORDER HEIGHT=&HT WIDTH=&WDT;
   PROC SGRENDER DATA=&INDAT. &RSTRCT. TEMPLATE=BLOCKPLOT; RUN;
%MEND BLOCKPLOT;

%BLOCKPLOT(INDAT=TEMP_ALL,Y1=X1,X=MTH,HT=1IN,WDT=4IN,SZ=8,RSTRCT=(WHERE=(UPCASE(PROCESS)="PROCESS_1")),ENTFT=TABLE 1: EXAMPLE
OF BLOCKPLOT STATEMENT);
```

1: # Cases	259	256	259	250	256	259	256	251	256	254	253	252
2: 25th Pctl	10	10	9	11	11	11	12	11	10	10	10	11
3: Median	18	20	19	19	20	19	21	20	18	20	19	19
4: 75th Pctl	37	39	36	38	39	37	43	34	39	39	40	34
5: Mean	31	34	36	33	33	30	35	31	30	31	35	32

Table 1: Example of BLOCKPLOT Statement

BOXPLOTPARM STATEMENT:

The BOXPLOTPARM statement is another plot statement within GTL that provides box plots using summary descriptive statistics, where the BOXPLOT statement is used for the raw data. The descriptive summary statistics fed into BOXPLOTPARM can be generated from PROC SHEWHART's BOXPLOT Statement and OUTBOX= option and/or one of the descriptive statistic summary procedures, i.e., PROC MEANS or PROC SUMMARY with some additional data manipulations to capture additional summary statistics if PROC SHEWHART is not an option for box plots.

The flexibility of the options in the BOXPLOTPARM statements allows us to customize the output easily. The following piece of code contains some of the options we utilized for our box plots. The DISPLAY=STANDARD|ALL|(display options) allows us to use the prepackaged STANDARD|ALL option or to customize to with the display options available. For our efforts we utilized the display options, where the CONNECT=MEAN|MEDIAN|Q1|Q3|MIN|MAX option provides the option of connecting one of the listed summary statistics[1,2,3]. This option is particularly useful when plotting a measure over time (see Table 2).

```
%MACRO BOXPLOT(INDAT=,Y1=,X=,YR1=,YR2=,HT=,WDT=,SZ=,MTH=,RSTRCT=,DSPL=,LBL1=,ENTFT=,NO_GRPS=8);

   %GLOBAL MIN_Y1 MAX_Y1 INCRMT_Y1 ;


   PROC SQL NOPRINT;
      SELECT
         MIN(0,MIN(CASE WHEN BX_STATS='MIN' THEN BX_&Y1. END)) AS MIN_Y1
         ,MAX(CASE WHEN BX_STATS='MAX' THEN
             (CASE   WHEN 0<=  (BX_&Y1.)< 0          THEN .
                WHEN 0<= (BX_&Y1.)<=1                THEN ROUND((BX&Y1.+.1),.1)
                WHEN 1<  (BX_&Y1.)<100               THEN ROUND((BX_&Y1.+20),10)
                WHEN 100<=(BX_&Y1.)<500              THEN ROUND((BX_&Y1.+50),100)
                WHEN 500<=(BX_&Y1..)                 THEN ROUND((BX_&Y1.+500),1000)
                ELSE 0 END)
         END) AS MAX_Y1
         ,CEIL(CALCULATED MAX_Y1/&NO_GRPS.) AS INCRMT_Y1
            INTO :MIN_Y1,  :MAX_Y1,   :INCRMT_Y1
      FROM &INDAT. &RSTRCT.;QUIT;

   %LET OFFSETS = OFFSETMIN=0.1 OFFSETMAX=0.05;

   PROC TEMPLATE;
      DEFINE STATGRAPH BOXPLOT;
         BEGINGRAPH;
            LAYOUT OVERLAY/XAXISOPTS=(&OFFSETS LABEL=' ' TICKVALUEATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+2)PT))
               YAXISOPTS=( &OFFSETS GRIDDISPLAY=ON   GRIDATTRS=(COLOR=LIGHTGRAY) LABEL="&LBL1."
               LABELATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+3)PT) TICKVALUEATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+2)PT)
               LINEAROPTS=(TICKVALUESEQUENCE=(START=&MIN_Y1. END=&MAX_Y1. INCREMENT=&INCRMT_Y1.) TICKVALUEPRIORITY=TRUE));
                  BOXPLOTPARM Y=BX_&Y1. X=BX_&X. STAT=BX_STATS /CONNECT=MEDIAN DISPLAY=&DSPL.  DATALABEL=DATALABEL ;
                  REFERENCELINE Y=&YR1./LINEATTRS=(PATTERN=DASH COLOR=BLUE);
                  REFERENCELINE Y=&YR2./LINEATTRS=(PATTERN=SOLID );
            ENDLAYOUT;
               ENTRYFOOTNOTE HALIGN=LEFT "&ENTFT." /TEXTATTRS=(SIZE=%EVAL(&SZ.+1)PT STYLE=ITALIC);
         ENDGRAPH;
   END;RUN;

