Nesting Multiple Box Plots and BLOCKPLOTS using GTL and Lattice Overlay

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,
  ENTTF=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/
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. 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. 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. 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.

%MACRO BLOCKPLOT(INDAT=,Y1=,X=,HT=100PX,WDT=9000PX,ZS=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=CENTE
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);
  ENDDGRAPH;
END;
RUN;
ODS GRAPHICS ON / BORDER=OFF NOBORDER HEIGHT=&HT WIDTH=&WDT;
PROC SGRRENDER 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);

<table>
<thead>
<tr>
<th></th>
<th># Cases</th>
<th>259</th>
<th>256</th>
<th>259</th>
<th>250</th>
<th>256</th>
<th>259</th>
<th>256</th>
<th>251</th>
<th>256</th>
<th>254</th>
<th>253</th>
<th>252</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25th Pctl</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>11</td>
<td>11</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Median</td>
<td>18</td>
<td>20</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>21</td>
<td>20</td>
<td>18</td>
<td>20</td>
<td>19</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>75th Pctl</td>
<td>37</td>
<td>39</td>
<td>36</td>
<td>38</td>
<td>39</td>
<td>37</td>
<td>43</td>
<td>34</td>
<td>39</td>
<td>39</td>
<td>40</td>
<td>34</td>
</tr>
<tr>
<td>5</td>
<td>Mean</td>
<td>31</td>
<td>34</td>
<td>36</td>
<td>33</td>
<td>33</td>
<td>30</td>
<td>35</td>
<td>31</td>
<td>30</td>
<td>31</td>
<td>35</td>
<td>32</td>
</tr>
</tbody>
</table>

Table 1: Example of BLOCKPLOT Statement
The BOXPLOT 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 BOXPLOT 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 BOXPLOT statement 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` 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. 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=)

%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.) END)
ELSE 0 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=(&OFSETS 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;

%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).

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).

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

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

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.

Table 3: Example of SERIESPLOT Statement

Table 4: Example of BANDPLOT Statement
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 COLUMNSGUTTER= options adjust the space between the plot regions where the COLUMNWEIGHTS=( ) and ROWWEIGHTS=( ) options assign a weight to the column and rows for the plot area, where weights are between \([0,1]\).

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:

\[
\begin{align*}
\%MACRO \ BXPLOT2(INDAT=,Y1=,Y2=,Y3=,Y4=,PLOT\_TYPE1=,PLOT\_TYPE2=,PLOT\_TYPE3=,PLOT\_TYPE4=,X=,YR1=\ ' , YR2=\ ' , HT=1000\,PX, WDT=900\,PX, SZ=5, RSTRCT=, ENTTL=, ENTFT=, DSPL=,LBL1=,LBL2=,LBL3=,LBL4=, NO\_GRPS=8); \\
PROC TEMPLATE; \\
DEFINE STATGRAPH BOXPLOT2; \\
BEGINGRAPH; \\
\textbf{ENTRYTITLE} "&ENTTL."; \\
\textbf{LAYOUT LATTICE} / ROWS=2 COLUMNS=2 ROWGUTTER=5 COLUMNGUTTER=10 COLUMNWEIGHTS=(.5 .5) ROWWEIGHTS=(.5 .5); \\
\%DO J=1 %TO 4; \\
\%PLOTS; \\
\%END; \\
\%ENDLAYOUT; \\
ENTRYFOOTNOTE HALIGN=LEFT "&ENTFT." /TEXTATTRS=(SIZE=%EVAL(&SZ.+1)PT STYLE=ITALIC); \\
ENDGRAPH; \\
END; \\
\%MEND BXPLOT2; 
\end{align*}
\]

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):
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.

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

Table 5: Layout of summary statistics for BOXPLOTPARM

<table>
<thead>
<tr>
<th>PROCESS</th>
<th>MTH</th>
<th>X1</th>
<th>X1_LCLM</th>
<th>X1_UCLM</th>
<th>X4</th>
<th>X4_LCLM</th>
<th>X4_UCLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS_1</td>
<td>201401</td>
<td>31</td>
<td>26.3</td>
<td>35.1</td>
<td>(\bar{x}_4)</td>
<td>(x_{4LCLM})</td>
<td>(x_{4UCLM})</td>
</tr>
<tr>
<td>PROCESS_1</td>
<td>201402</td>
<td>34</td>
<td>28.7</td>
<td>39.7</td>
<td>(x_4)</td>
<td>(x_{4LCLM})</td>
<td>(x_{4UCLM})</td>
</tr>
<tr>
<td>PROCESS_1</td>
<td>201403</td>
<td>31</td>
<td>25.6</td>
<td>36.1</td>
<td>(\bar{x}_4)</td>
<td>(x_{4LCLM})</td>
<td>(x_{4UCLM})</td>
</tr>
</tbody>
</table>

