

SAS Graph: Introduction to the World of Boxplots

Brian Spruell, Constella Group LLC, Durham, NC

ABSTRACT

Boxplots provide a graphical representation of a data's distribution. Every elementary statistical course includes an introduction to the construction and use of boxplots. Although boxplots prove to be quite useful they tend to be somewhat tedious to create. Thankfully, SAS graph comes equipped with the ability to generate boxplots. This paper will serve as an introduction to SAS graph boxplots. It will show several techniques the author has learned to manipulate SAS into generating a boxplot to conform to user specifications. Some topics covered in this paper include displaying several boxplots on one graph, median connectors, and horizontal boxplots. The paper will also demonstrate modifying a boxplot with annotate.

INTRODUCTION

You can graphically display summary data with boxplots. The most common boxplots show the data's median, first quartile, third quartile, minimum and maximum data points. Boxplots provide a pretty good picture of a data's variation. You can use a boxplot to help detect whether data is skewed or flag a potential outlier. Boxplots do tend to be somewhat tedious to produce by hand. Thankfully SAS graph comes equipped with the ability to generate boxplots. I will start with simple examples and eventually progress to more complex ones.

STANDARD BOXPLOT

The following table displays all the points scored by the 2006 North Carolina State WolfPack basketball team. The team played a total of thirty four games. The table only lists opponents and the amount of points NCSU scored and the outcome of the game from North Carolina State's perspective:

Opponent	NCSU points	Outcome
Univ. of Guelph (Exh.)	75	W
Mount Olive (Exh.)	97	W
Stetson (Hisp. College Fund Class.)	91	W
Citadel (Hisp. College Fund Class.)	91	W
Delaware(Hisp. College Fund Class.)	73	W
VMI	75	W
Notre Dame (Wooden Classic)	61	W
Iowa (ACC/Big 10 Challenge)	42	L
App. State (Reynolds Coliseum)	92	W
UNC-Asheville	86	W
Miami	81	W
Alabama	68	W
New Hampshire	81	W
George Washington	79	W
UNC-Greensboro	83	W
North Carolina	69	L
Boston College	78	W
Georgia Tech	87	W
Duke	68	L
Wake Forest	92	W
Seton Hall	65	L
Clemson	94	W
Virginia	66	W
Maryland	62	W
Miami	87	W
Georgia Tech	68	L
Florida State	86	W
Virginia Tech	70	W
North Carolina	95	L
Boston College	74	L
Wake Forest	63	L
Wake Forest (ACC Tournament)	71	L

Cal (NCAA Tournament)	58	W
Texas (NCAA Tournament)	54	L

I developed the following piece of sas code to display the basketball data with a boxplot

```
GOPTIONS  RESET = all GSFNAME = gsasfile GSFMODE = replace
          ROTATE=landscape FTEXT = swiss CBACK = white
          TARGETDEVICE= HPLJGL2 dev=jpeg
          xmax=11in ymax=8.5in xpixels=3300 ypixels=2550;

filename gsasfile "D:\My Documents\SEGUI\OCT2006\SESUGI1.jpeg";

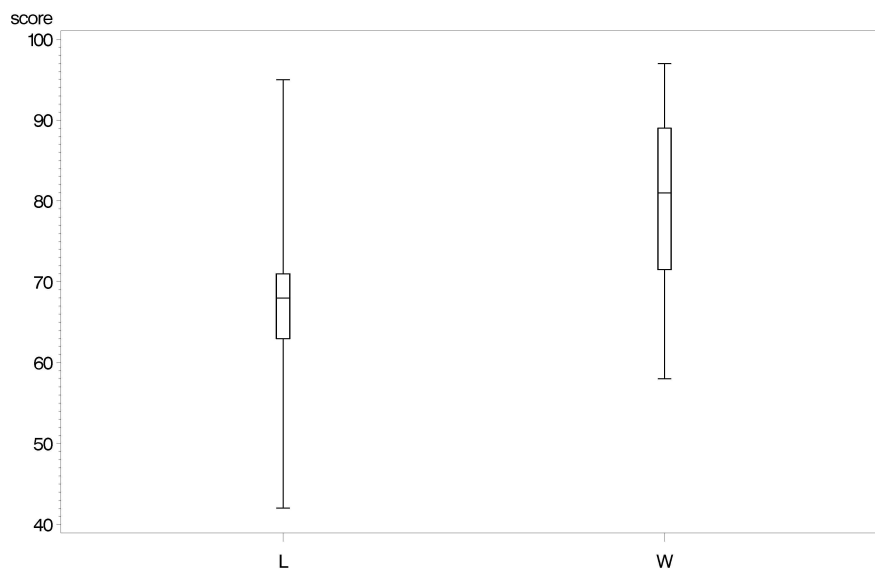
symbol1 interpol=BOXT00 color=black width=3 bwidth=2;
title h=1.7 'Total Breakdown of Points vs Outcome';

axis1 label=(h=1.2 ' ') minor=none offset=(25 pct);

proc gplot data=ncsu06;
    plot score*outcome/ haxis=axis1;
run;
quit;
```

The 'symbol1 interpol=BOXT00' tells SAS I want a boxplot to be produced by my proc gplot statement. I added some axis options to make the graph more presentable.

Total Breakdown of Points vs Outcome



BOXPLOT OPTIONS

Several SAS graph options exist which you can use to modify the appearance of your boxplot. The most noticeable modification involves changing the appearance of the line connectors, also known as 'whiskers'. The default setting, 'interpol=BOX', will only draw a connector line from the box to 1.5 times the interquartile range (IQR). The interquartile range is the difference between the 75th and 25th percentiles. Data points outside 1.5 times the IQR are classified as potential outliers. Modifying the line connectors involves making changes to the Interpol statement by adding a value to the end of 'BOX'. SAS boxplot values range from '00' to '25'. In the example I presented earlier I specified the option '00'. I added it to the end of 'interpol=BOXT'. This option tells SAS to draw the connector lines from the box (25th and 75th percentiles) to the minimum and maximum values respectively. Using '25' as an option prevents the drawing of the line

connectors. In those instances only the box is displayed. '05' produces a boxplot with the connectors going from 5th percentiles lowest to 95th percentile highest.

Modifying the appearance of the line connectors may cause several data points to fall outside your range. Data points that fall outside your line connector range are marked by a plot symbol. No data falls outside the '00' option since the line connectors go from the quartiles to the minimum and maximum values within a dataset. The plot symbol is designated within the symbol statement. In the examples to follow it is given a value of 'circle'. You can control the color and size of plot symbols by specifying both an 'h' and 'cv' after 'value='.

The graphs below (along with the code which produced them with differences bolded) demonstrate modifications to a boxplot's line connectors:

Options '10':

```
GOPTIONS RESET = all GSFNAME = gsasfile GSFMODE = replace
          ROTATE=landscape FTEXT = swiss CBACK = white
          TARGETDEVICE= HPLJGL2 dev=jpeg
          xmax=11in ymax=8.5in xpixels=3300 ypixels=2550;

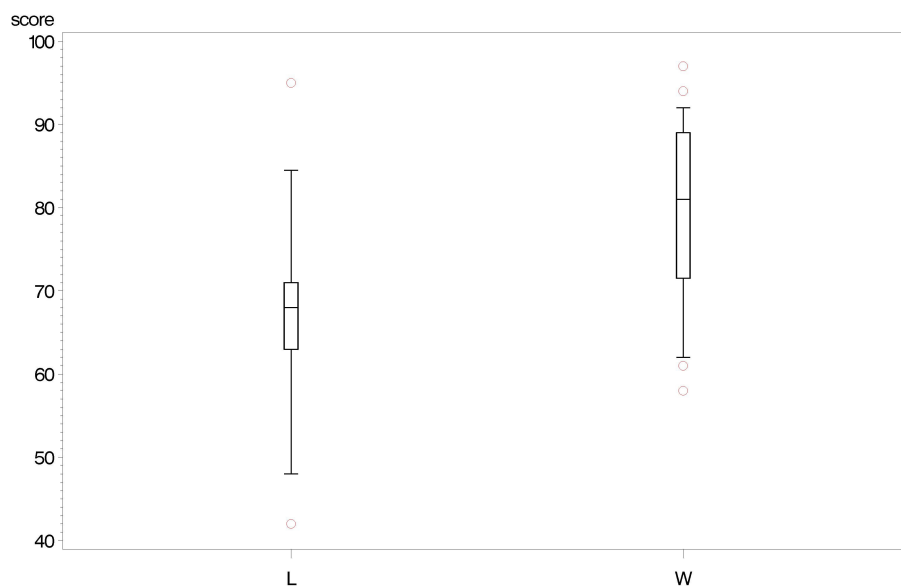
symbol11 interp=BOXT10 color=black width=3 bwidth=2 value=circle
          cv=red height=1;

title h=1.7 'Total Breakdown of Points vs Outcome';

axis1 label=(h=1.2 ' ') minor=none offset=(25 pct);

proc gplot data=ncsu06;
          plot score*outcome/ haxis=axis1;
run;quit;
```

Total Breakdown of Points vs Outcome



Options '25':

```
GOPTIONS RESET = all GSFNAME = gsasfile GSFMODE = replace
          ROTATE=landscape FTEXT = swiss CBACK = white
          TARGETDEVICE= HPLJGL2 dev=jpeg
          xmax=11in ymax=8.5in xpixels=3300 ypixels=2550;

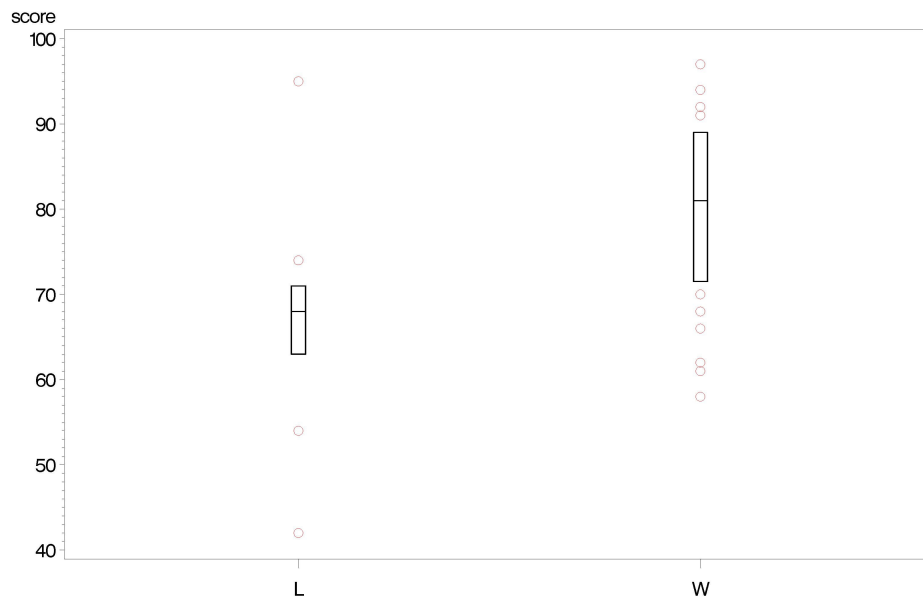
symbol1 interpol=BOXT25 color=black width=3 bwidth=2 value=circle
        cv=red height=1;

title h=1.7 'Total Breakdown of Points vs Outcome';

axis1 label=(h=1.2 ' ') minor=none offset=(25 pct);

proc gplot data=ncsu06;
    plot score*outcome/ haxis=axis1;
run;quit;
```

Total Breakdown of Points vs Outcome



You can change the color of the lines outlining the boxplots, as well as the color which fills them. Option F colors the boxplot with the color specified in 'CV='. The outline of the boxplot is modified by changing the value to 'CO='. I used the 'T' option in all examples presented so far. This option tells SAS to draw tops and bottoms to the line connectors. You can also add a line which connects the medians between neighboring boxplots through the 'J' option. Below I am displaying a graph, along with the code which generates it, demonstrating several of the options just discussed:

Red boxplot with blue outlines ('F' option):

```
GOPTIONS      RESET = all GSFNAME = gsasfile GSFMODE = replace
              ROTATE=landscape FTEXT = swiss CBACK = white
              TARGETDEVICE = HPLJGL2 dev=jpeg  xmax=11in ymax=8.5in
              xpixels=3300 ypixels=2550;

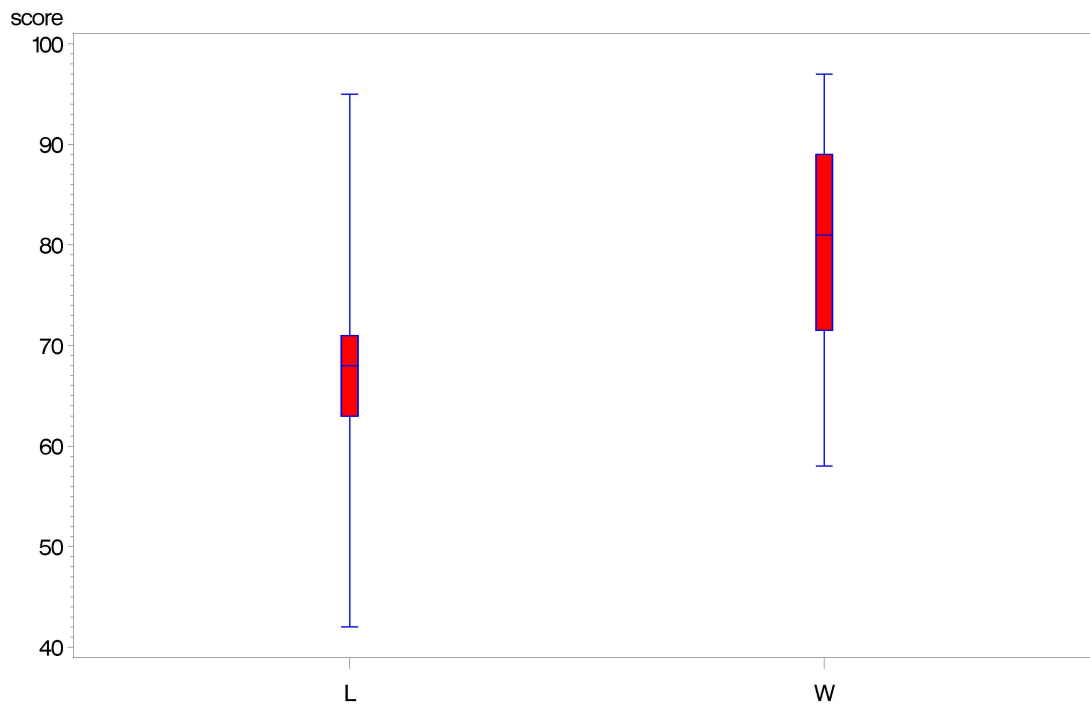
symbol11 interpol=BOXFT00 color=blue width=3  bwidth=2  value=circle
         cv=red height=1;

title h=1.7 'Total Breakdown of Points vs Outcome';

axis1 label=(h=1.2 '  ') minor=none offset=(25 pct);

proc gplot data=ncsu06;
    plot score*outcome/ haxis=axis1;
run;quit;
```

Total Breakdown of Points vs Outcome



In this next set of code I will make use of the 'J' option which will produce a connector line from the median of one boxplot to another:

J Option:

```
GOPTIONS      RESET = all GSFNAME = gsasfile GSFMODE = replace
              ROTATE=landscape FTEXT = swiss CBACK = white
              TARGETDEVICE = HPLJGL2 dev=jpeg  xmax=11in ymax=8.5in
              xpixels=3300 ypixels=2550;

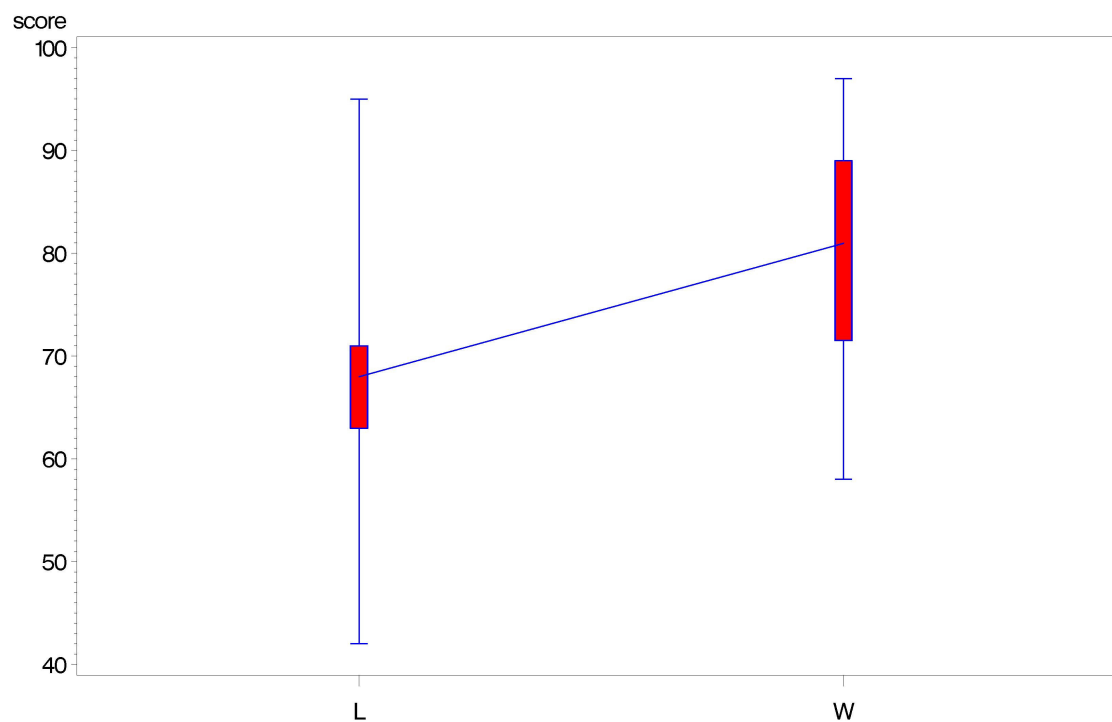
symbol1 interpol=BOXJFT00 color=blue width=3 bwidth=2 value=circle cv=red
height=1;

title h=1.7 'Total Breakdown of Points vs Outcome';

axis1 label=(h=1.2 ' ') minor=none offset=(25 pct);

proc gplot data=ncsu06;
  plot score*outcome/ haxis=axis1;
run;quit;
```

Total Breakdown of Points vs Outcome



PROFICIENCY TESTING BOXPLOTS

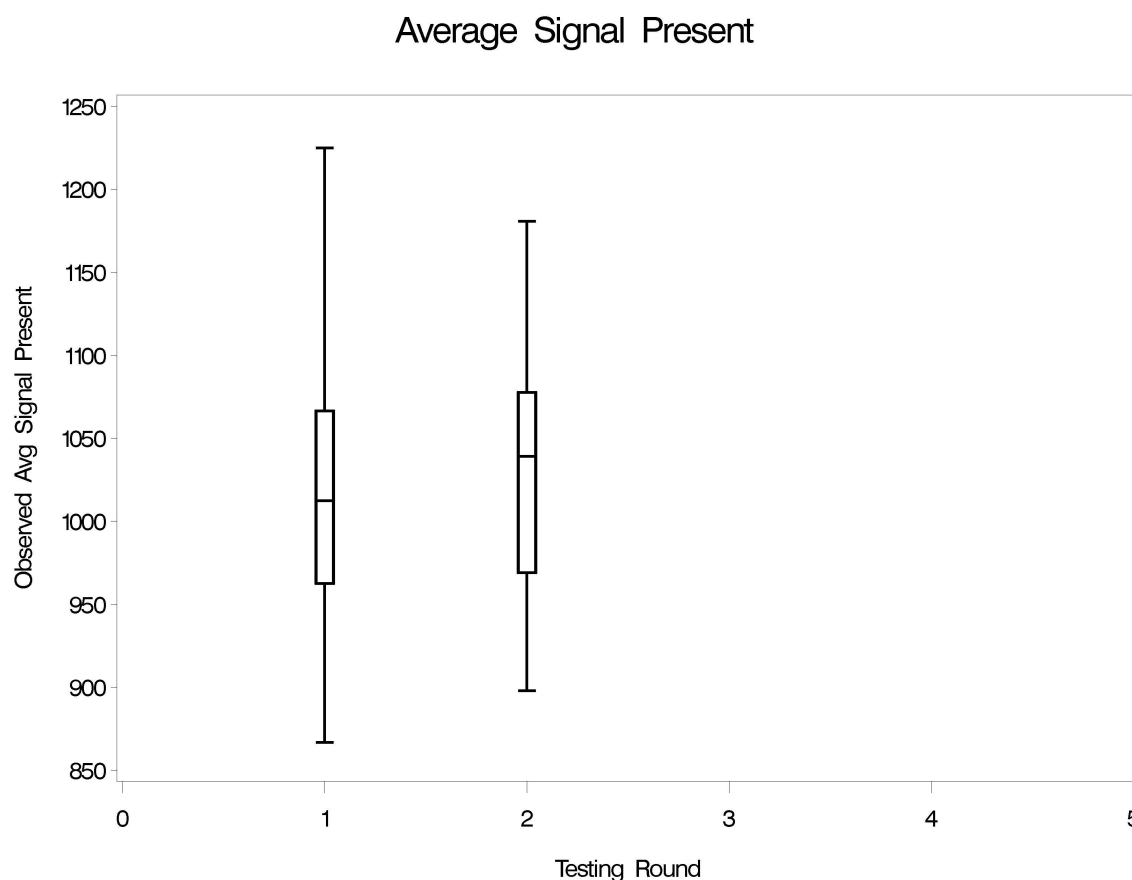
In 2004 I assisted several SAS programmers in developing proficiency testing code for laboratories involved in gene expression analysis. The proficiency testing report had a page which displayed lab variation with boxplots. I was put in charge of producing the sas code which generated these boxplots.

The proficiency testing involved four rounds. The example I will use throughout the remainder of the paper contains data from the first two rounds for a particular lab. The first round of testing took place in April 2004 and contained data for thirteen laboratories. The second round of testing was undertaken in September 2004 and had data for sixteen labs. Several laboratories were added between the first two rounds. The lab we are going to look at has data for both round one and round two.

The boxplots I generated for the proficiency testing report displayed data for a specific lab against other labs within a round as well as the distribution of data among labs between rounds. This involved generating boxplots which would be displayed side by side on the same axis.