   ODS GRAPHICS ON / BORDER=OFF NOBORDER HEIGHT=&HT. WIDTH=&WDT.;
   PROC SGRENDER DATA=&INDAT. &RSTRCT. TEMPLATE=BOXPLOT;RUN;
%MEND BOXPLOT;
```

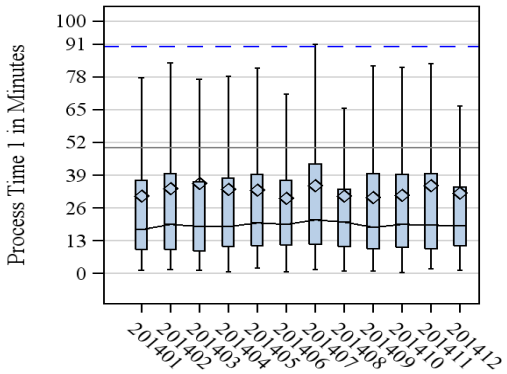


Table 2: Example of BOXPLOTPARM Statement

%*BOXPLOT* (INDAT=TEMP_ALL, Y1=X1, X=MTH, YR1=90, YR2=50, HT=3IN, WDT=3IN,SZ=8, RSTRCT=(WHERE=(UPCASE(PROCESS)="PROCESS_1")), DSPL=(MEDIAN MEAN CAPS FILL CONNECT), LBL1=PROCESS TIME 1 IN MINUTES, ENTFT=TABLE 2: EXAMPLE OF BOXPLOTPARM STATEMENT);

SERIESPLOT STATEMENT:

The SERIESPLOT statement utilizes line segments to connect data points and requires X= and Y=. We did not use any of the additional options for the SERIESPLOT and defer the reader to the SAS documentation (Code for Table3 is in Appendix).

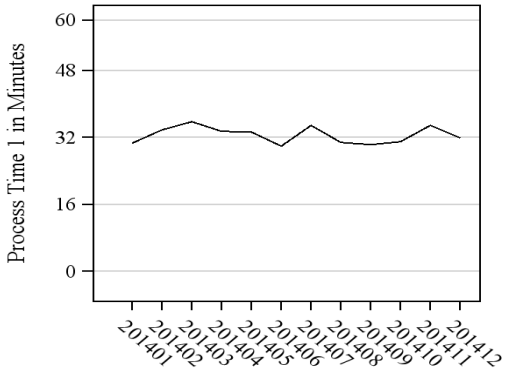


Table 3: Example of SERIESPLOT Statement

BANDPLOT STATEMENT:

The BANDPLOT statement plots the area within an upper and lower limit and is generally used for confidence limits, control limits, etc. either across the x-axis or y-axis. The BANDPLOT statement requires BANDPLOT X= or BANDPLOT Y= with LIMITLOWER= and LIMITUPPER=. In our efforts we used BANDPLOT X=. We used the options TYPE=, DATATRANSPARENCY= and FILLATTRS=. The TYPE= allows us to choose between a SERIES (Smooth) band or a STEP(stepwise) plot. DATATRANSPARENCY= option allows us to adjust the transparency of the shaded area that ranges from 0=Opaque to 1=Total Transparency. The FILLATTRS= option allows us to adjust the attributes within the area (Code for Table 4 is in Appendix).

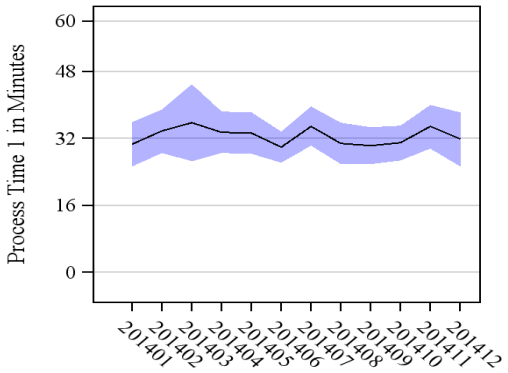


Table 4: Example of BANDPLOT Statement

LAYOUT OVERLAY STATEMENT:

The LAYOUT OVERLAY statement allows us to overlay plots on the same graph. The options of the LAYOUT OVERLAY statement also allows us to customize the x-axis and y-axis with the XASISOPTS and YAXISPOTS=options[1,2,3].

Continuing with our example from the BOXPLOTPARM section (see Table 2) we overlaid a couple of reference lines, include gridlines, and adjust the tick values, labels and scale of the y-axis on the box plots. The OFFSETMIN= and OFFSETMAX= options are on a [0,1] scale and allows us to adjust the plot area where the tick marks do not expand beyond the defined offset area[1,2,3,6].

The LABEL= option allows us to label the axis. The TICKVALUEATTRS= option allows us to adjust the attributes of the tick values such as the color size, etc. The GRIDDISPLAY= and GRIDATTRS= options allows us to display a Grid overlay and adjust the gridline color, size, etc. The LINEAROPTS=Option has several options that allows us to adjust the displayed axis. The TICKVALUESEQUENCE=(sequence options) provides us the flexibility of defining our start and end points for the tick values, where we can also adjust the increments for the gridlines and tick marks[2,3].

LAYOUT LATTICE STATEMENT:

The LAYOUT LATTICE statement provides the opportunity to plot charts and graphs in a $n\ row\ by\ m\ column$ plot area where n and m are not necessarily equal. The ROWS= and COLUMNS= options allow us to define the number of rows and columns needed for the charts and graphs. The ROWGUTTER= and COLUMNGUTTER= options adjusts the space between the plot regions where the COLLUMNWEIGHTS=() and ROWWEIGHTS=() options assign a weight to the column and rows for the plot area, where weights are between [0,1] [2,3,5,6].

The following code provides a view of how the LAYOUT, BLOCKPLOT and PLOT statements are combined for BOXPLOTPARM, SERIESPLOT, and BANDPLOT. In %BXPLOT2, the LAYOUT LATTICE divides the plot area into four equal quadrants:

```
%MACRO BXPLOT2(INDAT=,Y1=,Y2=,Y3=,Y4=,PLOT_TYPE1=,PLOT_TYPE2=,PLOT_TYPE3=,PLOT_TYPE4=, X=, YR1=' ', YR2=' ', HT=1000PX, WDT=900PX,
SZ=5, RSTRCT=, ENTTL=, ENTFT=, DSPL=,LBL1=,LBL2=,LBL3=,LBL4=, NO_GRPS=8);
   PROC TEMPLATE;
      DEFINE STATGRAPH BOXPLOT2;
         BEGINGRAPH;
            ENTRYTITLE "&ENTTL. ";
            LAYOUT LATTICE / ROWS=2 COLUMNS=2 ROWGUTTER=5 COLUMNGUTTER=10 COLUMNWEIGHTS=(.5 .5) ROWWEIGHTS=(.5 .5);
            ⋮
               %DO J=1 %TO 4;
                  %PLOTS;
               %END;
            ⋮
            ENDLAYOUT;
            ENTRYFOOTNOTE HALIGN=LEFT "&ENTFT." /TEXTATTRS=(SIZE=%EVAL(&SZ.+1)PT STYLE=ITALIC);
         ENDGRAPH;
   END;
   RUN;
%MEND BXPLOT2;
```

In %PLOTS, the LAYOUT OVERLAY (1) is specific for each plot within a quadrant and within each quadrant the LAYOUT LATTICE statement with ROWWEIGHTS=(.20 .80) (2) segments the quadrants into a 20/80 split, where the .20 is the area for the BLOCKPLOT statement and the .80 is for the BOXPLOTPARM, SERIESPLOT, and BANDPLOT Statements[2,3,4,5]. The last LAYOUT OVERLAY (3) is for the BOXPLOTPARM, SERIESPLOT and/or BANDPLOT Statement(s):