Table 6: Layout of summary statistics for SERIESPLOT and BANDPLOT

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

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

<table>
<thead>
<tr>
<th></th>
<th># Cases</th>
<th>1/25th Pctl</th>
<th>2/25th Pctl</th>
<th>3/Med</th>
<th>4/75th Pctl</th>
<th>5/Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plot</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>232, 257, 254, 255, 254, 255, 258, 253, 251, 253, 250, 253</td>
<td>9, 11, 9, 11, 11, 10, 10, 11, 10, 10, 9</td>
<td>19, 21, 18, 21, 18, 20, 20, 21, 19, 21, 21</td>
<td>37, 40, 38, 41, 37, 41, 39, 41, 41, 39, 36</td>
<td>31, 34, 31, 37, 30, 33, 38, 33, 33, 33, 32</td>
<td></td>
</tr>
</tbody>
</table>
Example 2: As an extension of example 1, %BXPLT2 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 indicated 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.
CONCLUSION

In the overview we broke down the statements and associated options involved in the macro %BXPLT2. 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, %BXPLT2 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 ORIENTA
TION=PORTRAIT;

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

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

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

%MACRO BX_XCHT;
/*CONTROL LIMITS*/
BANDPLOT X=&X LIMITUPPER= &Y& LIMITLOWER= &Y& /TYPE=STEP DATATRANSPARENCY=.7 FILLATTRS=(COLOR=BLUE) EXTEND=TRUE;
SERIESPLOT Y=&Y& X=&X;
%MEND BX_XCHT;

%MACRO XCHART;
/*CONTROL LIMITS*/
BANDPLOT X=&X LIMITUPPER= &Y& LIMITLOWER= &Y& /TYPE=STEP DATATRANSPARENCY=.7 FILLATTRS=(COLOR=BLUE) EXTEND=TRUE;
SERIESPLOT Y=&Y& 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; COLUMMAXIS/GRI
NDDISPLAY=ON LABEL=""; ENDCOLUMMAX
ES;

/* DEFINE AREA FOR BLOCKPLOT STATEMENT FOR EACH ROW AND COLUMN IN %BXPLOT2*/
BLOCKPLOT X=BL_&X. BLOCK=BL_&Y&/CLASS=BL_STATS INCLUDEMISSINGCLASS=FALSE REPEATEDVALUES=TRUE DISPLAY=(VALUES LABEL OUTLINE ) OUTLINEATTRS=(COLOR=LIGHTGRAY) VALUEHA
ALIGN=center LABELATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+1)PT) VALUEATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+2)PT);

/* DEFINE AREA FOR THE PLOT STATEMENTS WITH LAYOUT OVERLAY AND LAYOUT LATTICE FOR EACH ROW AND COLUMN IN %BXPLOT2 */
LAYOUT OVERLAY /AXISOPTS=(OFFSETS LABEL="" TICKVALUEATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+2)PT) YAXISOPTS=(OFFSETS GRIDDISPLAY=ON GRIDATTRS=(COLOR=LIGHTGRAY) LABEL="" &LBL&") LABELATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+3)PT)
TICKVALUEATTRS=GRAPHDATATEXT(SIZE=%EVAL(&SZ.+2)PT) LINEAROPTS=(TICKVALUESEQUENCE=(START=&MIN_Y& END=&MAX_Y& INCREMENT=&INCRMT_Y&)
TICKVALUEPRI
ORITY=TRUE));
/*CALLING IN PLOTS*/
%IF %UPCASE(&&PLOT_TYPE&J.)=SERIES %THEN DO;
  SERIES_MNS
%END;
%IF %UPCASE(&&PLOT_TYPE&J.)=BXPLOT %THEN DO;
  BX_PLOT
%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;
  REFERENCENAME Y=&&YR1./LINEATTRS=(PATTERN=DASH COLOR=BLUE);
%END;
%IF &&YR2. NE "" %THEN DO;
  REFERENCENAME 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 BX_&&Y&I-END) AS MAX_Y&I;
    ,CEIL(CALCULATED MAX_Y&I./&NO_GRPS.) AS 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;
END;

*/
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.
QUIT;

/*PERCENT OFFSET ON THE BOX PLOT SERIES.*/
%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=%EV(AL(&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")),
ENTTI=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=xXCHT,
X=MTH,
/*YR1=,*/
/*YR2=,*/
HT=1100PX,
WDT=1000PX,
SZ=6,
RSTRCT=(WHERE=(UPCASE(PROCESS)="PROCESS_1")),
ENTTI=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=,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*/
%MATERIAL BNDPLOT(INDAT=,Y1=,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 DATATRANSPRENCY=.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;
%MATERIAL 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);

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.