

## Complementary Overlay – A Programmatic Approach to Figure Output Validation

Jesse Pratt, Rayce Wiggins, PPD, part of Thermo Fisher Scientific

### ABSTRACT

Effective and efficient validation of figures for a clinical trial can be a challenging process. Common techniques for validating outputs, such as PROC COMPARE and RTF parsing are not viable for figure validation. Without these tools, programmers must rely on double programming the input data set and manually inspecting the output against corresponding tables and listings. Many times, the input data sets are not “one PROC away”, and manual inspection of the output is prone to human error. This paper illustrates a novel programmatic method to figure validation by leveraging the combined usage of complementary colors and the DRAWIMAGE statement within the SAS Graph Template Language (GTL).

### INTRODUCTION

The process for validating figures via double programming can be summarized using the following steps:

1. Production side programmer writes specifications for the figure including:
  - o Dependent, independent, and grouping variables.
  - o Positioning parameters if annotation is used.
  - o Attributes such as colors, marker symbols, line patterns, font sizes, etc.
  - o Other image attributes such as size in pixels, DPI, etc.
  - o Programming syntax should be avoided.
2. Production side generates an image file (i.e. .PNG) of the figure according to specification.
3. Validation side also generates a version of the figure using complementary colors to the production side output.
4. Validation side uses annotation to bring in the figure created by production side.
5. Validation side adjusts the size and transparency of the production figure to ensure that it is the same size as the validation version of the figure.
6. Validation side overlays the transparent production side figure on top of the validation side figure.
7. Validation side inspects the overlaid figures for discrepancies that will appear in color (matching data will appear as gray).

With a general understanding of this process, it is important now articulate the core theory behind this method.

### SOME BRIEF COLOR THEORY

Central to the approach of validating overlaid figures is the concept and subsequent interaction of complementary colors. Since complementary colors are used to highlight differences in the outputs, it is important to understand the relevant theory behind them.

In the context of this approach, complementary colors are best understood as two colors that cancel each other's hues when mixed, ultimately producing a grayscale color. Additionally, it is important to understand that black can be thought of as the complete absence of hue, while white is the complete presence of all hues. With this in mind, let's look at how this theory is applied using SAS.

First, it is important to know that there are eight different color-naming schemes that are available in SAS. For the following demonstration we have chosen to use the RGB (red, green, blue) color-scheme. In this scheme, colors are represented by names in the form *CXrrggbb* where:

- *CX* indicates to SAS that this is an RGB color specification.
- *rr* denotes the RED component.
- *gg* denotes the GREEN component.
- *bb* denotes the BLUE component.
- The *rr*, *gg*, and *bb* components are represented by hexadecimal numbers that range from 00-FF (0%-100% of the respective hue).

In practice, for any given RGB color *CXrrggbb* a complementary color can be calculated by subtracting the respective *rr*, *gg*, and *bb* components from the hexadecimal number FF. In other words, we can think of this as “subtracting” the color from white:

$$\begin{array}{ccc} \text{CXFFFFFF} & - & \text{CXrrggbb} \\ \text{(white)} & & \text{(starting color)} \end{array}$$

Now, let’s look at a way to implement this in SAS. In order to illustrate how this concept works, let’s look at how we would derive the complementary color of RoyalBlue(CX4169E1):



Consider the following code:

```
data vhex;
  input COLOR $ WHITE $;
  datalines;
  CX4169E1 FFFFFFFF
  ;
run;

data vhex2;
  set vhex;
  VCOLOR="CX"||put(input(WHITE,hex6.)-
    input(strip(tranwrd(COLOR,"CX","")),hex6.),hex6.);
run;
```

This yields the complementary color CXBE961E:



## DETAILED EXAMPLE – SCATTERPLOT, 1 GROUP

Now we will illustrate the validation process with a simple scatterplot example. Consider some simulated data:

```
data main0;
  do OBS=1 to 50;
    XVAR=100*ranuni(922);
    YVAR=100*ranuni(9922);
    TRT=rand("table",0.5,0.5);
    output;
  end;
run;
```

```

data main;
  set main0;
  if OBS=27 then XVAR=.;
run;

data val;
  set main0;
  if OBS=11 then YVAR=YVAR-1;
  if OBS=42 then XVAR=.;
run;

```

In the MAIN and VAL datasets we have intentionally modified some observations to highlight the mismatches. With the data now in place, let's examine the production code responsible for plotting the figure. Note: GTL will be used throughout this paper, however similar methodology can be used with the SGPLOT and SGPANEL procedures. We have:

```

proc template;
  define statgraph mtemp;
    begingraph;
      layout overlay / xaxisopts=(label="X Variable")
                     yaxisopts=(label="Y Variable")
                     walldisplay=(fill);
      scatterplot x=XVAR y=YVAR / markerattrs=(symbol=circlefilled
        color=CX4169E1);
    endlayout;
  endgraph;
end;
run;

ods graphics / imagefmt=png imagename="gtlmain" noborder;
ods listing gpath="C:\TLF\Validation";

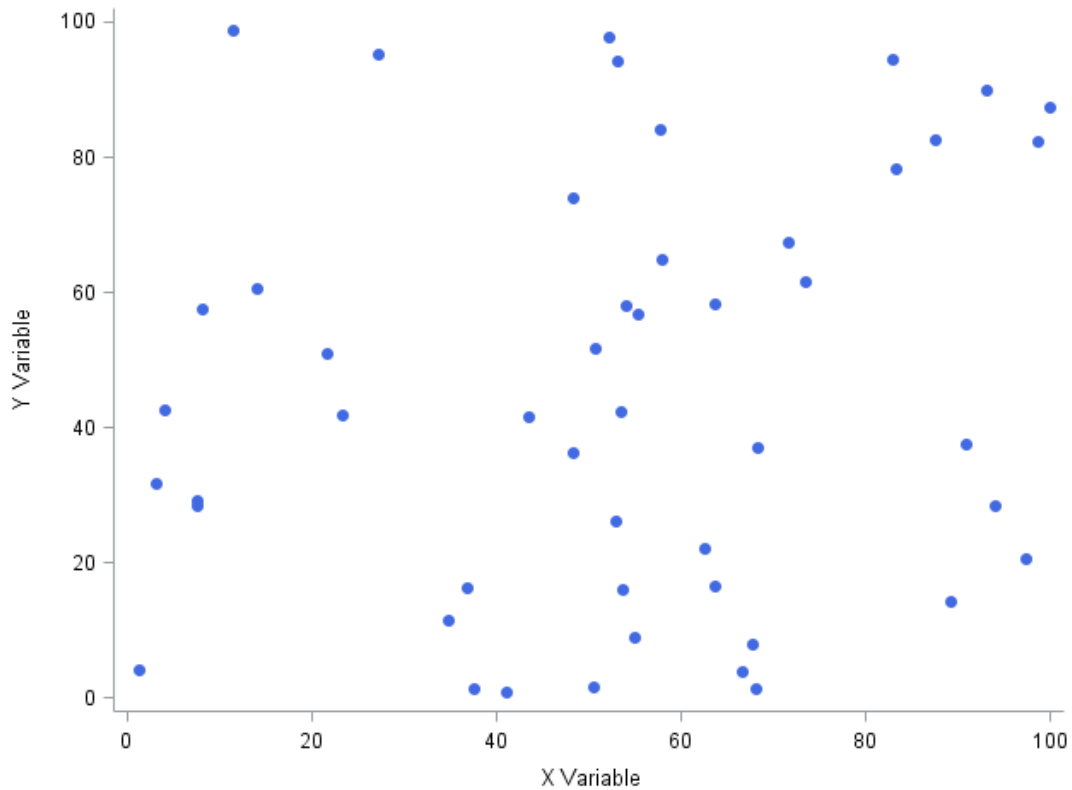
proc sgrender data=main template=mtemp;
run;

```

A few notes:

- The TEMPLATE procedure generates the template (MTEMP) for the production side figure while the SGRENDER procedure produces the actual output.
- The ODS GRAPHICS and ODS LISTING statements save a .PNG file of the output entitled GTLMAIN. It's then placed in the folder given in the GPATH= option.