```
%MACRO PLOTS;
   LAYOUT OVERLAY;                                                                                          (1)
      LAYOUT LATTICE/COLUMNS=1 ROWWEIGHTS=(.20 .80) ROWGUTTER=2;                                            (2)
         COLUMNAXES; COLUMNAXIS/GRIDDISPLAY=ON LABEL=' '; ENDCOLUMNAXES;
            BLOCKPLOT X=BL_&X. BLOCK=BL_&&Y&J. /CLASS=BL_STATS
            INCLUDEMISSINGCLASS=FALSE REPEATEDVALUES=TRUE
            DISPLAY=(VALUES LABEL OUTLINE ) OUTLINEATTRS=(COLOR=LIGHTGRAY) VALUEHALIGN=CENTER
            LABELATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+1)PT) VALUEATTRS=GRAPHDATATEXT(SIZE=&SZ.PT);
            LAYOUT OVERLAY /XAXISOPTS=(&OFFSETS LABEL=' ' TICKVALUEATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+2)PT))
            ⋮                                                                                               (3)
            %IF %UPCASE(&&PLOT_TYPE&J.)=BXPLOT %THEN %DO;
               %BXPLOT
            %END;
            ⋮
            ENDLAYOUT;
      ENDLAYOUT;
   ENDLAYOUT;
%MEND PLOTS;
```

DATA AND EXAMPLES:

The following examples use %BXPLOT2 and illustrate the final product. Example 1 illustrates a series of box plots month over month for four process measures for calendar year 2014 for a process. Example 2, is an expansion of example 1 which illustrates four different types of plots to illustrate the flexibility to incorporate different plots. All measures are process times in minutes, but the measures can represent dollars or ratios.

The data used for the examples are for illustrative purposes only and do not represent any process measures for any given entity. Reference lines are hypothetical benchmarks to illustrate the use of them on the plots and therefore do not represent any specific criteria. The data were summarized by the process and month categories for the BLOCKPLOT, BOXPLOTPARM SERIESPLOT and BANDPLOT statements. We followed a similar summarization method as in the SAS 9.2 documentation for the BOXPLOTPARM example, but our example utilizes a categorical variable for the BLOCKPLOT statement whereas the example in the SAS documentation uses individual BLOCKPLOT statements. [2,3]

The data were summarized and grouped in logical blocks. The first block is for the BOXPLOTPARM statement (Table 5), which requires a prefix of "BX_" for the box plots. The second block is for the SERIESPLOT and BANDPLOT statements (Table 6), which does not require a prefix. The third block is for the BLOCKPLOT statement (Table 7), which requires a prefix of "BL_". The data was joined at the process and month categories to be called in by SGRENDER Procedure for each of the plots respectfully.

BX_PROCESS	BX_MTH	BX_STATS	BX_X1	BX_X2	BX_X3	BX_X4
PROCESS_1	201401	MAX	78	186	93	86
PROCESS_1	201401	MEAN	31	60	29	29
PROCESS_1	201401	MEDIAN	19	28	21	21
PROCESS_1	201401	MIN	1	0	0	0
PROCESS_1	201401	N	252	355	432	129
PROCESS_1	201401	Q1	9	5	8	9
PROCESS_1	201401	Q3	37	78	42	41
PROCESS_1	201401	UIF	82	197	93	90
PROCESS_1	201401	UOF	125	302	145	146

Table 5: Layout of summary statistics for BOXPLOTPARM

PROCESS	MTH	X1	X1_ LCLM	X1_ UCLM	…	X4	X4_ LCLM	X4_ UCLM
PROCESS_1	201401	31	26.3	35.1	…	$\overline{x4}_1$	$x4\_LCLM_1$	$x4\_UCLM_1$
PROCESS_1	201402	34	28.7	39.7	…	$\overline{x4}_2$	$x4\_LCLM_2$	$x4\_UCLM_2$
PROCESS_1	201403	31	25.6	36.1	…	$\overline{x4}_3$	$x4\_LCLM_3$	$x4\_UCLM_3$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
PROCESS_1	201412	32	25.8	37.2	…	$\overline{x4}_{12}$	$x4\_LCLM_{12}$	$x4\_UCLM_{12}$

Table 6: Layout of summary statistics for SERIESPLOT and BANDPLOT

BL_PROCESS	BL_MTH	BL_STATS	BL_X1	BL_X2	BL_X3	BL_X4
PROCESS_1	201401	1: # Cases	252	355	432	129
PROCESS_1	201401	2: 25th Pctl	9	5	8	9
PROCESS_1	201401	3: Median	19	28	21	21
PROCESS_1	201401	4: 75th Pctl	37	78	42	41
PROCESS_1	201401	5: Mean	31	60	29	29

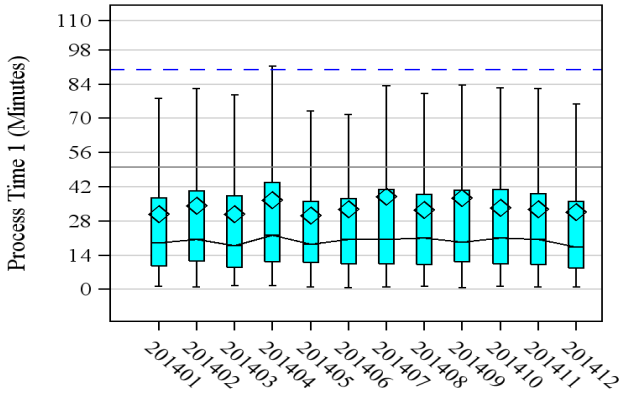
Table 7: Layout of summary statistics for BLOCKPLOT

Example 1: The following plots provide a series of modified box plots over time for four different measures. The diamonds are the means, the line connects the medians. We see the means are higher than the medians, which indicate the data are skewed to the right indicating the data are not normally distributed. Notice how the shaded region (Inner Quartile Range) shifts up and down and expands and contracts for each plot. In Plot 2, the mean is higher than the 75th percentile and the median decreases for December 2014. Plots 3 and 4 both illustrate the mean and medians are decreasing over time as are the shaded regions.

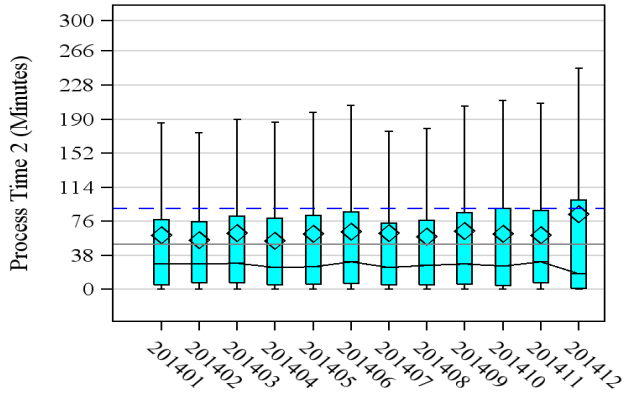
So what is the importance? Why not simply observe a trend of the means? Suppose in plot 2, there were events that spiked the average and had a high financial risk associated with it. Would the series plot display those events? The box plots with the inclusion of the block plots are more exploratory in nature and illustrate the central tendency, distribution, potential outliers and volume, which provide additional information to allow the stakeholders to quickly visualize the variation in the data.

Combined Process Times January 2014 to December 2014

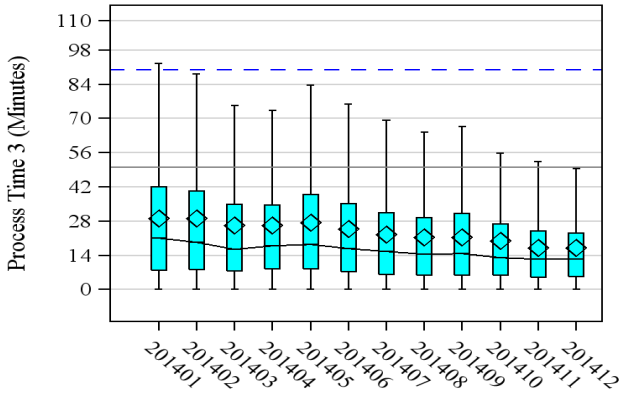
	201401	201402	201403	201404	201405	201406	201407	201408	201409	201410	201411	201412
1: # Cases	252	257	254	255	254	258	253	251	253	253	250	253
2: 25th Pctl	9	11	9	11	11	11	10	10	11	10	10	9
3: Median	19	21	18	22	18	20	20	21	19	21	21	17
4: 75th Pctl	37	40	38	44	36	37	41	39	41	41	39	36
5: Mean	31	34	31	37	30	33	38	33	38	33	33	32



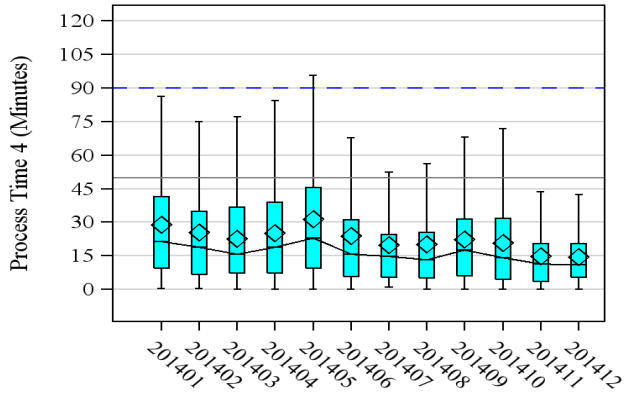
	201401	201402	201403	201404	201405	201406	201407	201408	201409	201410	201411	201412
1: # Cases	355	359	352	352	357	358	353	356	356	359	350	352
2: 25th Pctl	5	7	8	5	6	6	5	5	6	4	7	1