This first example simply shows all labs' average signal present values within a round:

All Labs' Average Signal Present within a Round:



I used the following code to generate the above plot:

```
GOPTIONS      RESET = all GSFNAME = gsasfile GSFMODE = replace
              ROTATE=landscape FTEXT = swiss CBACK = white
              TARGETDEVICE = HPLJGL2 dev=jpeg  xmax=11in ymax=8.5in
              xpixels=3300 ypixels=2550;

symbol11 interpol=BOXT00  c=blue l=2 w=2.5;

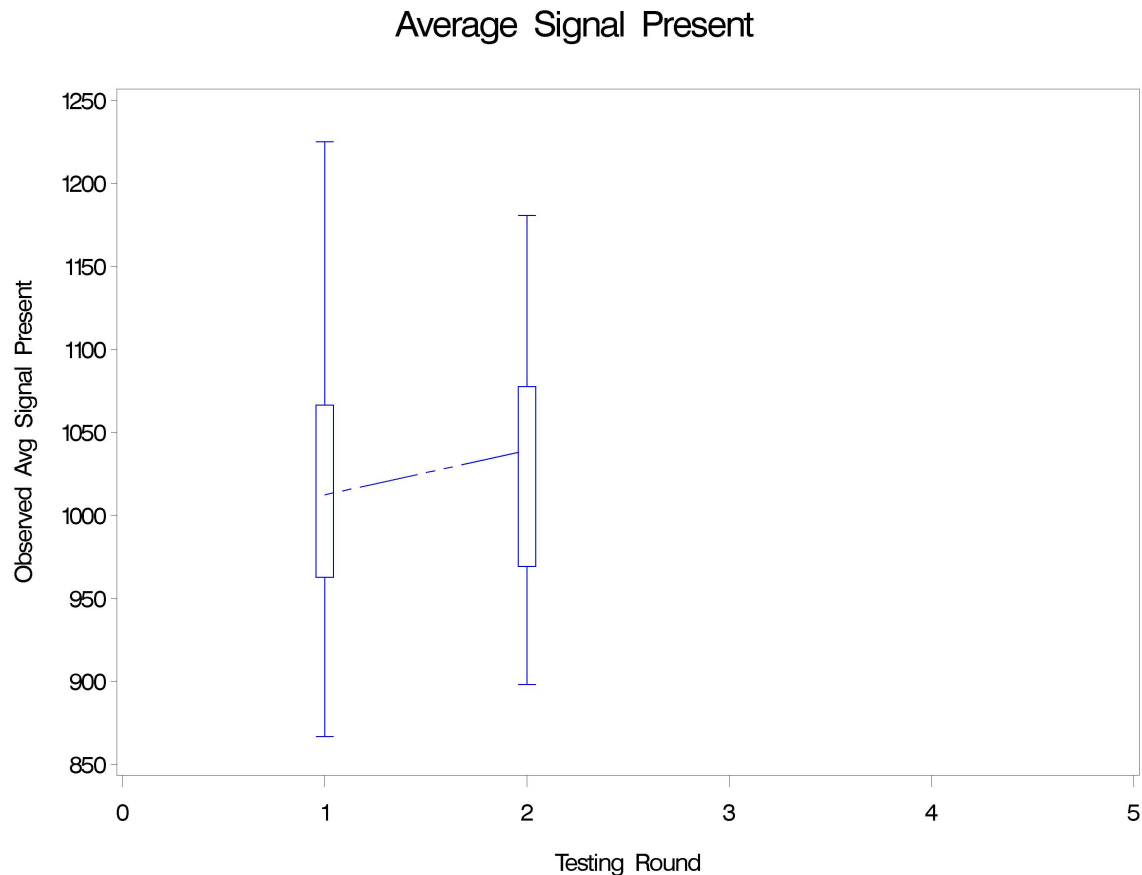
title h=1.5 'Average Signal Present';

axis1  order=(0 to 5 by 1) minor=none label=(a=360 h=2.5pct
                                              font='SWISS' "Testing Round") ;
axis2  order=(850 to 1250 by 50) minor=none label=(a=90 h=2.5pct
                                              font='SWISS' "Observed Avg Signal Present");

proc gplot data=inputds ;
    plot value*time = group  /haxis=axis1  vaxis=axis2
                                nolegend annotate=anno_all;
run;
quit;
```

By using interpol option BOXT00 I forced the boxplot whiskers to stretch from the minimum average signal value to the maximum signal value. The 'T' option causes the tops to be drawn on the whiskers.

I then added a connector which connected the medians of the two boxplots:



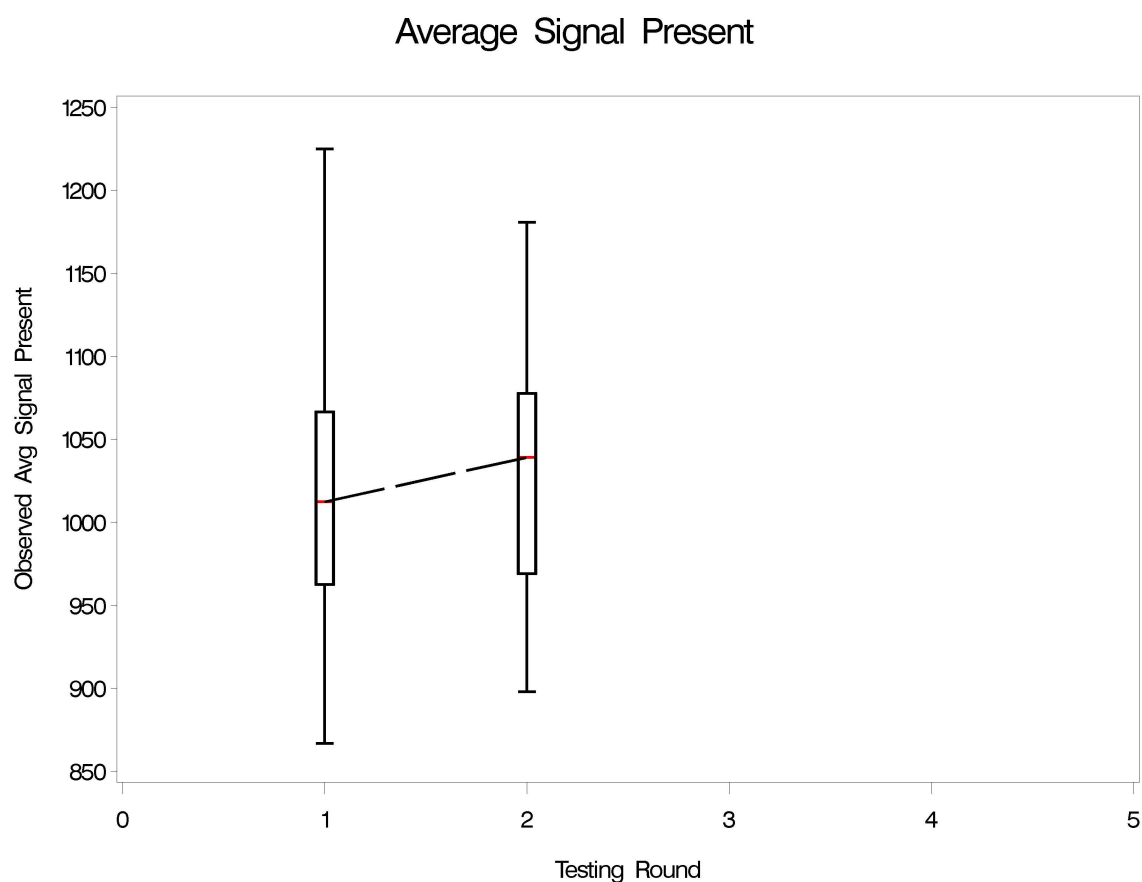
The following symbol statement produced the above graph:

```
Symbol1 interpol=BOXJT00 c=blue l=2 w=2.5;
```

I can change the color of the connector by changing the value to the 'c' parameter. I can also adjust line type by changing the value given to the 'l' parameter. 'W' changes the width of the connector line.