This gives us the following image:



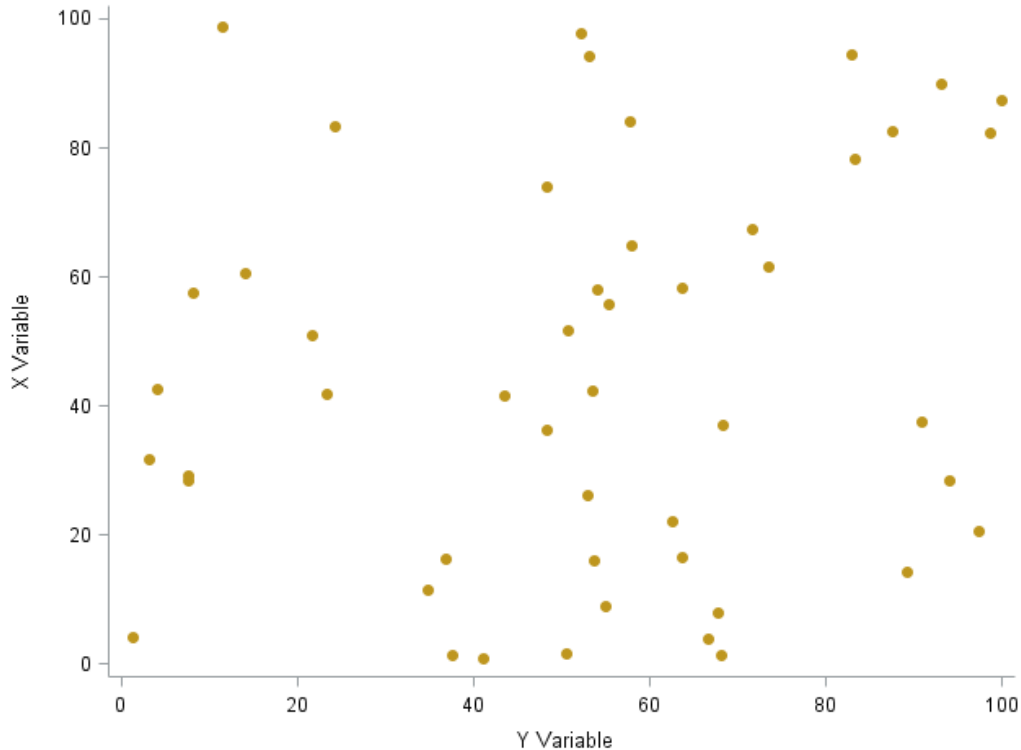
**Figure 1: Simple Scatterplot – Production Side**

Let's see what needs to be done on validation side:

```
proc template;
  define statgraph vtemp;
    begingraph;
      layout overlay / xaxisopts=(label="Y Variable")
                     yaxisopts=(label="X Variable")
                     walldisplay=(fill);
      scatterplot x=XVAR y=YVAR / markerattrs=(symbol=circlefilled
        color= CXBE961E);
    endlayout;
  endgraph;
end;
run;

proc sgrender data=val template=vtemp;
run;
```

This yields the following image on validation side:



**Figure 2: Simple Scatterplot – Validation Side**

Note the usage of the complementary color to RoyalBlue in the validation side output. As it stands now, it would be difficult to visually detect any mismatches with a side-by-side comparison. To make the differences more apparent let's make the following modifications to our validation side code highlighted below:

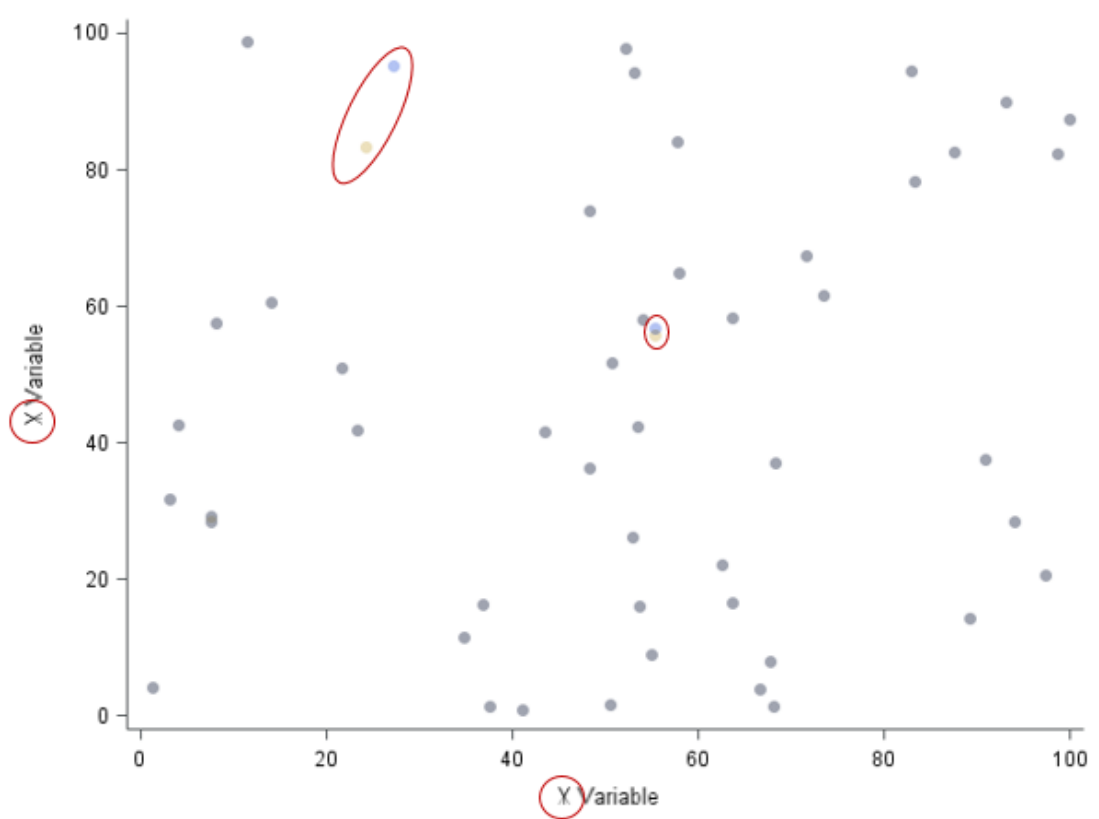
```
proc template;
  define statgraph vtemp;
    begingraph;
      layout overlay / xaxisopts=(label="Y Variable")
        yaxisopts=(label="X Variable")
        walldisplay=(fill);
      scatterplot x=XVAR y=YVAR / markerattrs=(symbol=circlefilled
        color=CXBE961E) datatransparency=0.3;
      drawimage "C:\TLF\Validation\gtlmain.png" / x=0 y=100
        drawspace=graphpercent anchor=topleft transparency=0.6;
    endlayout;
  endgraph;
end;
run;

proc sgrender data=val template=vtemp;
run;
```

Some notes:

- In addition to using the complementary colors, we also need to make both the validation side output, and the overlaid production .PNG file transparent.
- We set a value of 0.3 to the DATATRANSPARENCY= option for the validation side output. This value is not arbitrary because it is needed to obtain a consistent shade of gray.
- The DRAWIMAGE statement reads in the production side .PNG file and overlays it on top of the validation side output. The options are set precisely to accomplish this.

This gives us the following output (red circles added to highlight differences):

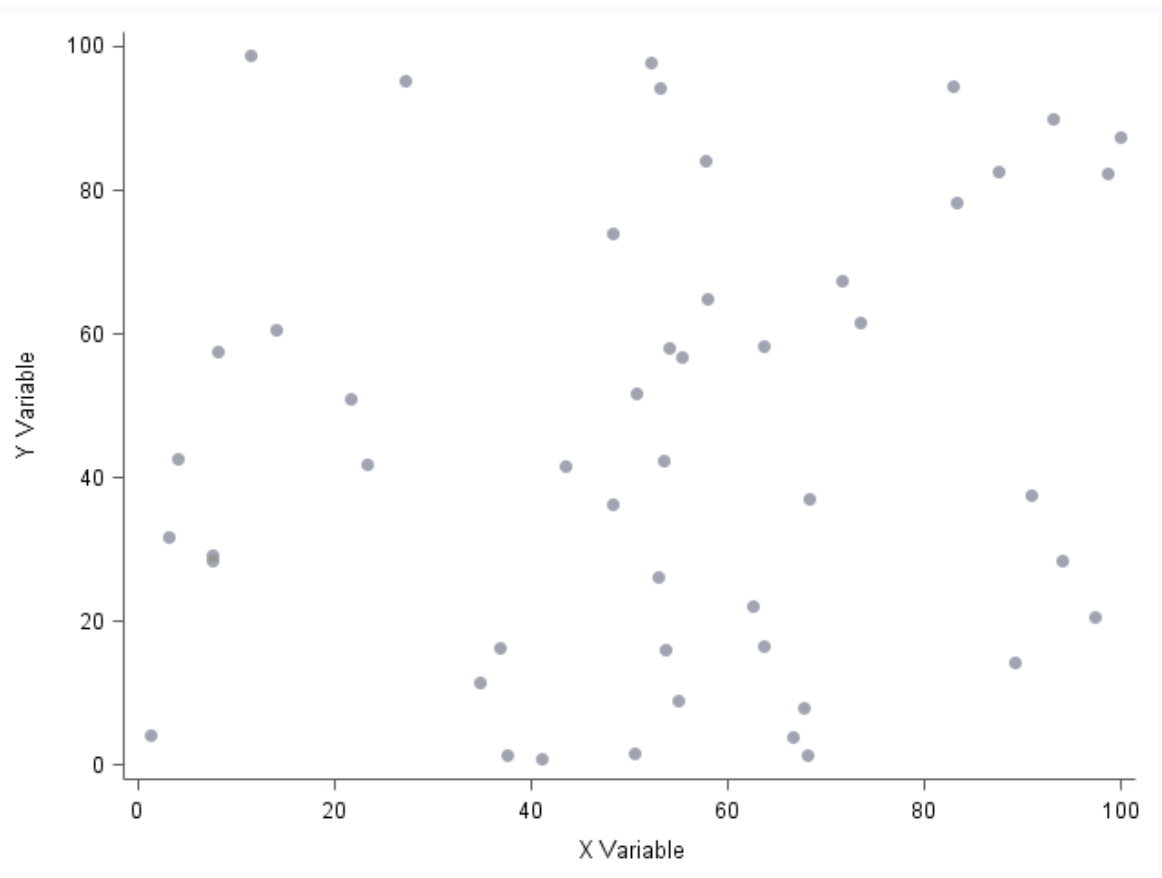


**Figure 3: Overlaid Validation Output with Mismatches**

Some notes:

- The matching portions of the data now present as gray.
- The mismatches are in color.
- The labels are mismatched with a “shadow” effect present in the text.