3: Median	28	28	29	25	26	31	24	27	29	26	31	18
4: 75th Pctl	78	76	82	79	83	86	74	77	86	90	88	100
5: Mean	60	55	63	55	62	65	63	59	65	62	61	84



	201401	201402	201403	201404	201405	201406	201407	201408	201409	201410	201411	201412
1: # Cases	432	426	434	435	435	433	428	427	434	430	435	426
2: 25th Pctl	8	8	8	8	9	7	6	6	6	6	5	5
3: Median	21	19	17	18	19	17	16	15	15	13	12	12
4: 75th Pctl	42	40	35	35	39	35	31	29	31	27	24	23
5: Mean	29	29	26	26	27	25	22	21	21	20	17	17

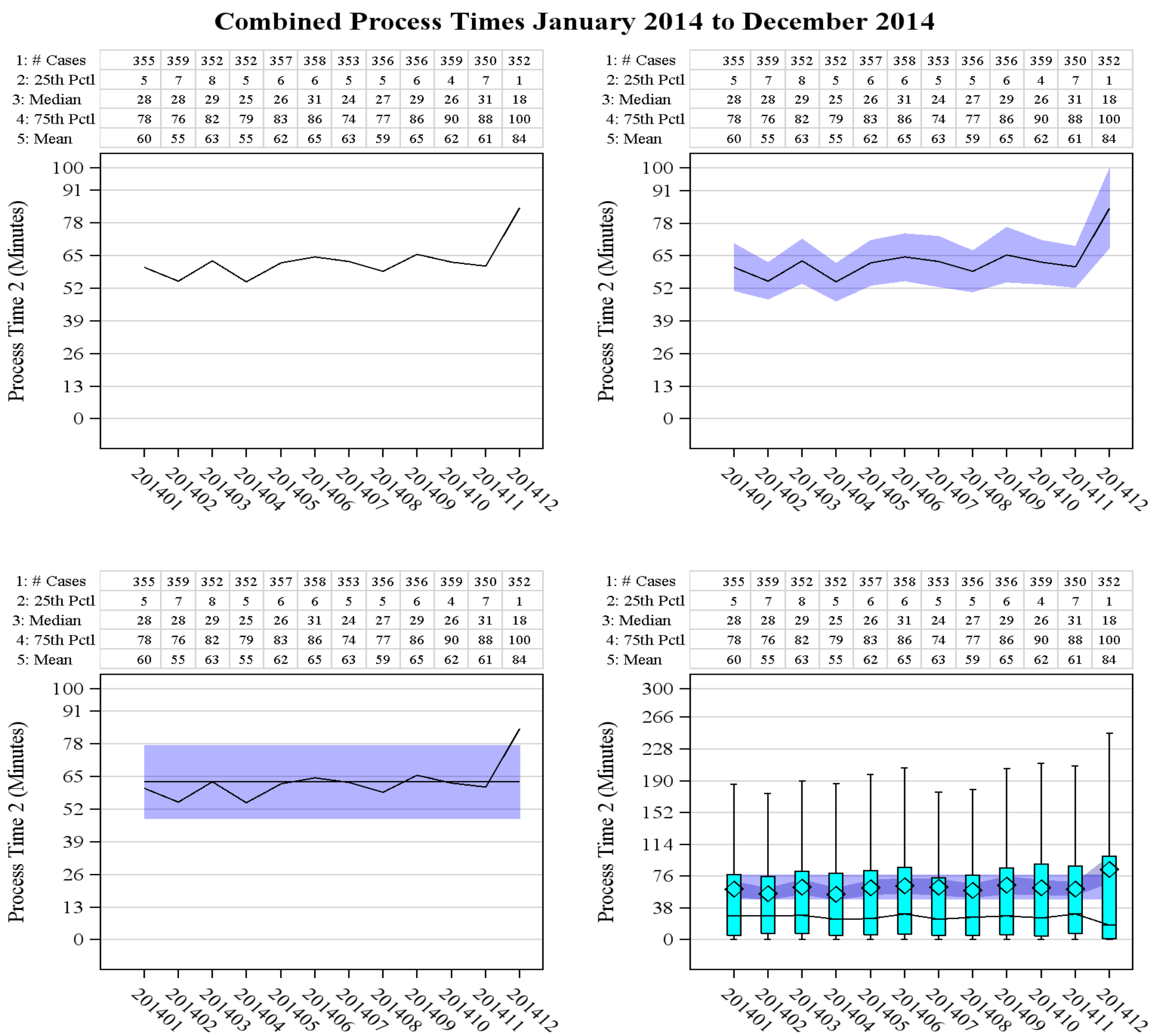


	201401	201402	201403	201404	201405	201406	201407	201408	201409	201410	201411	201412
1: # Cases	129	122	126	127	123	129	123	127	122	122	129	120
2: 25th Pctl	9	7	7	7	9	6	5	5	6	5	4	5
3: Median	21	19	16	19	23	16	15	13	18	14	11	11
4: 75th Pctl	41	35	37	39	45	31	24	25	31	32	20	20
5: Mean	29	25	23	25	31	24	20	20	22	21	15	15



Graphs contain modified box plots

Example 2: As an extension of example 1, %BXPLOT2 can generate series plots and band plots. Plot 1 provides a series plot of the means. Plot 2 expands on Plot 1 by including 95% confidence intervals, which illustrate variation about the mean. Plots 3 and 4 are sourced off of summary statistics from the SHEWHART Procedure using the BOXPLOT statement. Plot 3 contains the control limits for the twelve periods of data where the observed mean exceeded the (3σ) upper control limit in December 2014. Plot 4 contains box plots as in example 1 and overlays the 95% confidence intervals from plot 2 and (3σ) control limits from plot 3 to illustrate the inclusion of multiple band plots on a chart. Plots 2 and 3 provide more information than Plot 1 and have their purposes and are effective when used appropriately, but as indicted in example 1 plot 4 display the distribution of the data and provide opportunities to identify potential outliers quickly, which in conjunction of other plots can help lead to alternative methods to address performance improvement efforts.



Graphs contain modified box plots

CONCLUSION

In the overview we broke down the statements and associated options involved in the macro %BXPLOT2. The purpose of breaking the code down was to help sort through the full code and provide a general overview of the statements and options used.

The examples illustrate how nesting the LAYOUT LATTICE coupled with the LAYOUT OVERLAY within GTL can produce custom reports. The second example illustrates how we can interchange and overlay plots. Although the examples were for a single process, %BXPLOT2 can also be used to observe multiple processes simultaneously for a given set of measures for a given period of time, i.e., month, quarter, semi-annual or annual, which can be performed by putting the process variable along the x-axis and restrict for a given time period. In our examples we have summary statistics per month, if the desire is to summarize per quarter, 6 months or year, there is a need to aggregate to those respective levels.

While our efforts were specific in nature to summarize our data using a nested Layout Lattice in GTL, there is room for enhancements and acknowledge there are alternative ways to approach this problem. A potential avenue to explore is to imbed multiple GTL plots within ODS LAYOUT with the GRIDDED options, which may be simpler and a bit more flexible. Additionally, if SAS ADDINS to MSOFFICE is available, could also explore the option to embed the GTL plots in Power Point or Word Document and allow the user to update the graphs as data is refreshed. We strictly focused on summary level data, but it may be worthwhile exploring merging the summary level data for the BLOCKPLOT with the detailed level data for other PLOT statements such as BOXPLOT, SCATTERPLOT, LOESSPLOT, etc. Lastly we focused on continuous level data for the box plots, where further exploration is needed to incorporate plots for categorical data. In spite of some limitations, the examples illustrate how GTL provides a way to customize reports and complements the other statistical plot procedures such as SGPLOT and SGPANNEL.

REFERENCES

1.) SAS/GRAPH® 9.3 Graph Template Language Reference. Second Edition
2.) SAS(R) 9.3 Graph Template Language: Reference, Third Edition
3.) SAS/GRAPH(R) 9.2: Graph Template Language Reference, Second Edition
4.) http://support.sas.com/kb/39/132.html
5.) Matange, Sanjay. Introduction to the Graph Template Language
6.) Kuhfeld Warren F., The Graph Template Language and the Statistical Graphics Procedures: An Example-Driven Introduction SAS Global Forum 2010

CONTACT INFORMATION

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

Greg Stanek MS
hgs90@yahoo.com

APPENDIX

The following code is for Macro %BXPLOT2. The PLOT Statements for (GTL) were pulled out into individual macros to compartmentalize them in order to make adjustments easier. Similarly the Macro %PLOTS was developed to set up the plots and to be called into Macro %BXPLOT2 one time versus repeating it within the Macro %BXPLOT2 multiple times.

Macro %BXPLOT2 additionally generates macro variables for the axis options for the plots and allows us to generate plots for a process over time or to generate plots of a set of processes within a given time period. Currently %BXPLOT2 requires that the input dataset to be summarized for the BLOCKPLOT and BOXPLOTPARM, SERIESPLOT and BANDPLOT statements. SAS has some general code developed under the GTL BOXPLOTPARM documentation where they illustrate how to incorporate the BLOCKPLOT statement. Again, one of the differences between their example and the one listed below is that they define separate BLOCKPLOTS where we generate a categorical variable called by the class option under the BLOCKPLOT Statement. The parameters are listed below followed by the code.