MEDIAN LINE WITH MEDIAN CONNECTORS

You will notice that adding a connector line will suppress the printing of the median line in both boxplots. I did not want the line to be suppressed. I even wanted to add the line in a different color (red) to emphasize the location of each round's median. I accomplished this goal by drawing two boxplots, one on top of the other. The first boxplot would be produced without the connector line. It would be in the color I wanted the median line to appear. On top of this first boxplot I drew a second one with a connector line. The desired result is shown below:



The above graph was generated by adding two symbol statements to the gplot procedure:

- 1) ***** BOXPLOT IN RED TO DRAW MEDIAN ***;**
`symbol1 interpol=BOXT00 color=red width=7 bwidth=2;`
- 2) ***** BOXPLOT IN BLACK / GRAY TO CONNECT ALL MEDIANS ***;**
`symbol2 interpol=BOXJT00 color=black /*grayaa*/ line=2 width=7
bwidth=2 ;`

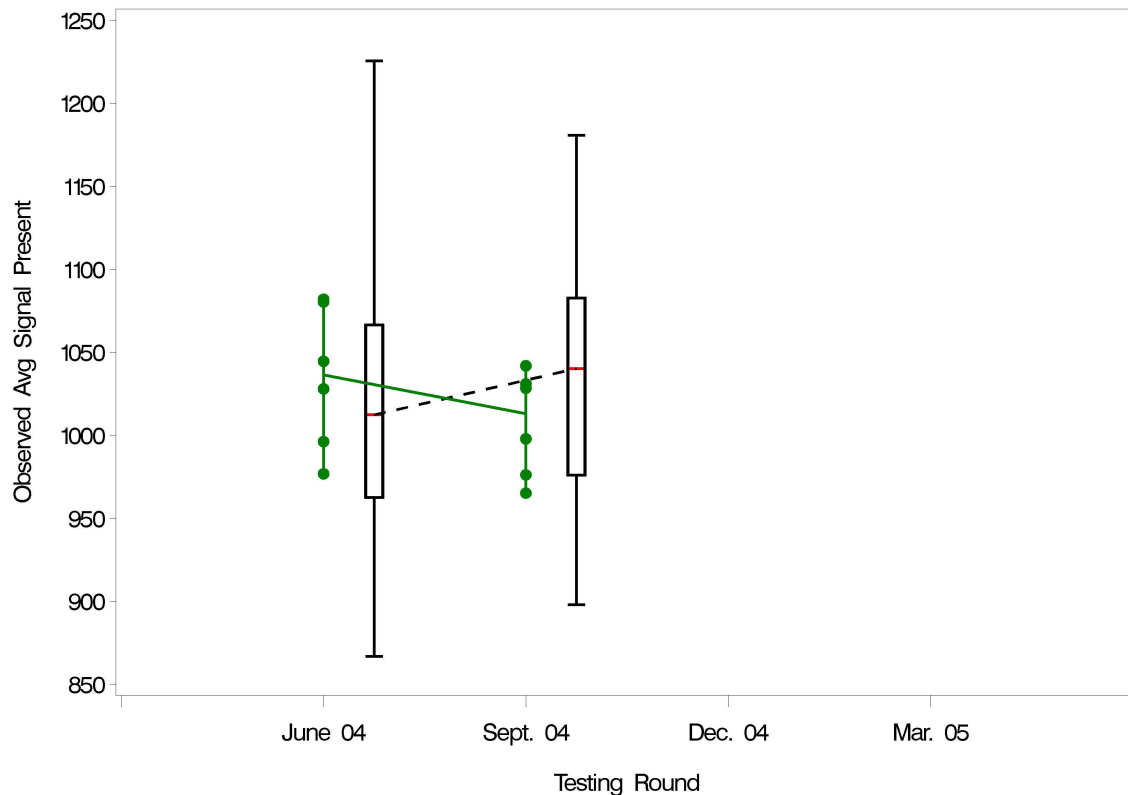
The first symbol statement draws the boxplot in red with the median line. The second symbol statement adds the 'J' to draw the connector between the two median lines.

INDIVIDUAL LAB NEXT TO ENTIRE ROUND

Next I wanted to display individual lab data next to each boxplot. This was accomplished through the use of the following symbol statement:

```
symbol1 interpol=HILOCTJ value=DOT h=.5 c=black l=1;
```

Average Signal Present



The green plot displays data for an individual lab. I wrote several other symbol statements to generate a line connector between the two plots.

```
*** LINE IN GREEN TO CONNECT MEDIAN FOR INDIVIDUAL LAB ***;
```

```
symbol3 interpol=join value=none color=green line=1 width=7;
```

```
*** LINE IN GREEN TO CONNECT DOTS AT EACH TIME POINT FOR INDIVIDUAL LAB ***;
```

```
symbol4 interpol=join value=DOT height=1 color=green line=1 width=7 repeat=4;
```

The 'interpol=hilo' statement tells sas to generate a vertical line which connects y-axis values for each x-axis value. Adding 'C' causes sas to draw marks at the close value instead of the default mean value. Like with boxplot the 'T' will add a top and bottom to each line and the 'J' option causes a connector line to link the two mean values between the two rounds.

ANNOTATE

I then decided to add the maximum value above each boxplot in blue. I determined the maximum value for each boxplot dataset and placed that value in a macro variable. Using annotate I was able to display the value above each plot.

```

data max_anno;
  %label(1,1245,compress("&max_val1"),BLUE,90,0,1,'SWISS',5);
  %label(2,1200,compress("&max_val2"),BLUE,90,0,1,'SWISS',5);
run;

```

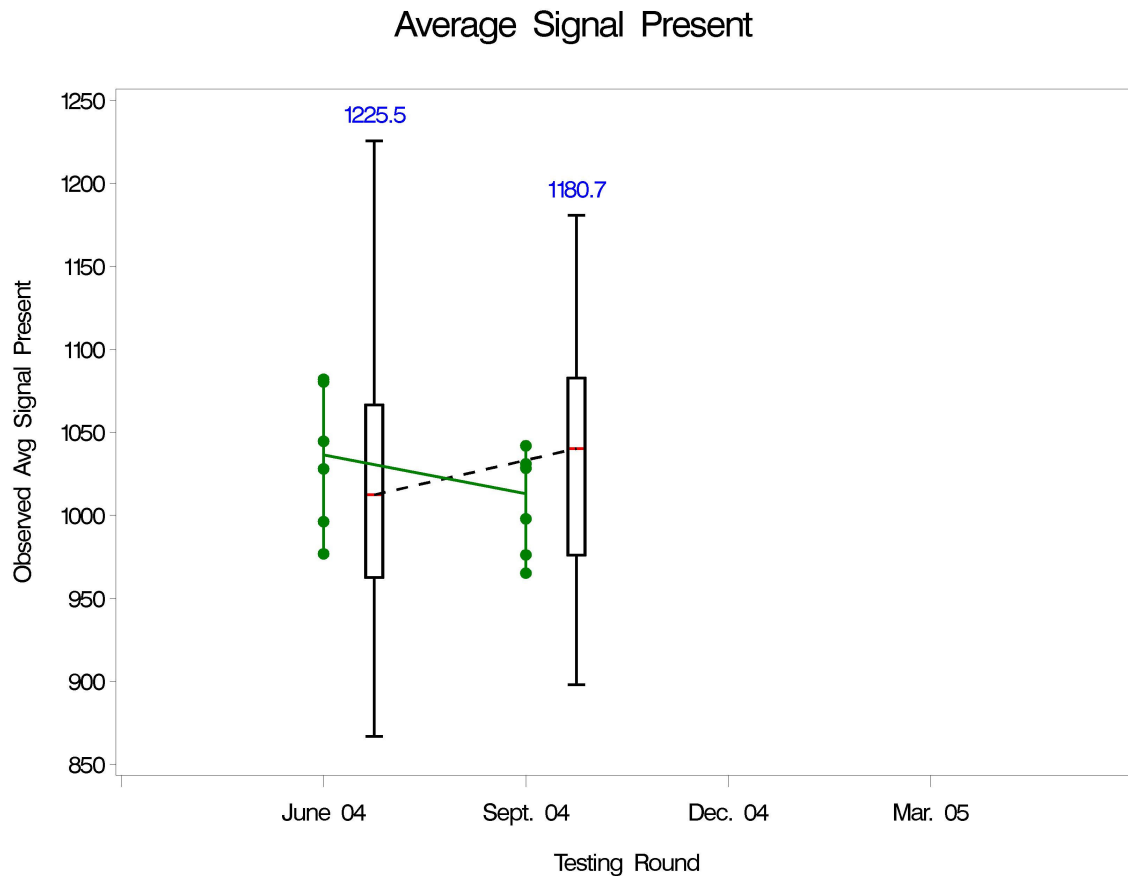
That annotate dataset was then appended to the proc gplot statement in order for SAS to use it:

```

proc gplot data=inputds
  format shift_time timefm. value ;
  plot value*shift_time = group /haxis=axis1 vaxis=axis2
                                nolegend annotate=max_anno;
run;
quit;