A fully validated version of this figure would look like:

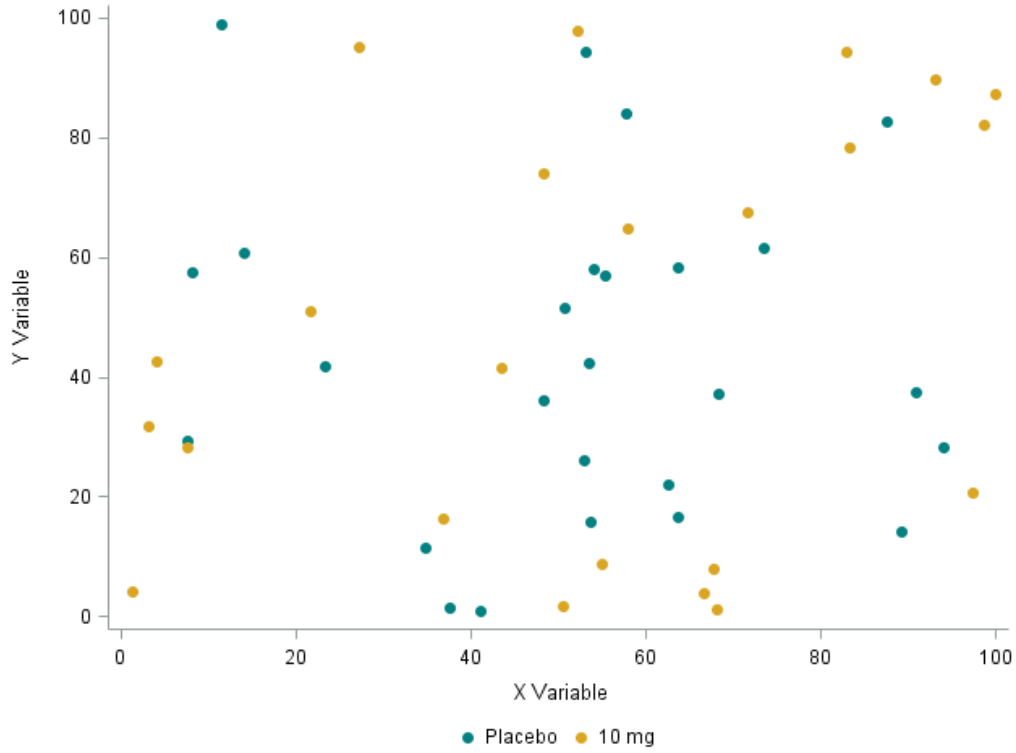


**Figure 4: Overlaid Validation Output with No Mismatches**

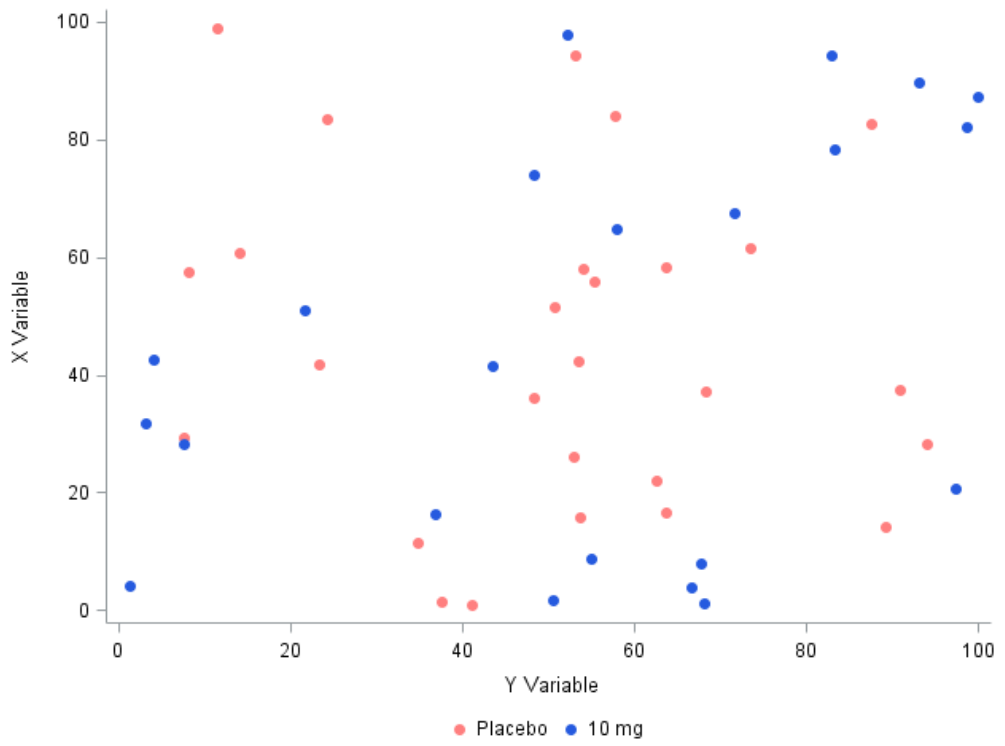
This method can be used for any type of figure, including those with grouped data having more than one color per group. To further demonstrate this technique, let's look at some other types of figures. Code will be provided in the appendix for the rest of these examples.

### EXAMPLE – SCATTERPLOT, 2 GROUPS

For this next example let's suppose that we now have two different treatment groups. How would this change our approach? In the GTL code, production side would assign colors to each treatment group whereas validation side would calculate the corresponding complementary colors. Plots would then be overlaid as in the previous example. Let's see how these outputs would look:

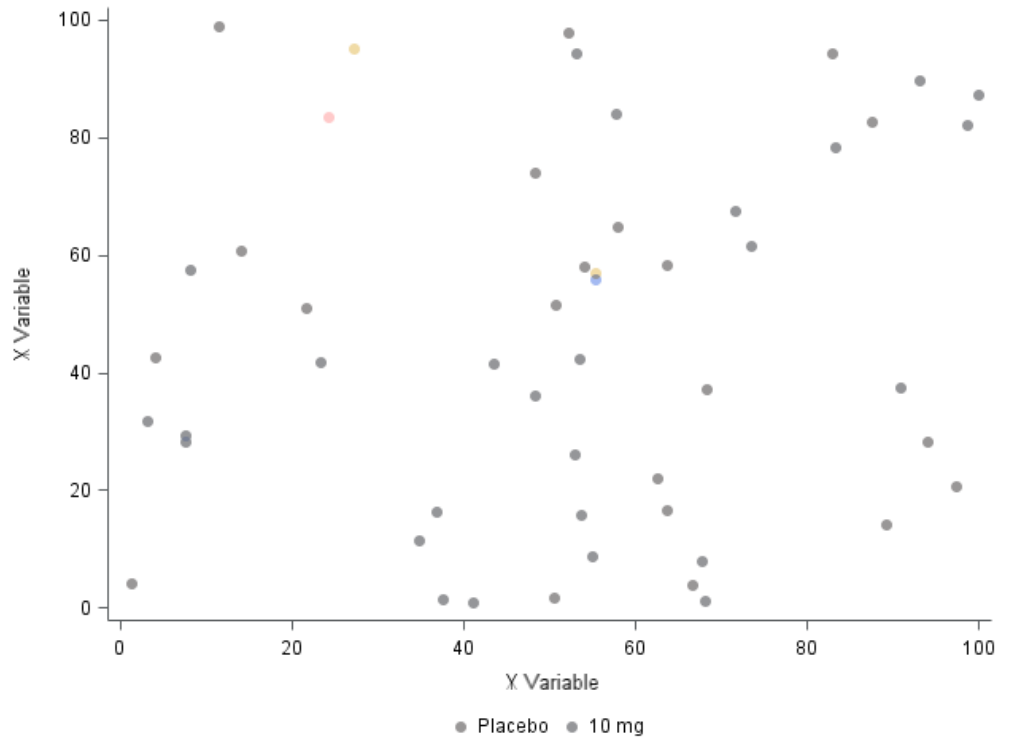


**Figure 5: Scatterplot with Two Treatment Groups – Production Side**

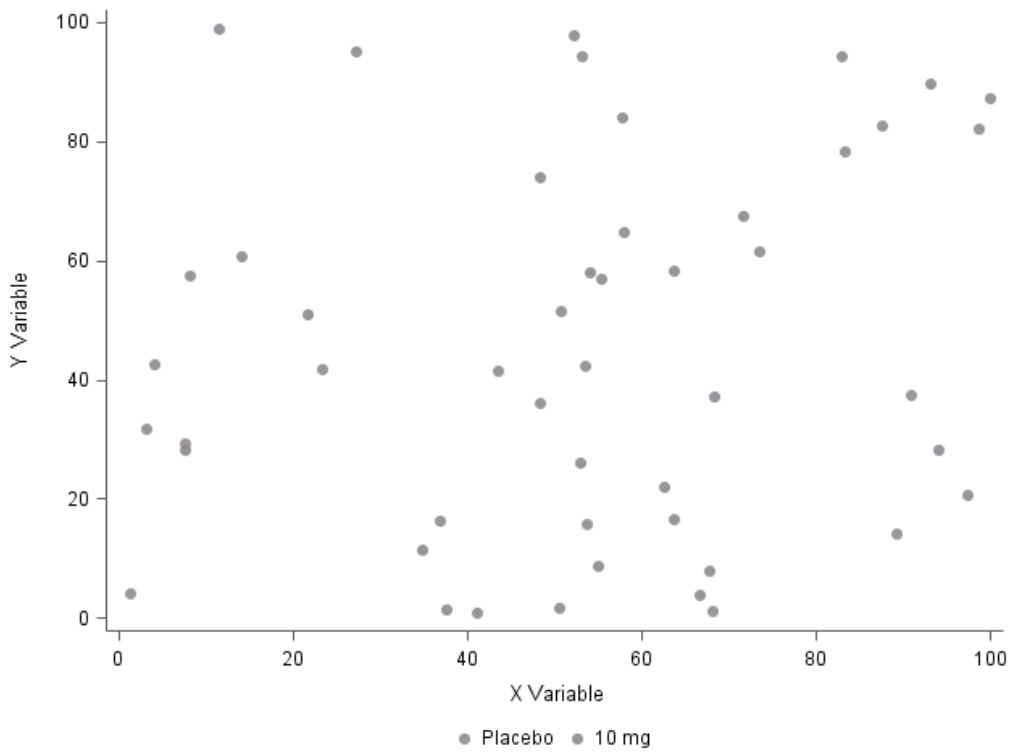


**Figure 6: Scatterplot with Two Treatment Groups – Validation Side**





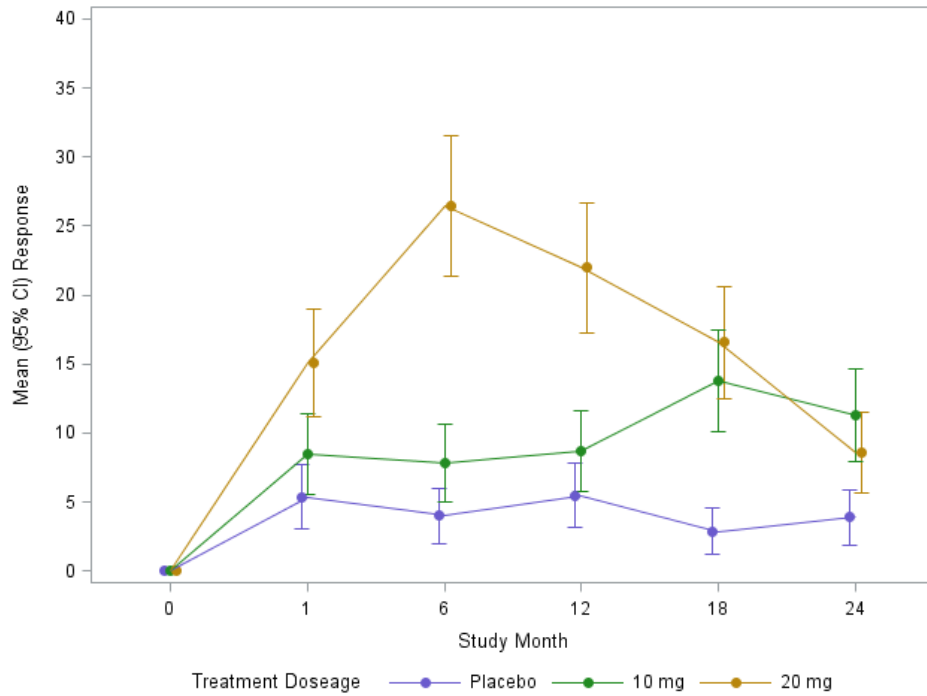
**Figure 7: Overlaid Validation Output with Mismatches**



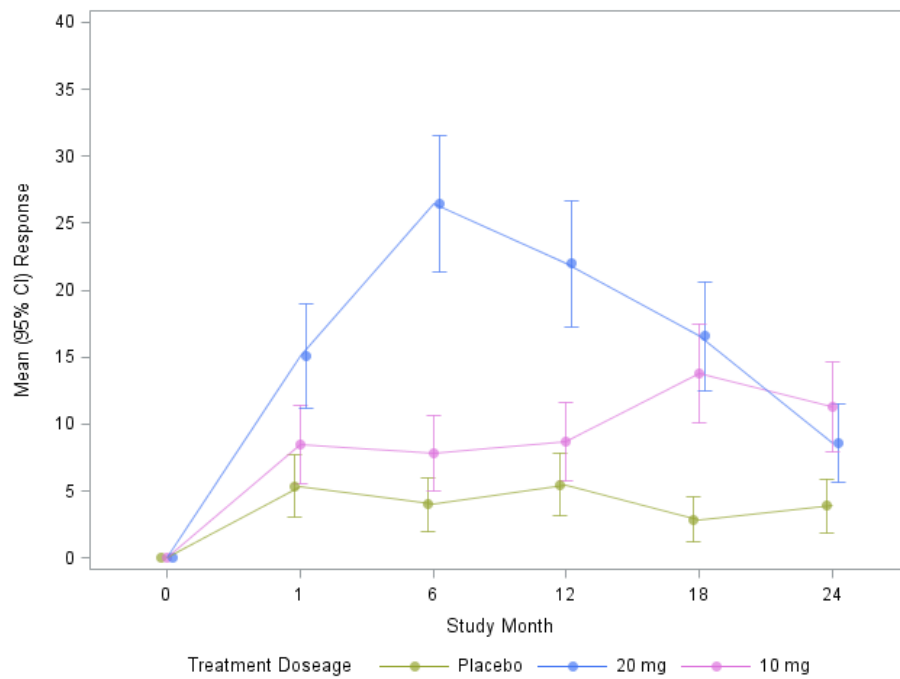
**Figure 8: Overlaid Validation Output with No Mismatches**

## EXAMPLE – LINE PLOT, 3 GROUPS

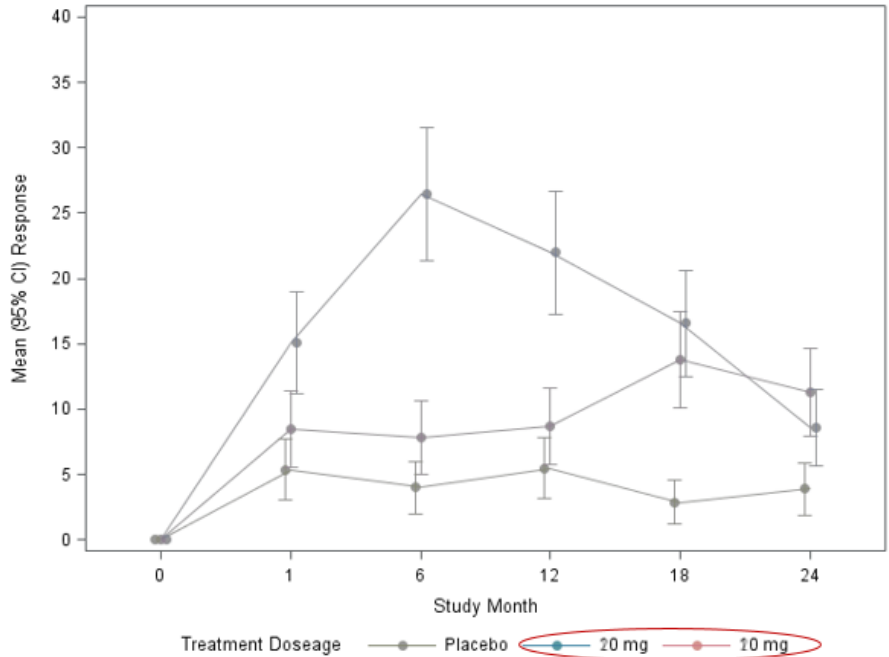
This technique can also be used to highlight discrepancies in legends, as the following example shows.