```
 % BXPLOT2(
INDAT=Input Dataset,
Y1=Y-Axis Measure variable for top right quadrant,
Y2= Y-Axis Measure variable for top left quadrant,
Y3= Y-Axis Measure variable for bottom right quadrant,
Y4= Y-Axis Measure variable for bottom left quadrant,
PLOT_TYPE1=Type of Chart for Y1 (SERIES, BOXPLOT, etc.),
PLOT_TYPE2= Type of Chart for Y2 (SERIES, BOXPLOT, etc.),
PLOT_TYPE3= Type of Chart for Y3 (SERIES, BOXPLOT, etc.),
PLOT_TYPE4= Type of Chart for Y4 (SERIES, BOXPLOT, etc.),
x=X-Axis variable,
YR1=Y-Axis Reference Line 1,
YR2= Y-Axis Reference Line 2,
HT=Adjust the height of the report. Default 1000px,
WDT=Adjust the width of the report. Default 900px,
SZ=Adjusts the size of the font default 5,
RSTRCT=Restriction criteria for data= in SGRENDER,
ENTTL=Title Name for Report for Entry Title in GTL,
ENTFT=Footer Name for Report for Entry Footer in GTL,
DSPL=Display options for the BOXPLOTPARM Statement (Connect, Median, Means, CAPS, Fill)
LBL1= Label for Measure variable for top right quadrant,
LBL2= Label for  Measure variable for top left quadrant,
LBL3= Label for  Measure variable for bottom right quadrant,
LBL4= Label for  Measure variable for bottom left quadrant
NO_GRPS=Number of BINS for the Y-Axis);
```

%BXPLOT2 SYNTAX:

```
OPTIONS NODATE NONUMBER NOMPRINT NOMLOGIC NOSYMBOLGEN ORIENTATION=PORTRAIT;

/*MACROS FOR VARIOUS PLOTS  TO BE CALLED INTO MACRO PLOTS.*/

   %MACRO SERIES_MNS;
      SERIESPLOT Y=&&Y&J. X=&X.;
   %MEND SERIES_MNS;

   %MACRO SERIES_BAND;
/*CONFIDENCE INTERVALS*/
      BANDPLOT X=&X LIMITUPPER= &&Y&J._UCLM
         LIMITLOWER= &&Y&J._LCLM / TYPE=SERIES DATATRANSPARENCY=.7 FILLATTRS=(COLOR=BLUE) EXTEND=TRUE;
      SERIESPLOT Y=&&Y&J. X=&X.;
   %MEND SERIES_BAND;

   %MACRO BXPLOT;
      BOXPLOTPARM Y=BX_&&Y&J. X=BX_&X. STAT=BX_STATS /CONNECT=MEDIAN DISPLAY=&DSPL. FILLATTRS=(COLOR=CYAN) DATALABEL=DATALABEL ;
   %MEND BXPLOT;

   %MACRO BX_BAND;
      BANDPLOT X=&X LIMITUPPER= &&Y&J._UCLM
         LIMITLOWER= &&Y&J._LCLM / TYPE=SERIES  DATATRANSPARENCY=.7 FILLATTRS=(COLOR=BLUE) EXTEND=TRUE;
      BOXPLOTPARM Y=BX_&&Y&J. X=BX_&X. STAT=BX_STATS /CONNECT=MEDIAN DISPLAY=&DSPL. DATALABEL=DATALABEL FILLATTRS=(COLOR=CYAN);
   %MEND BX_BAND;

   %MACRO BX_XCHT;
/*CONFIDENCE INTERVALS*/
      BANDPLOT X=&X LIMITUPPER= &&Y&J._UCLM
         LIMITLOWER= &&Y&J._LCLM / TYPE=SERIES  DATATRANSPARENCY=.7 FILLATTRS=(COLOR=DARKBLUE) EXTEND=TRUE;
/*CONTROL LIMITS*/
      BANDPLOT X=&X LIMITUPPER= &&Y&J._UCLX
         LIMITLOWER= &&Y&J._LCLX / TYPE=SERIES  DATATRANSPARENCY=.7 FILLATTRS=(COLOR=BLUE) EXTEND=TRUE;
      BOXPLOTPARM Y=BX_&&Y&J. X=BX_&X. STAT=BX_STATS /CONNECT=MEDIAN DISPLAY=&DSPL. DATALABEL=DATALABEL FILLATTRS=(COLOR=CYAN);
   %MEND BX_XCHT;

   %MACRO XCHART;
/*CONTROL LIMITS*/
      BANDPLOT X=&X LIMITUPPER=&&Y&J._UCLX
         LIMITLOWER=&&Y&J._LCLX/ TYPE=STEP DATATRANSPARENCY=.7 FILLATTRS=(COLOR=BLUE) EXTEND=TRUE;
      SERIESPLOT Y=&&Y&J. X=&X.;
      SERIESPLOT Y=&&Y&J._PROCMEAN X=&X./LINEATTRS=(PATTERN=SOLID COLOR=BLACK);
   %MEND XCHART;

/*MACRO %PLOTS COMBINES THE BLOCKPLOT AND PLOTS EMBEDED IN PREVIOUS STATED MACROS AND IS CALLED INTO MACRO %BXPLOT2. */

   %MACRO PLOTS;
      LAYOUT OVERLAY;
         LAYOUT LATTICE/COLUMNS=1 ROWWEIGHTS=(.20 .80) ROWGUTTER=2;
            COLUMNAXES; COLUMNAXIS/GRIDDISPLAY=ON LABEL=' '; ENDCOLUMNAXES;

/* DEFINE AREA FOR BLOCKPLOT STATEMENT FOR EACH ROW AND COLUMN IN %BXPLOT2*/
               BLOCKPLOT X=BL_&X. BLOCK=BL_&&Y&J. /CLASS=BL_STATS
                  INCLUDEMISSINGCLASS=FALSE REPEATEDVALUES=TRUE
                  DISPLAY=(VALUES LABEL OUTLINE ) OUTLINEATTRS=(COLOR=LIGHTGRAY) VALUEHALIGN=CENTER
                  LABELATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+1)PT) VALUEATTRS=GRAPHDATATEXT(SIZE=&SZ.PT);

/* DEFINE AREA FOR THE PLOT STATEMENTS WITH LAYOUT OVERLAY AND LAYOUT LATTICE FOR EACH ROW AND COLUMN IN %BOXPLOT2 */
               LAYOUT OVERLAY /XAXISOPTS=(&OFFSETS LABEL=' ' TICKVALUEATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+2)PT))
                  YAXISOPTS=( &OFFSETS GRIDDISPLAY=ON GRIDATTRS=(COLOR=LIGHTGRAY) LABEL="&&LBL&J."
                  LABELATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+3)PT)
                  TICKVALUEATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+2)PT)
                  LINEAROPTS=(TICKVALUESEQUENCE=(START=&&MIN_Y&J. END=&&MAX_Y&J. INCREMENT=&&INCRMT_Y&J.)
                  TICKVALUEPRIORITY=TRUE));
```

```
/*CALLING IN PLOTS*/
                        %IF %UPCASE(&&PLOT_TYPE&J.)=SERIES %THEN %DO;
                            %SERIES_MNS
                        %END;
                        %IF %UPCASE(&&PLOT_TYPE&J.)=BXPLOT %THEN %DO;
                            %BXPLOT
                        %END;
                        %IF %UPCASE(&&PLOT_TYPE&J.)=BXBAND %THEN %DO;
                            %BX_BAND
                        %END;
                        %IF %UPCASE(&&PLOT_TYPE&J.)=BXXCHT %THEN %DO;
                            %BX_XCHT
                        %END;
                        %IF %UPCASE(&&PLOT_TYPE&J.)=XCHART %THEN %DO;
                            %XCHART
                        %END;
                        %IF %UPCASE(&&PLOT_TYPE&J.)=SERIES_BAND %THEN %DO;
                           %SERIES_BAND
                        %END;
                        %IF &YR1. NE ' ' %THEN %DO;
                           REFERENCELINE Y=&YR1./LINEATTRS=(PATTERN=DASH COLOR=BLUE);
                        %END;
                        %IF &YR2. NE ' ' %THEN %DO;
                           REFERENCELINE Y=&YR2./LINEATTRS=(PATTERN=SOLID );
                        %END;
            ENDLAYOUT;
          ENDLAYOUT;
       ENDLAYOUT;
   %MEND PLOTS;

/*MACRO %BXPLOT2 CALLS IN MACRO %PLOTS TO BUILD OUT REPORT*/

   %MACRO BXPLOT2(INDAT=,Y1=,Y2=,Y3=,Y4=,PLOT_TYPE1=,PLOT_TYPE2=,PLOT_TYPE3=,PLOT_TYPE4=, X=,YR1=' ',YR2=' ', HT=1000PX, WDT=900PX,
   SZ=5, RSTRCT=, ENTTL=, ENTFT=,  DSPL=, LBL1=, LBL2=, LBL3=, LBL4=,NO_GRPS=8);