```

The above code with the annotate dataset generates the following plot:

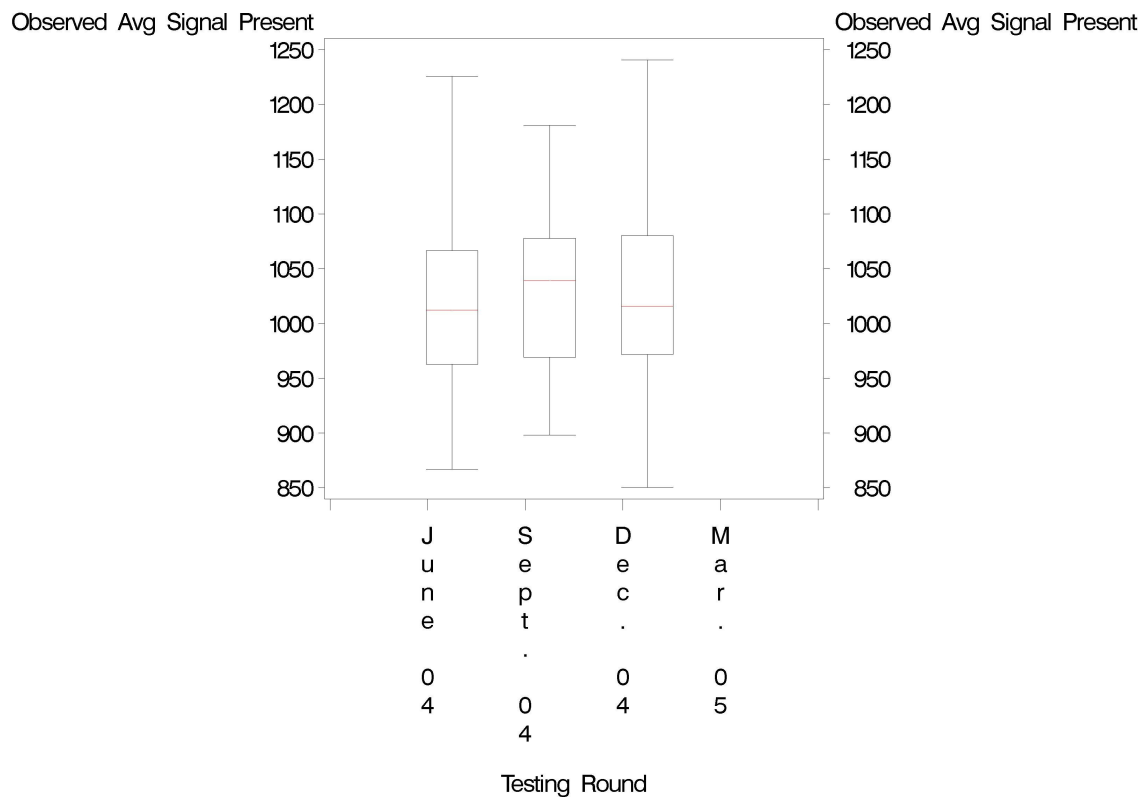


HORIZONTAL BOXPLOT

Next I will demonstrate how to generate horizontal boxplots. The client wanted to see the above data presented in boxplots which were horizontal in orientation. There was no option I could use to force SAS to produce the boxplot horizontally. I contacted SAS technical support and they suggested I make use of plot2 associated with SAS gplot. Using the plot2 statement I was able to generate axis labels for the other side of the graphical window. I then manipulated the orientation of my text so I could flip the graph using greplay.

Before any manipulation on the axis values I wanted to show you the output I received when I used a plot2 statement within SAS:

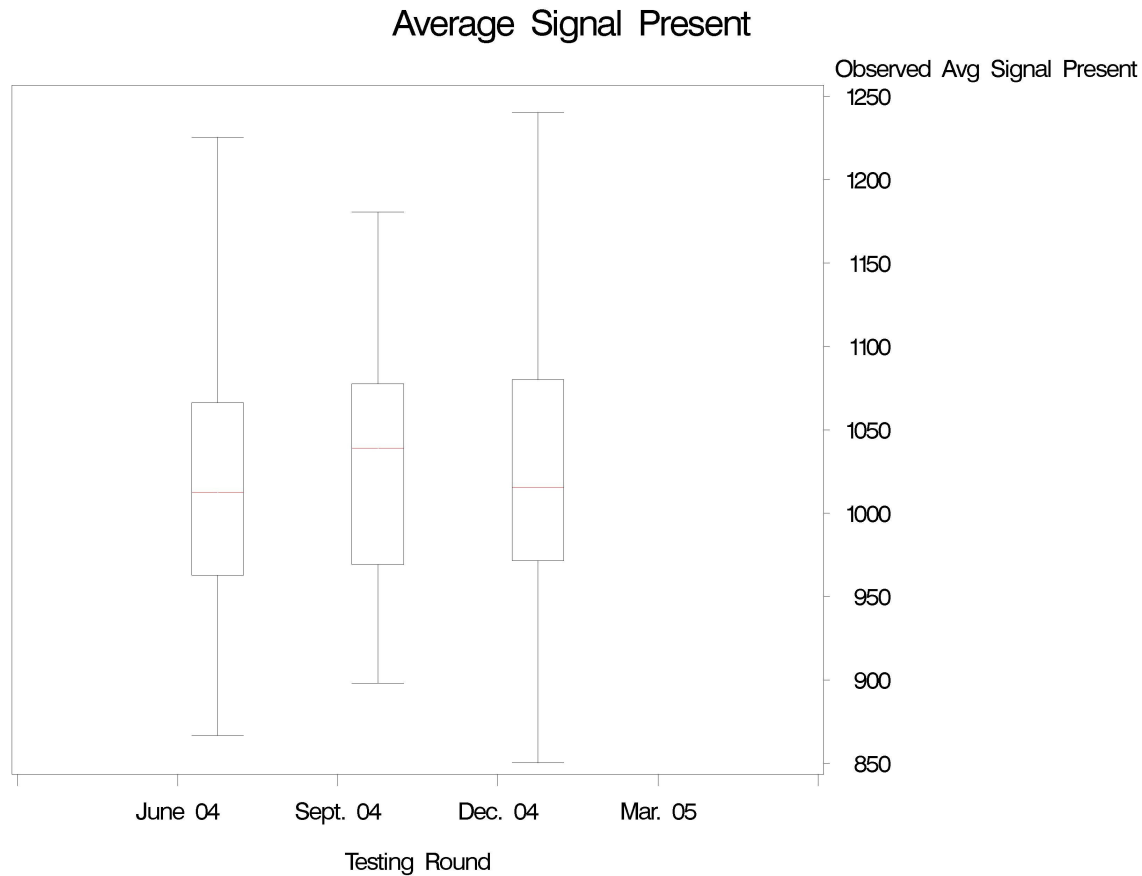
Average Signal Present



Notice the extra axes on the right of the graph. When I rotate my graph with `proc greplay` I will want the values displayed on the right to become my x-axis. I no longer need the y-axis labels and values associated with my first plot statement. I will modify my axis statement to suppress printing of values, labels and tick marks for my y-axis with the following axis statement:

```
axis2 order=(850 to 1250 by 50) minor=none major=none minor=none
      value=none label=none;
```