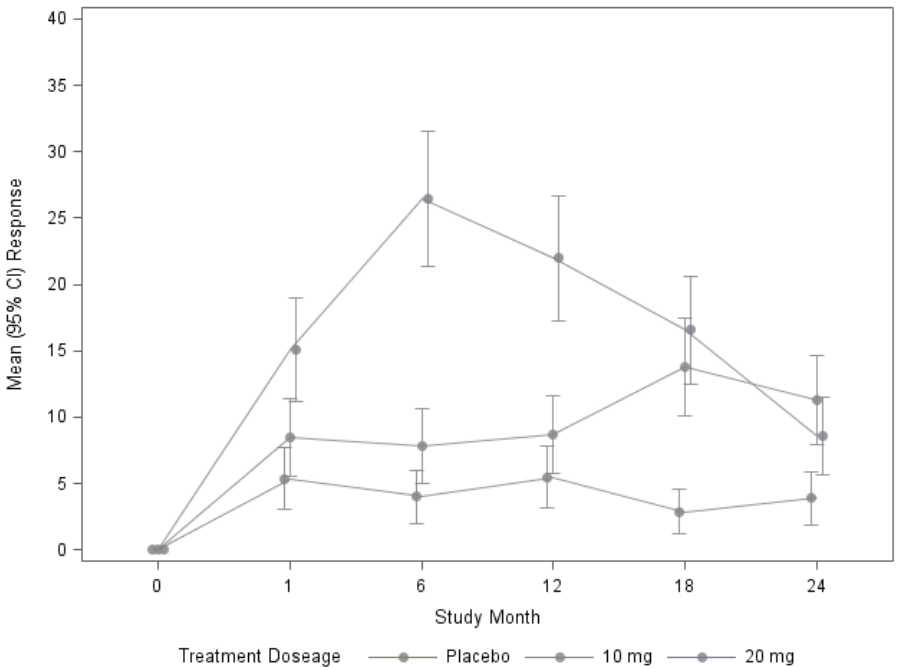
**Figure 9: Line Plot with Three Treatment Groups – Production Side**



**Figure 10: Line Plot with Three Treatment Groups – Validation Side**



**Figure 11: Overlaid Validation Output with Mismatched Legend**



**Figure 12: Overlaid Validation Output with No Mismatches**

## EXAMPLE – BOXPLOT, 2 GROUPS

Fill attributes, such as with boxplots, also lend themselves to this technique.

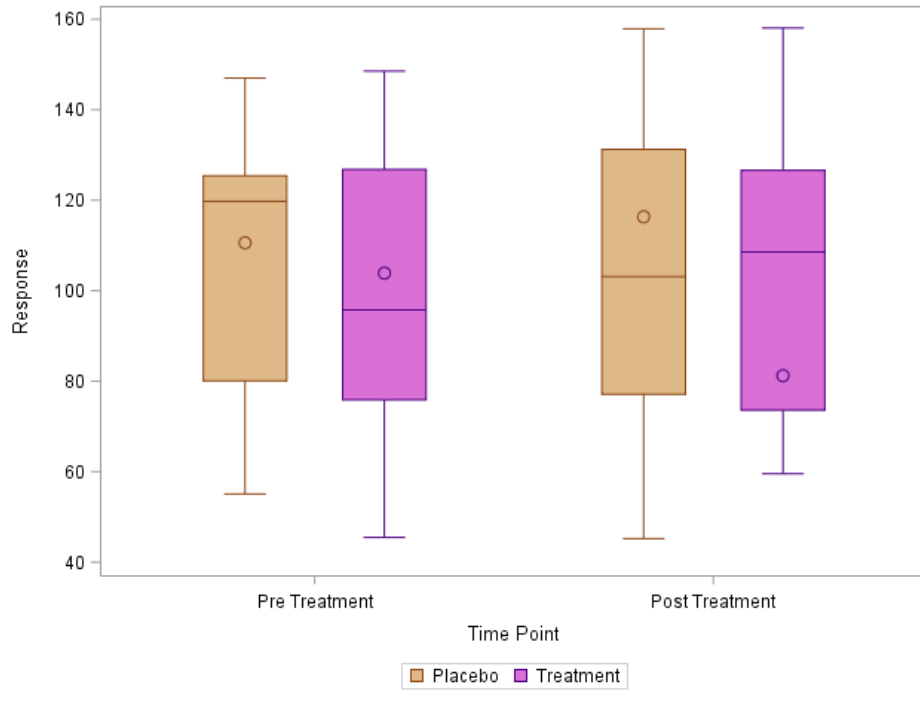


Figure 13: Boxplot with Two Treatment Groups – Production Side

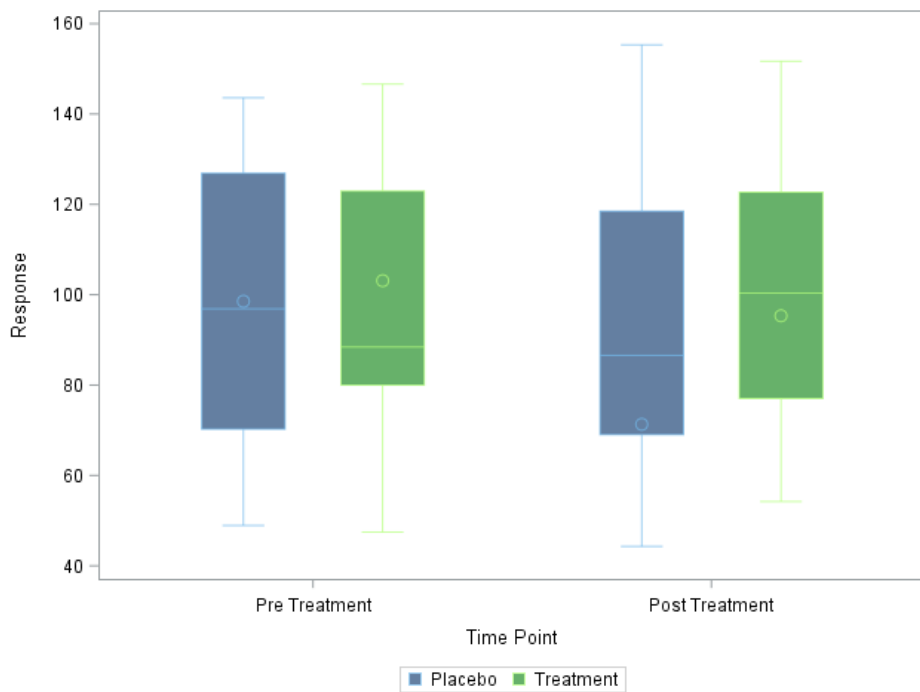
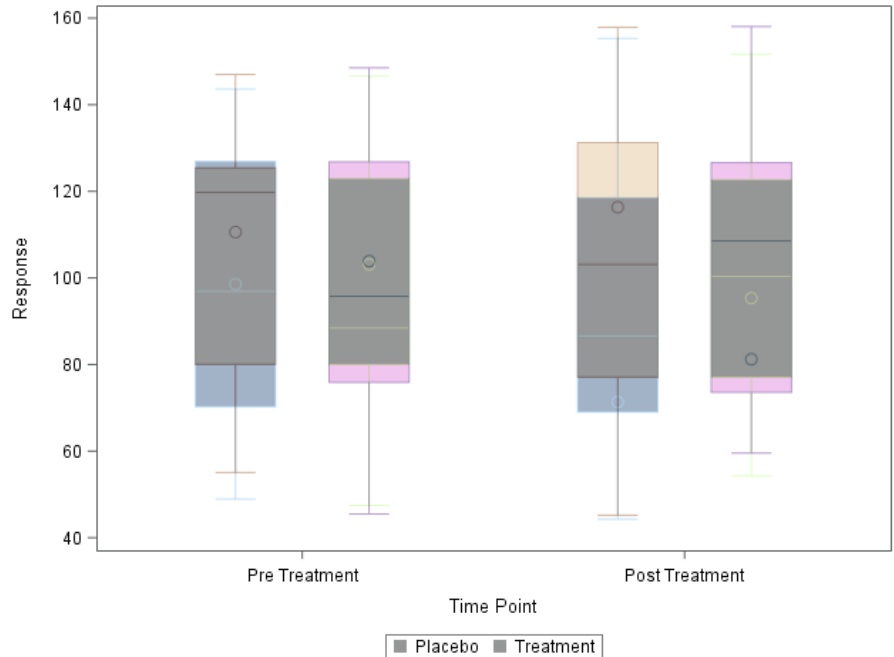
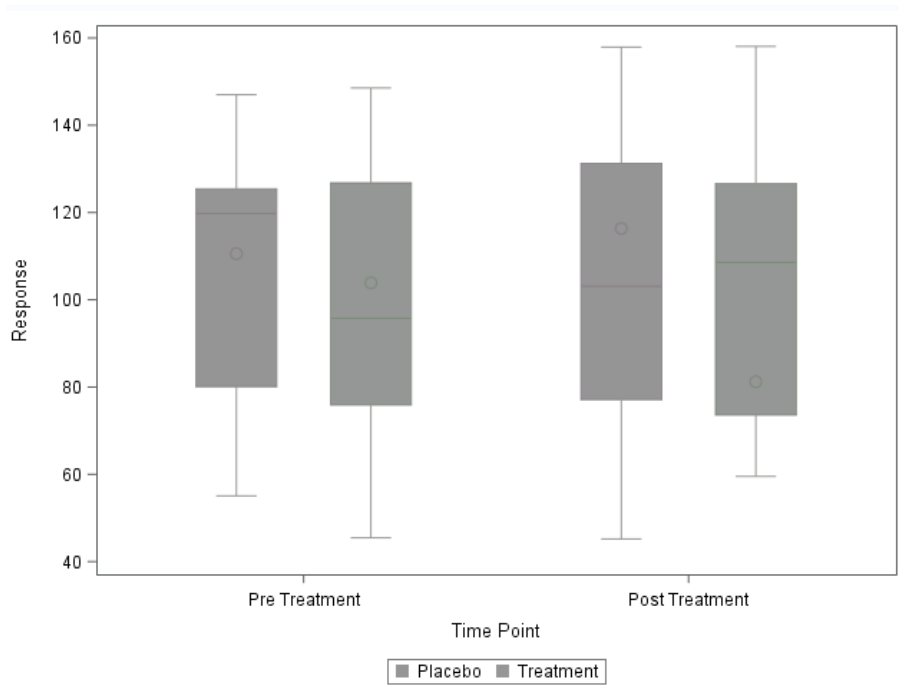


Figure 14: Boxplot with Two Treatment Groups – Validation Side



**Figure 15: Overlaid Validation Output with Mismatches**



**Figure 16: Overlaid Validation Output with No Mismatches**

## EXAMPLE – GRAYSCALE, 3 GROUPS

This technique can also work for grayscale figures. Simply choose any color with a hue that is not grayscale (e.g. red) for the validation side figure, and watch for the “shadowing” effect.