/*GENERATE GLOBAL MACRO VARIABLES*/

       %DO K=1 %TO 4;
          %GLOBAL  MIN_Y&K. MAX_Y&K. INCRMT_Y&K.;
       %END;

/*GENERATE MACRO VARIABLES FOR TICKVALUESEQUENCE*/
       PROC SQL NOPRINT;
          %DO I=1 %TO 4;
             %IF %UPCASE(&&PLOT_TYPE&I.)=BXPLOT OR %UPCASE(&&PLOT_TYPE&I.)=BXBAND OR %UPCASE(&&PLOT_TYPE&I.)=BXXCHT %THEN %DO;
                SELECT
                   MIN(0,MIN(CASE WHEN BX_STATS='MIN' THEN BX_&&Y&I. END)) AS MIN_Y&I.
                   ,MAX(CASE WHEN BX_STATS='MAX' THEN
                       (CASE
                          WHEN 0<=  (BX_&&Y&I.)< 0       THEN .
                          WHEN 0<= (BX_&&Y&I.)<=1        THEN ROUND((BX_&&Y&I.+.1),.1)
                          WHEN 1<  (BX_&&Y&I.)<100       THEN ROUND((BX_&&Y&I.+20),10)
                          WHEN 100<=(BX_&&Y&I.)<500   THEN ROUND((BX_&&Y&I.+50),100)
                          WHEN 500<=(BX_&&Y&I.)          THEN ROUND((BX_&&Y&I.+500),1000)
                          ELSE 0
                          END)
                   END) AS MAX_Y&I.
                   ,CEIL(CALCULATED MAX_Y&I./&NO_GRPS.) AS INCRMT_Y&I.
                   INTO :MIN_Y&I.,  :MAX_Y&I.,  :INCRMT_Y&I.
            %END;

            %IF %UPCASE(&&PLOT_TYPE&I.)=SERIES  OR %UPCASE(&&PLOT_TYPE&I.)=SERIES_BAND
               OR %UPCASE(&&PLOT_TYPE&I.)=XCHART %THEN %DO;
```

```
            SELECT
               MIN(0,MIN(&&Y&I.)) AS MIN_Y&I.
               ,MAX(CASE
                        WHEN (&&Y&I.)<0              THEN .
                        WHEN 0<= (&&Y&I.)<=1         THEN ROUND((&&Y&I.+.1),.1)
                        WHEN 1<  (&&Y&I.)<100        THEN ROUND((&&Y&I.+20),10)
                        WHEN 100<=(&&Y&I.)<500       THEN ROUND((&&Y&I.+50),100)
                        WHEN 500<=(&&Y&I.)           THEN ROUND((&&Y&I.+500),1000)
                        ELSE 0
               END) AS MAX_Y&I.
               ,CEIL(CALCULATED MAX_Y&I./&NO_GRPS.) AS INCRMT_Y&I.
               INTO :MIN_Y&I.,  :MAX_Y&I.,  :INCRMT_Y&I.
         %END;
            FROM &INDAT. &RSTRCT.;
      %END;
   QUIT;

/*PERCENT OFFSET ON THE BOX PLOT SEREIES.*/
   %LET OFFSETS = OFFSETMIN=0.1 OFFSETMAX=0.05;

/* GTL CODE*/
      PROC TEMPLATE;
         DEFINE STATGRAPH BOXPLOT2;
            BEGINGRAPH;
               ENTRYTITLE "&ENTTL. ";
/*OUTER LAYOUT LATTICE DIVIDES THE PLOT AREA INTO EQUAL QUADRANTS AND DEFINES SPACE BETWEEN PLOTS*/
               LAYOUT LATTICE / ROWS=2 COLUMNS=2 ROWGUTTER=5 COLUMNGUTTER=10 COLUMNWEIGHTS=(.5 .5) ROWWEIGHTS=(.5 .5);

/* SETTING UP PLOT AREA FOR THE BLOCK PLOT AND CALLING MACRO PLOTS WITH LAYOUT OVERLAY AND LAYOUT LATTICE */
                     %DO J=1 %TO 4;
                     %PLOTS;
                     %END;

               ENDLAYOUT;
            ENTRYFOOTNOTE HALIGN=LEFT "&ENTFT." /TEXTATTRS=(SIZE=%EVAL(&SZ.+1)PT STYLE=ITALIC);
         ENDGRAPH;
      END;
   RUN;

/*TURN ON ODS GRAPHICS FOR PLOTS. ADJUST HEIGHT AND WIDTH OF PLOT AREA ";*/
      ODS GRAPHICS ON / BORDER=OFF NOBORDER HEIGHT=&HT. WIDTH=&WDT.;

/*CALLS GTL BOXPLOT2 FOR GRAPHS;*/
      PROC SGRENDER DATA=&INDAT. &RSTRCT. TEMPLATE=BOXPLOT2; RUN;
   %MEND BXPLOT2;
```

```
/*Generate Reports*/

  ODS PATH WORK.TEMPLAT(UPDATE) SASUSER.TEMPLAT(READ) SASHELP.TMPLMST(READ);
  OPTIONS NODATE NONUMBER MPRINT MLOGIC SYMBOLGEN ORIENTATION=PORTRAIT;

  /*Example 1 */

  %BXPLOT2(INDAT=TEMP_ALL,
  Y1=X1,
  Y2=X2,
  Y3=X3,
  Y4=X4 ,
  PLOT_TYPE1=BXPLOT,
  PLOT_TYPE2=BXPLOT,
  PLOT_TYPE3=BXPLOT,
  PLOT_TYPE4=BXPLOT,
  X=MTH,
  YR1=90,
  YR2=50,
  HT=1100PX,
  WDT=1000PX,
  SZ=6,
  RSTRCT=(WHERE=(UPCASE(PROCESS)="PROCESS_1")),
  ENTTL=Combined Process Times January 2014 to December 2014,
  ENTFT=Graphs contain modified box plots,
  DSPL=( MEDIAN MEAN CAPS FILL CONNECT),
  LBL1=Process Time 1 (Minutes) ,
  LBL2=Process Time 2 (Minutes),
  LBL3=Process Time 3 (Minutes) ,
  LBL4=Process Time 4 (Minutes),
  no_grps=8);