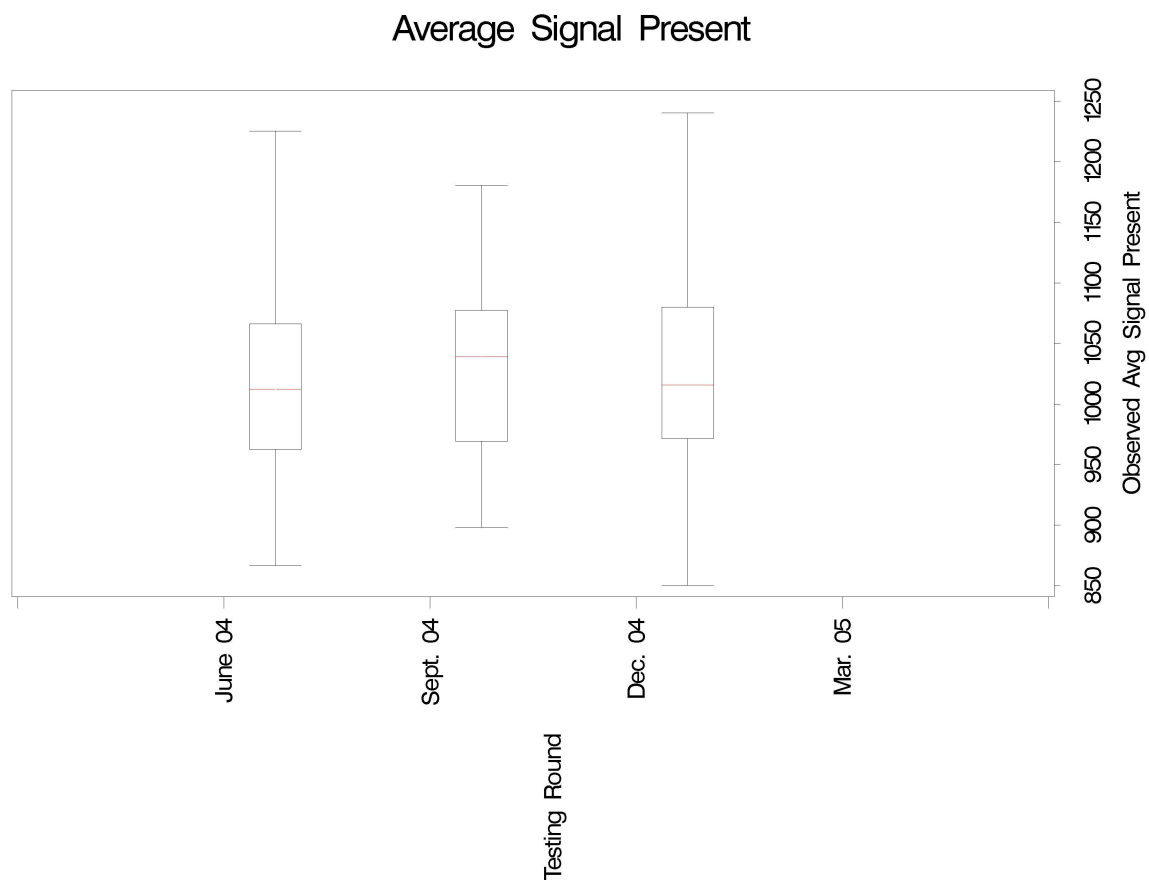
The resulting graph will look something like this:



I now need to modify the orientation of the labels and tick mark values for both remaining axes. Currently they look to be correctly orientated, but when I rotate my graphs they will be off. So I modify the two other axis statements to get the orientation I want:

```
axis1 value=(a=90) order=(0 to 5 by 1) minor=None label=(a=90 h=2.5pct
font='SWISS' "Testing Round") ;

axis3 value=(a=90) order=(850 to 1250 by 50) minor=None label=(a=90
h=2.5pct font='SWISS' "Observed Avg Signal Present");
```

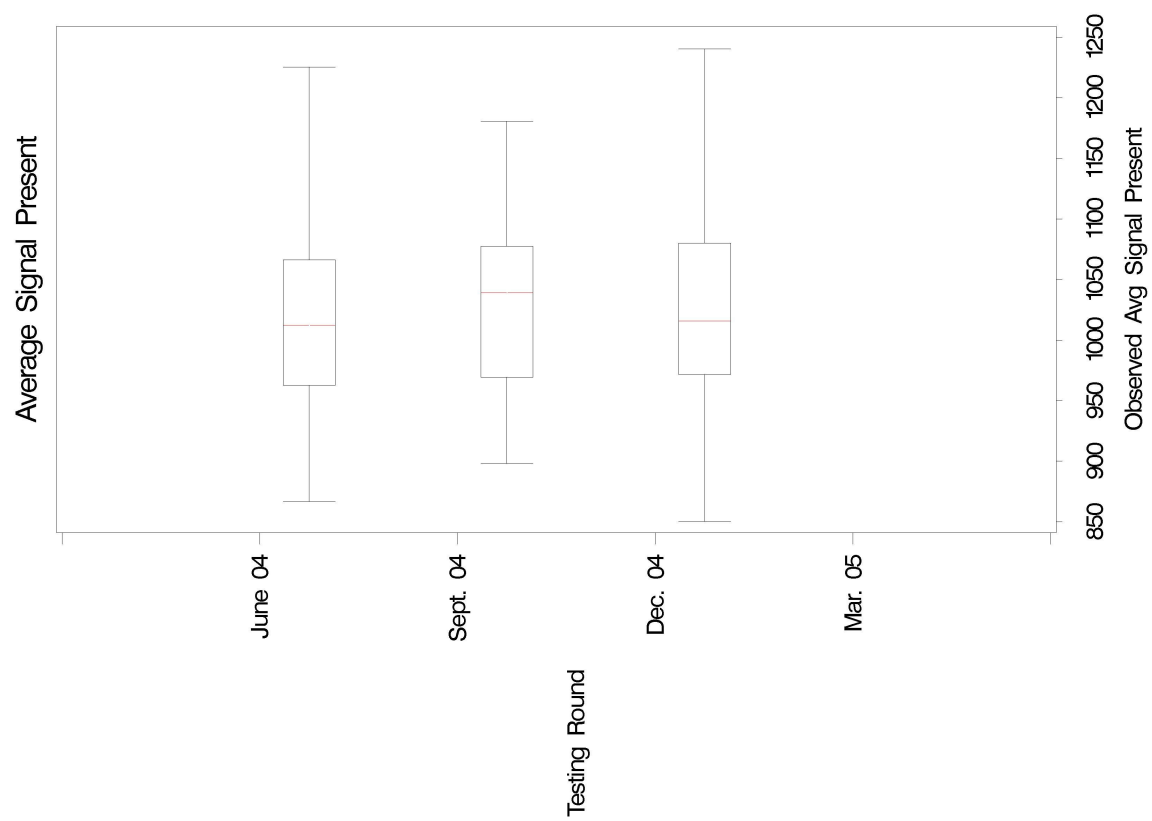


One last modification remains. Currently the title is in its proper location, however, once we rotate the graph it will go from being at the top of the graph to being off to its right. In order to correct this I will have to suppress the printing of the title and add it as a label for the current y-axis.