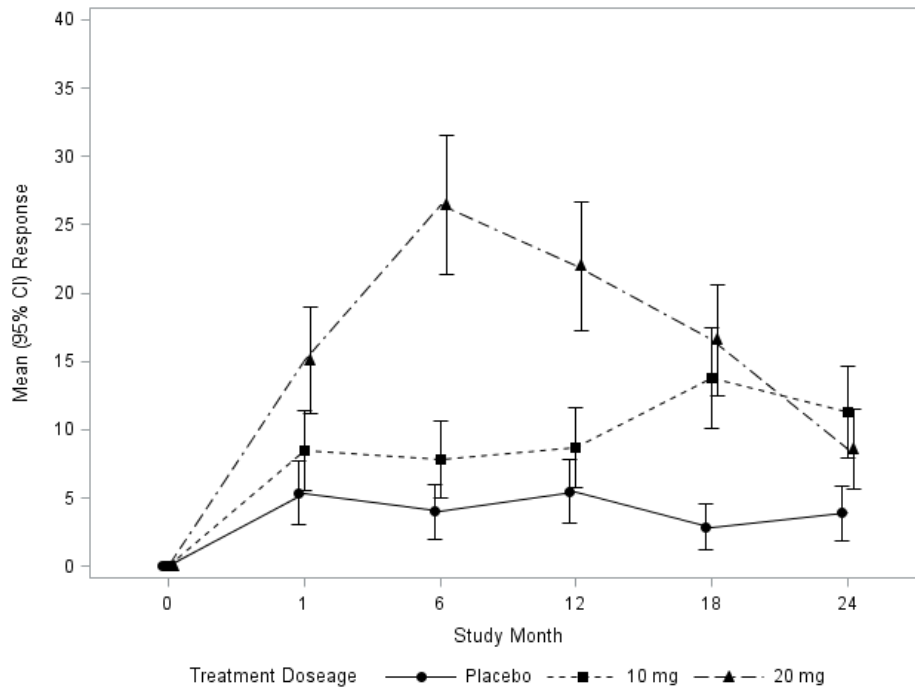


Figure 17: Grayscale Figure – Production Side

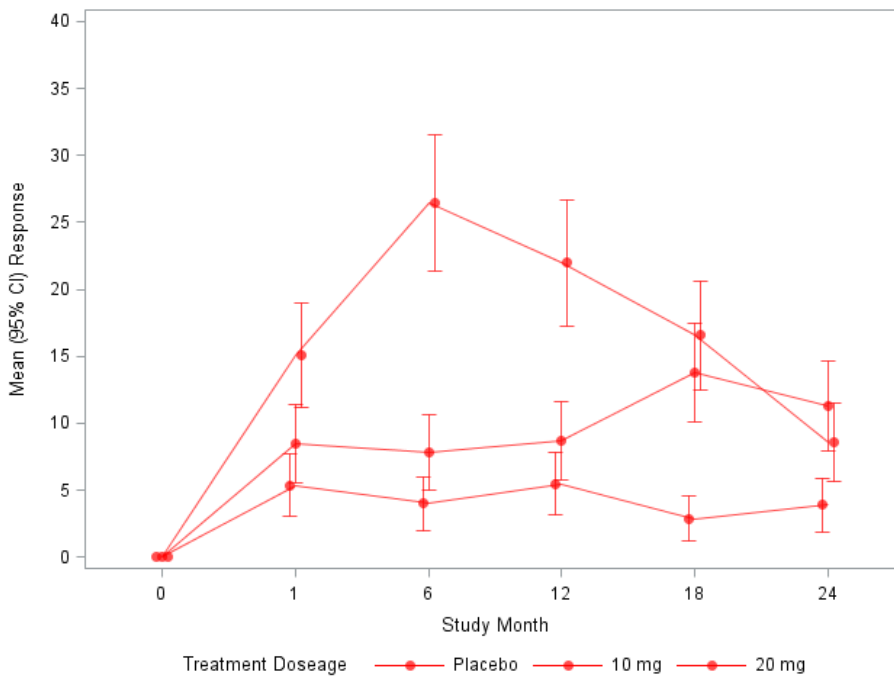
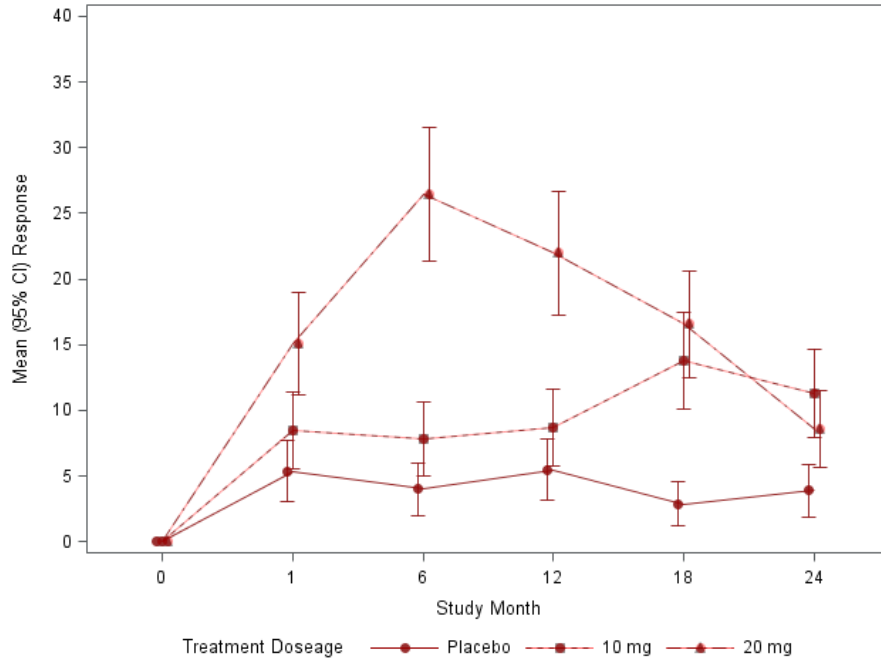
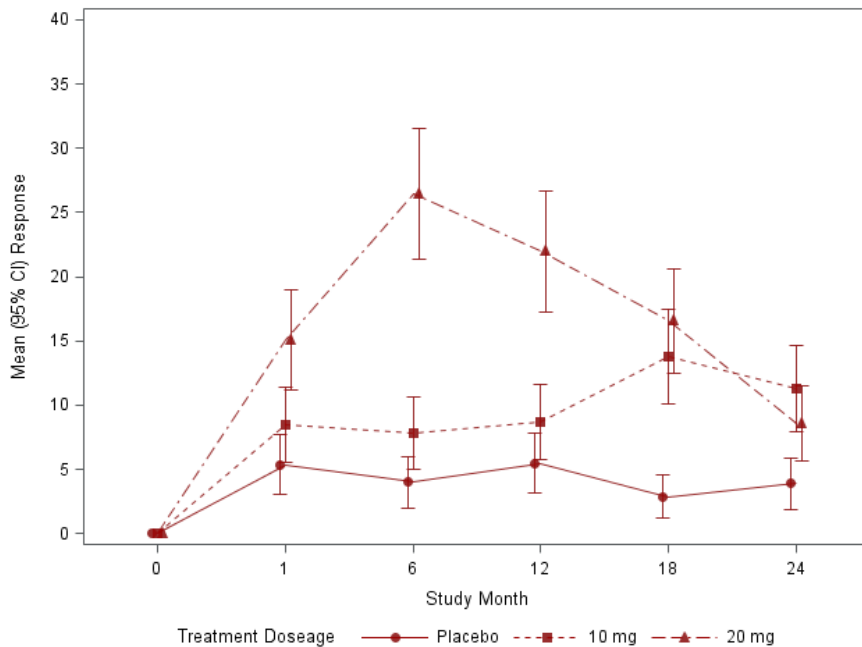


Figure 18: Grayscale Figure – Validation Side



**Figure 19: Overlaid Validation Output with Mismatches**

If we look where the light red shadows appear in the figure above, we can see that production and validation are differing on their choice of line patterns for the three treatments (dashed vs. solid respectively). We can also see that this mismatch is evident in the legend as well.



**Figure 20: Overlaid Validation Output with No Mismatches**

## CONCLUSION

Until now, a tool that is as powerful as PROC COMPARE for data sets or RTF parsing for tables has not emerged. By combining complementary color theory with overlaying techniques, however, we now have a third arrow in our quiver to more effectively and robustly validate figures. This method has the potential to increase the quality of the final product by greatly reducing the likelihood of human error from the figure validation process.