  /*Example 2 */

  %BXPLOT2(INDAT=TEMP_ALL,
  Y1=X2,
  Y2=X2,
  Y3=X2,
  Y4=X2,
  PLOT_TYPE1=SERIES,
  PLOT_TYPE2=SERIES_BAND,
  PLOT_TYPE3=XCHART,
  PLOT_TYPE4=BXXCHT,
  X=MTH,
  /*YR1=,*/
  /*YR2=,*/
  HT=1100PX,
  WDT=1000PX,
  SZ=6,
  RSTRCT=(WHERE=(UPCASE(PROCESS)="PROCESS_1")),
  ENTTL=Combined Process Times January 2014 to December 2014,
  ENTFT=Graphs contain modified box plots,
  DSPL=( MEDIAN MEAN CAPS FILL CONNECT),
  LBL1=Process Time 2 (Minutes) ,
  LBL2=Process Time 2 (Minutes),
  LBL3=Process Time 2 (Minutes) ,
  LBL4=Process Time 2 (Minutes),
  no_grps=8);
```

Syntax for tables 3 and 4 for series plot and band plot:

```
/*SERIESPLOT GTL*/
%MACRO SRSPLOT(INDAT=,Y1=,X=,YR1=,YR2=,HT=,WDT=,SZ=,MTH=,RSTRCT=,LBL1=,ENTFT=,NO_GRPS=8);

%GLOBAL MIN_Y1 MAX_Y1 INCRMT_Y1 ;

   PROC SQL NOPRINT;
   SELECT
      MIN(0,MIN(&Y1.)) ,
      CASE
         WHEN MAX(&Y1.)<=60 THEN ROUND(MAX(&Y1.)+20,10)
          WHEN MAX(&Y1.)>60 THEN ROUND(MAX(&Y1.)+20,10)
          WHEN MAX(&Y1.)>=100 THEN ROUND(MAX(&Y1.),100)
         END AS MAX_Y1,
         CEIL(CALCULATED MAX_Y1/&NO_GRPS.) AS INCRMT_Y1
         INTO :MIN_Y1, :MAX_Y1, :INCRMT_Y1
      FROM &INDAT. &RSTRCT.;QUIT;

%LET OFFSETS = OFFSETMIN=0.1 OFFSETMAX=0.05;

   PROC TEMPLATE;
      DEFINE STATGRAPH SRSPLOT;
         BEGINGRAPH;
            LAYOUT OVERLAY/XAXISOPTS=(&OFFSETS LABEL=' ' TICKVALUEATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+2)PT))
               YAXISOPTS=( &OFFSETS GRIDDISPLAY=ON   GRIDATTRS=(COLOR=LIGHTGRAY) LABEL="&LBL1."
                  LABELATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+3)PT) TICKVALUEATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+2)PT)
                  LINEAROPTS=(TICKVALUESEQUENCE=(START=&MIN_Y1. END=&MAX_Y1. INCREMENT=&INCRMT_Y1.) TICKVALUEPRIORITY=TRUE));
                     SERIESPLOT Y=&Y1. X=&X. ;
            ENDLAYOUT;
               ENTRYFOOTNOTE HALIGN=LEFT "&ENTFT." /TEXTATTRS=(SIZE=%EVAL(&SZ.+1)PT STYLE=ITALIC);
         ENDGRAPH;
      END;
   RUN;

ODS GRAPHICS ON / BORDER=OFF NOBORDER HEIGHT=&HT. WIDTH=&WDT.;
PROC SGRENDER DATA=&INDAT. &RSTRCT. TEMPLATE=SRSPLOT;RUN;
%MEND SRSPLOT;

%SRSPLOT (
INDAT=TEMP_ALL,Y1=X1,X=MTH,HT=3IN, WDT=3IN,SZ=8 ,RSTRCT=(WHERE=(UPCASE(PROCESS)="PROCESS_1"))
,LBL1=PROCESS TIME 1 IN MINUTES, ENTFT=TABLE 3: EXAMPLE OF SERIESPLOT STATEMENT);
```

```
/*BANDPLOT GTL*/
%MACRO BNDPLOT(INDAT=,Y1=,X=,YR1=,YR2=,HT=,WDT=,SZ=,MTH=,RSTRCT=,LBL1=,ENTFT=,NO_GRPS=8);

%GLOBAL MIN_Y1 MAX_Y1 INCRMT_Y1 ;

PROC SQL NOPRINT;
SELECT
MIN(0,MIN(&Y1.)) ,
CASE WHEN MAX(&Y1.)<=60 THEN ROUND(MAX(&Y1.)+20,10)
 WHEN MAX(&Y1.)>60 THEN ROUND(MAX(&Y1.)+20,10)
 WHEN MAX(&Y1.)>=100 THEN ROUND(MAX(&Y1.),100)
END AS MAX_Y1,
CEIL(CALCULATED MAX_Y1/&NO_GRPS.) AS INCRMT_Y1
INTO :MIN_Y1,    :MAX_Y1,  :INCRMT_Y1
FROM &INDAT. &RSTRCT.;QUIT;

%LET OFFSETS = OFFSETMIN=0.1 OFFSETMAX=0.05;

PROC TEMPLATE;
DEFINE STATGRAPH SRSPLOT;
BEGINGRAPH;
LAYOUT OVERLAY/XAXISOPTS=(&OFFSETS LABEL=' ' TICKVALUEATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+2)PT))
YAXISOPTS=( &OFFSETS GRIDDISPLAY=ON   GRIDATTRS=(COLOR=LIGHTGRAY) LABEL="&LBL1."
LABELATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+3)PT) TICKVALUEATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+2)PT)
LINEAROPTS=(TICKVALUESEQUENCE=(START=&MIN_Y1. END=&MAX_Y1. INCREMENT=&INCRMT_Y1.)
TICKVALUEPRIORITY=TRUE));
BANDPLOT X=&X LIMITUPPER= &Y1._UCLM
LIMITLOWER= &Y1._LCLM / TYPE=SERIES DATATRANSPARENCY=.7 FILLATTRS=(COLOR=BLUE)
 EXTEND=TRUE;
SERIESPLOT Y=&Y1. X=&X. ;
ENDLAYOUT;
ENTRYFOOTNOTE HALIGN=LEFT "&ENTFT." /TEXTATTRS=(SIZE=%EVAL(&SZ.+1)PT STYLE=ITALIC);
ENDGRAPH;
END;
RUN;

ODS GRAPHICS ON / BORDER=OFF NOBORDER HEIGHT=&HT. WIDTH=&WDT.;
PROC SGRENDER DATA=&INDAT. &RSTRCT. TEMPLATE=SRSPLOT;RUN;

ODS GRAPHICS ON / RESET=ALL;
%MEND BNDPLOT;

%BNDPLOT (INDAT=TEMP_ALL,Y1=X1 ,X=MTH,HT=3IN,WDT=3IN,SZ=8, RSTRCT=(WHERE=(UPCASE(PROCESS)="PROCESS_1"))
,LBL1=PROCESS TIME 1 IN MINUTES,ENTFT=TABLE 4: EXAMPLE OF BANDPLOT STATEMENT);
```