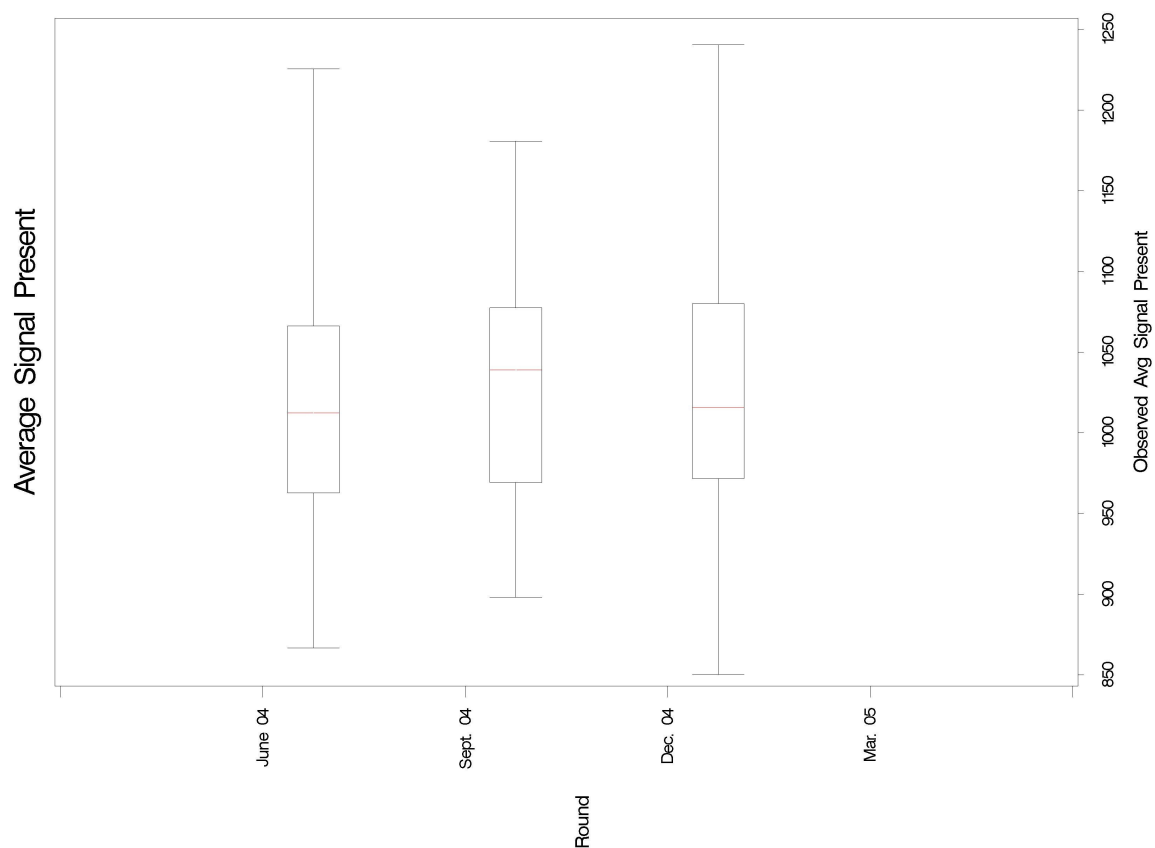
Instead of giving axis2's label a value of 'none' I will give it the following value:

```
axis2 order=(850 to 1250 by 50) minor=none major=none minor=none
value=none
label=(a=90 h=3pct font='SWISS' "Average Signal Present");
```

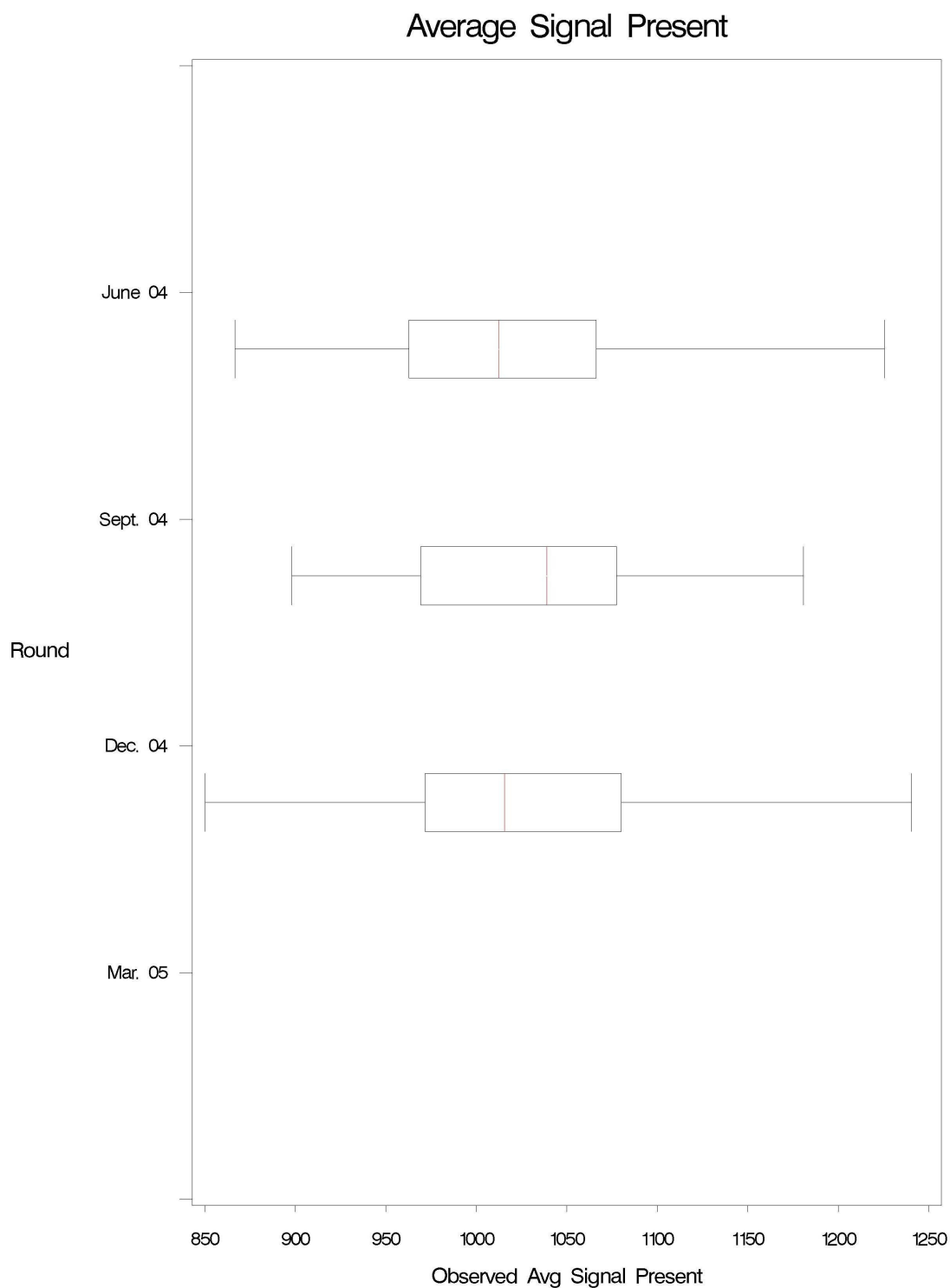
The resulting pre-rotated graph will look something like this:



I am almost at the point where I can rotate the graph to make the boxplots appear horizontally. I would like the axis on the right to be longer than it currently is. I can make minor modifications to the axis3 statement to lengthen the axis:



Rotating the above graph with `proc greplay` gives the following graphical output, which was our desired result:



CONCLUSION

Boxplots prove to be an excellent visualization of variation within data. SAS graph makes producing boxplots easy and simple. There exists much flexibility to ensure the data is presented in any desired format with few modifications. Hopefully the techniques presented within this paper will prove useful.

CONTACT INFORMATION

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

Brian Spruell
Constella Group, LLC
2605 Meridian Parkway, Suite 200
Durham, NC 27713
(919) 313 - 7673
bspruell@constellagroup.com
www.constellagroup.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.