## CONTACT INFORMATION

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

Jesse Pratt  
jesse.pratt@ppd.com

Rayce Wiggins  
rayce.wiggins@ppd.com

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

## APPENDIX – CODE FROM EXAMPLES

```
*****
*****
/* Validate two groups - Scatterplot */
*****
*****;
%let C1=CX008080;
%let C2=CXDAA520;

proc template;
  define statgraph mtemp2;
    begingraph;
      legenditem type=marker name="one" / label="Placebo"
        markerattrs=(symbol=circlefilled color=&C1);
      legenditem type=marker name="two" / label="10 mg"
        markerattrs=(symbol=circlefilled color=&C2);
      DiscreteAttrVar attrvar=ATTR_TRT var=TRT attrmap="myattrmap";
      DiscreteAttrMap name="myattrmap";
        Value "1" / markerattrs=(symbol=circlefilled color=&C1);
        Value "2" / markerattrs=(symbol=circlefilled color=&C2);
      EndDiscreteAttrMap;
      layout overlay / xaxisopts=(label="X Variable")
        yaxisopts=(label="Y Variable")
        walldisplay=(fill);
        scatterplot x=XVAR y=YVAR / group=ATTR_TRT;
        discretelegend "one" "two" / border=false valign=bottom;
      endlayout;
    endgraph;
  end;
run;
```



```

ods graphics / imagefmt=png imagename="gtlmain2" noborder height=480px
width=640px reset=index;
ods listing gpath="C:\Trunk\TLF\Validation";
proc sgrender data=main template=mtemp2;
run;

data vhex;
  input MCOLOR $ WHITE $;
  COLOR=dequote(resolve(quote(MCOLOR)));
  drop MCOLOR;
  datalines;
  &C1 ffffff
  &C2 ffffff
  ;
run;

data vhex2;
  set vhex;
  VCOLOR="CX"||put(input(WHITE,hex6.)-
    input(strip(tranwrd(COLOR,"CX","")),hex6.),hex6.);
run;

proc sql;
  select VCOLOR
  into :VC1-
  from vhex2;
quit;

proc template;
  define statgraph vtemp2;
    beginnograph;
      legenditem type=marker name="one" / label="Placebo"
        markerattrs=(symbol=circlefilled color=&VC1 transparency=0.3);
      legenditem type=marker name="two" / label="10 mg"
        markerattrs=(symbol=circlefilled color=&VC2 transparency=0.3);
      DiscreteAttrVar attrvar=ATTR_TRT var=TRT attrmap="myattrmap";
      DiscreteAttrMap name="myattrmap";
        Value "1" / markerattrs=(symbol=circlefilled color=&VC1);
        Value "2" / markerattrs=(symbol=circlefilled color=&VC2);
      EndDiscreteAttrMap;
      layout overlay / xaxisopts=(label="Y Variable")
        yaxisopts=(label="X Variable")
        walldisplay=(fill);
        scatterplot x=XVAR y=YVAR / group=ATTR_TRT
          datatransparency=0.3;
        discretelegend "one" "two" / border=false valign=bottom;
        drawimage "C:\Trunk\TLF\Validation\gtlmain2.png" / x=0 y=100
          anchor=topleft drawspace=graphpercent transparency=0.6;
      endlayout;
    endnograph;
  end;
run;
proc sgrender data=val template=vtemp2;
run;

```

```

*****
*****
/* Validate three groups - Line plot */
*****
*****;
%let C1=CX6A5ACD;
%let C2=CX228B22;
%let C3=CXB8860B;

data zero;
  do DOSE=1 to 3;
    do TIME=0 to 5;
      output;
    end;
  end;
run;

data mline;
  set zero;
  if TIME=0 then MRESP=0;
  if TIME > 0 and DOSE=1 then MRESP=round(rannor(129)+5,0.01);
  if TIME > 0 and DOSE=2 then MRESP=round(2*rannor(129)+10,0.01);
  if TIME > 0 and DOSE=3 then MRESP=round(5*rannor(129)+15,0.01);
  if TIME > 0 then do;
    RESPU=round(MRESP+sqrt(MRESP),0.01);
    RESPL=round(MRESP-sqrt(MRESP),0.01);
  end;
run;

/* Adjust for formats */
proc format;
  value DOSEF 1="Placebo"
              2="10 mg"
              3="20 mg";
  value TIMEF 0="0"
              1="1"
              2="6"
              3="12"
              4="18"
              5="24";
run;

```

```

proc template;
  define statgraph mlines;
    begingraph;
      legenditem type=markerline name="one" / label="Placebo"
        lineattrs=(pattern=solid color=&C1)
        markerattrs=(symbol=circlefilled color=&C1);
      legenditem type=markerline name="two" / label="10 mg"
        lineattrs=(pattern=solid color=&C2)
        markerattrs=(symbol=circlefilled color=&C2);
      legenditem type=markerline name="three" / label="20 mg"
        lineattrs=(pattern=solid color=&C3)
        markerattrs=(symbol=circlefilled color=&C3);
      DiscreteAttrVar attrvar=ATTR_DOSE var=DOSE attrmap="myattrmap";
      DiscreteAttrMap name="myattrmap";
        Value "Placebo" / lineattrs=(pattern=solid color=&C1)
          markerattrs=(symbol=circlefilled color=&C1);
        Value "10 mg" / lineattrs=(pattern=solid color=&C2)
          markerattrs=(symbol=circlefilled color=&C2);
        Value "20 mg" / lineattrs=(pattern=solid color=&C3)
          markerattrs=(symbol=circlefilled color=&C3);
      EndDiscreteAttrMap;
      layout overlay / xaxisopts=(label="Study Month")
        yaxisopts=(label="Mean (95% CI) Response"
          linearopts=(viewmin=0 viewmax=40
            tickvaluesequence=(start=0 end=40
              increment=5)));
      scatterplot x=TIME y=MRESP / group=ATTR_DOSE groupdisplay=cluster
        clusterwidth=0.12 markerattrs=(size=8) yerrorupper=RESPU
        yerrorlower=RESPL;
      seriesplot x=TIME y=MRESP / group=ATTR_DOSE;
      discretelegend "one" "two" "three" / valign=bottom
        location=outside down=1 border=false title="Treatment
          Doseage";
    endlayout;
  endgraph;
end;
run;

ods graphics / imagefmt=png imagename="gtlmain3" noborder height=480px
width=640px reset=index;
ods listing gpath="C:\Trunk\TLF\Validation";

proc sgrender data=mline template=mlines;
  format TIME TIMEF.;
  format DOSE DOSEF.;
run;

data vhex;
  input MCOLOR $ WHITE $;
  COLOR=dequote(resolve(quote(MCOLOR)));
  drop MCOLOR;
  datalines;
  &C1 fffffff
  &C2 fffffff
  &C3 fffffff
  ;
run;

```

```

data vhex2;
  set vhex;
  VCOLOR="CX" || put (input (WHITE,hex6.) -
    input (strip (tranwrd (COLOR,"CX","")),hex6.),hex6.);
run;

proc sql;
  select VCOLOR
  into :VC1-
  from vhex2;
quit;

proc template;
  define statgraph vlines;
    begingraph;
      legenditem type=markerline name="one" / label="Placebo"
        lineattrs=(pattern=solid color=&VC1)
        markerattrs=(symbol=circlefilled color=&VC1 transparency=0.3);
      legenditem type=markerline name="two" / label="10 mg"
        lineattrs=(pattern=solid color=&VC2)
        markerattrs=(symbol=circlefilled color=&VC2 transparency=0.3);
      legenditem type=markerline name="three" / label="20 mg"
        lineattrs=(pattern=solid color=&VC3)
        markerattrs=(symbol=circlefilled color=&VC3 transparency=0.3);
      DiscreteAttrVar attrvar=ATTR_DOSE var=DOSE attrmap="myattrmap";
      DiscreteAttrMap name="myattrmap";
        Value "Placebo" / lineattrs=(pattern=solid color=&VC1)
          markerattrs=(symbol=circlefilled color=&VC1);
        Value "10 mg" / lineattrs=(pattern=solid color=&VC2)
          markerattrs=(symbol=circlefilled color=&VC2);
        Value "20 mg" / lineattrs=(pattern=solid color=&VC3)
          markerattrs=(symbol=circlefilled color=&VC3);
      EndDiscreteAttrMap;
      layout overlay / xaxisopts=(label="Study Month")
        yaxisopts=(label="Mean (95% CI) Response"
          linearopts=(viewmin=0 viewmax=40
            tickvaluesequence=(start=0 end=40
              increment=5)));
        scatterplot x=TIME y=MRESP / group=ATTR_DOSE
          groupdisplay=cluster clusterwidth=0.12 markerattrs=(size=8)
          yerrorupper=RESPU yerrorlower=RESPL datatransparency=0.3;
        seriesplot x=TIME y=MRESP / group=ATTR_DOSE
          datatransparency=0.3;
        discretelegend "one" "two" "three" / valign=bottom
          location=outside down=1 border=false
          title="Treatment Doseage";
        drawimage "C:\Trunk\TLF\Validation\gtlmain3.png" / x=0 y=100
          anchor=topleft drawspace=graphpercent transparency=0.6;
    endlayout;
  endgraph;
end;
run;

proc sgrender data=mline template=vlines;
  format TIME TIMEF.;
  format DOSE DOSEF.;
run;

```

```

*****
*****
/* Validate Box Plots */
*****
*****;
%let C1=CXDEB887;
%let C2=CX8B4513;
%let C3=CXDA70D6;
%let C4=CX4B0082;

data mbox;
length STAT $8;
  do TRT01PN=1 to 2;
    do AVISITN=1 to 2;
      AVAL=15*rannor(31)+100;
      STAT="MEAN";
      output;
      AVAL=15*rannor(32)+100;
      STAT="MEDIAN";
      output;
      AVAL=5*rannor(32)+50;
      STAT="MIN";
      output;
      AVAL=5*rannor(32)+150;
      STAT="MAX";
      output;
      AVAL=5*rannor(32)+75;
      STAT="Q1";
      output;
      AVAL=5*rannor(32)+125;
      STAT="Q3";
      output;
    end;
  end;
run;

```

```

data vbox;
length STAT $8;
  do TRT01PN=1 to 2;
    do AVISITN=1 to 2;
      AVAL=15*rannor(1227)+100;
      STAT="MEAN";
      output;
      AVAL=15*rannor(1227)+100;
      STAT="MEDIAN";
      output;
      AVAL=5*rannor(1227)+50;
      STAT="MIN";
      output;
      AVAL=5*rannor(1227)+150;
      STAT="MAX";
      output;
      AVAL=5*rannor(1227)+75;
      STAT="Q1";
      output;
      AVAL=5*rannor(1227)+125;
      STAT="Q3";
      output;
    end;
  end;
run;

proc format;
  value XF 1="Pre Treatment"
          2="Post Treatment";
  value TRTF 1="Placebo"
            2="Treatment";
run;

data vhex;
input MCOLOR $ WHITE $;
COLOR=dequote(resolve(quote(MCOLOR)));
drop MCOLOR;
datalines;
&C1 fffffff
&C2 fffffff
&C3 fffffff
&C4 fffffff
;
run;

data vhex2;
set vhex;
VCOLOR="CX"||put(input(WHITE,hex6.)-
input(strip(tranwrd(COLOR,"CX","")),hex6.),hex6.);
run;

proc sql;
select VCOLOR
into :VC1-
from vhex2;
quit;

```

```

proc template;
  define statgraph boxes;
    begingraph;
      DiscreteAttrVar attrvar=ATTR_TRT var=TRT01PN attrmap="myattrmap";
      DiscreteAttrMap name="myattrmap";
      Value "Placebo" / lineattrs=(pattern=solid color=&C2)
        fillattrs=(color=&C1) markerattrs=(symbol=circle color=&C2);
      Value "Treatment" / lineattrs=(pattern=solid color=&C4)
        fillattrs=(color=&C3) markerattrs=(symbol=circle color=&C4);
      EndDiscreteAttrMap;
      layout overlay / yaxisopts=(label="Response")
        xaxisopts=(label="Time Point");
      boxplotparm x=AVISITN y=AVAL stat=STAT / group=ATTR_TRT
        groupdisplay=cluster name="one";
      discretelegend "one";
    endlayout;
  endgraph;
end;
run;

ods graphics / imagefmt=png imagename="gtlmain4" noborder height=480px
width=640px reset=index;
ods listing gpath="C:\Trunk\TLF\Validation";
proc sgrender data=mbox template=boxes;
  format AVISITN XF.;
  format TRT01PN TRTF.;
run;

proc template;
  define statgraph vboxes;
    begingraph;
      DiscreteAttrVar attrvar=ATTR_TRT var=TRT01PN attrmap="myattrmap";
      DiscreteAttrMap name="myattrmap";
      Value "Placebo" / lineattrs=(pattern=solid color=&VC2)
        fillattrs=(color=&VC1) markerattrs=(symbol=circle color=&VC2);
      Value "Treatment" / lineattrs=(pattern=solid color=&VC4)
        fillattrs=(color=&VC3) markerattrs=(symbol=circle color=&VC4);
      EndDiscreteAttrMap;
      layout overlay / yaxisopts=(label="Response")
        xaxisopts=(label="Time Point");
      boxplotparm x=AVISITN y=AVAL stat=STAT / group=ATTR_TRT
        groupdisplay=cluster name="one" datatransparency=0.3;
      discretelegend "one";
      drawimage "C:\Trunk\TLF\Validation\gtlmain4.png" / x=0 y=100
        anchor=topleft drawspace=graphpercent transparency=0.6;
    endlayout;
  endgraph;
end;
run;

proc sgrender data=vbox template=vboxes;
  format AVISITN XF.;
  format TRT01PN TRTF.;
run;

```

```

*****
*****
/* Gray Scale */
*****
*****;
%let C1=CX000000;

proc template;
  define statgraph mgray;
    begingraph;
      legenditem type=markerline name="one" / label="Placebo"
        lineattrs=(pattern=1 color=&C1)
        markerattrs=(symbol=circlefilled color=&C1);
      legenditem type=markerline name="two" / label="10 mg"
        lineattrs=(pattern=2 color=&C1)
        markerattrs=(symbol=squarefilled color=&C1);
      legenditem type=markerline name="three" / label="20 mg"
        lineattrs=(pattern=14 color=&C1)
        markerattrs=(symbol=trianglefilled color=&C1);
      DiscreteAttrVar attrvar=ATTR_DOSE var=DOSE attrmap="myattrmap";
      DiscreteAttrMap name="myattrmap";
        Value "Placebo" / lineattrs=(pattern=1 color=&C1)
          markerattrs=(symbol=circlefilled color=&C1);
        Value "10 mg" / lineattrs=(pattern=2 color=&C1)
          markerattrs=(symbol=squarefilled color=&C1);
        Value "20 mg" / lineattrs=(pattern=14 color=&C1)
          markerattrs=(symbol=trianglefilled color=&C1);
      EndDiscreteAttrMap;
      layout overlay / xaxisopts=(label="Study Month")
        yaxisopts=(label="Mean (95% CI) Response"
          linearopts=(viewmin=0 viewmax=40
            tickvaluesequence=(start=0 end=40
              increment=5)));
      scatterplot x=TIME y=MRESP / group=ATTR_DOSE groupdisplay=cluster
        clusterwidth=0.12 markerattrs=(size=8) yerrorupper=RESPU
        yerrorlower=RESPL;
      seriesplot x=TIME y=MRESP / group=ATTR_DOSE;
      discretelegend "one" "two" "three" / valign=bottom
        location=outside down=1 border=false
        title="Treatment Doseage";
    endlayout;
  endgraph;
end;
run;

ods graphics / imagefmt=png imagename="gtlmain5" noborder height=480px
width=640px reset=index;
ods listing gpath="C:\Trunk\TLF\Validation";
proc sgrender data=mline template=mgray;
  format TIME TIMEF.;
  format DOSE DOSEF.;
run;

```



```

%let VC1=CXFF0000;

proc template;
  define statgraph vgray;
    begingraph;
      legenditem type=markerline name="one" / label="Placebo"
        lineattrs=(pattern=1 color=&VC1)
        markerattrs=(symbol=circlefilled color=&VC1 transparency=0.3);
      legenditem type=markerline name="two" / label="10 mg"
        lineattrs=(pattern=2 color=&VC1)
        markerattrs=(symbol=squarefilled color=&VC1 transparency=0.3);
      legenditem type=markerline name="three" / label="20 mg"
        lineattrs=(pattern=14 color=&VC1)
        markerattrs=(symbol=trianglefilled color=&VC1 transparency=0.3);
      DiscreteAttrVar attrvar=ATTR_DOSE var=DOSE attrmap="myattrmap";
      DiscreteAttrMap name="myattrmap";
        Value "Placebo" / lineattrs=(pattern=1 color=&VC1)
          markerattrs=(symbol=circlefilled color=&VC1);
        Value "10 mg" / lineattrs=(pattern=2 color=&VC1)
          markerattrs=(symbol=squarefilled color=&VC1);
        Value "20 mg" / lineattrs=(pattern=14 color=&VC1)
          markerattrs=(symbol=trianglefilled color=&VC1);
      EndDiscreteAttrMap;
      layout overlay / xaxisopts=(label="Study Month")
        yaxisopts=(label="Mean (95% CI) Response"
          linearopts=(viewmin=0 viewmax=40
            tickvaluesequence=(start=0 end=40
              increment=5)));
      scatterplot x=TIME y=MRESP / group=ATTR_DOSE groupdisplay=cluster
        clusterwidth=0.12 markerattrs=(size=8) yerrorupper=RESPU
        yerrorlower=RESPL datatransparency=0.3;
      seriesplot x=TIME y=MRESP / group=ATTR_DOSE datatransparency=0.3;
      discretelegend "one" "two" "three" / valign=bottom
        location=outside down=1 border=false
        title="Treatment Doseage";
      drawimage "C:\Trunk\TLF\Validation\gtlmain51.png" / x=0 y=100
        anchor=topleft drawspace=graphpercent transparency=0.6;
    endlayout;
  endgraph;
end;
run;

proc sgrender data=mline template=vgray;
  format TIME TIMEF.;
  format DOSE DOSEF.;
run;

```