

Stylizing your SAS graph – A needs-based approach

Jerome Lechere, Novartis, Basel, Switzerland

The opinions expressed in this presentation and on the following slides are solely those of the presenter and not necessarily those of Novartis. Novartis does not guarantee the accuracy or reliability of the information provided herein.

ABSTRACT

When producing a SAS® graph, the default rendering is determined by the ODS style associated with the active ODS destination. However SAS allows users to take control over each of the graphical attributes (color, symbol, line pattern...) to tailor results as per their needs. While a quick and partial control might be sufficient to generate a single graph, users may want to take full control over the graph procedures to produce numerous graphs in a consistent manner or to produce complex graphs. This paper should guide readers to use the most appropriate method in order to stylize a SAS graph.

INTRODUCTION

The introduction of the Statistical Graphics (SG) procedures in SAS 9.2 was a great leap forward. Generating graphs became much faster and intuitive. However, the SG procedures came along with many options and statements and their usage might be confusing at first sight. When it comes down to stylizing a graph, several solutions are available to users. We will go through three main methods - looking at their advantages and inconvenience - to help users choose the option that best suits their needs. Before drilling down into the details of each proposed solution, we will have a brief introduction on the SAS ODS styles, which are permanently acting in the background.

BEHIND THE SCENE: SAS ODS STYLES

DEFINITION

An ODS style is a collection of style elements that provides specific graphical attributes for your SAS output. Every ODS output destination has an associated default style. Therefore, your output might look different depending on which destination you use:

- Default style for the PRINTER destination is 'Printer'
- Default style for the RTF destination is 'RTF'
- Default style for the HTML and HTML5 destinations is 'HTMLBlue'

When you issue an ODS statement like:

```
ods rtf body= sample.rtf.
```

You are really issuing the following:

```
ods rtf body= sample.rtf.
style=Default;
```

LIST ODS STYLES

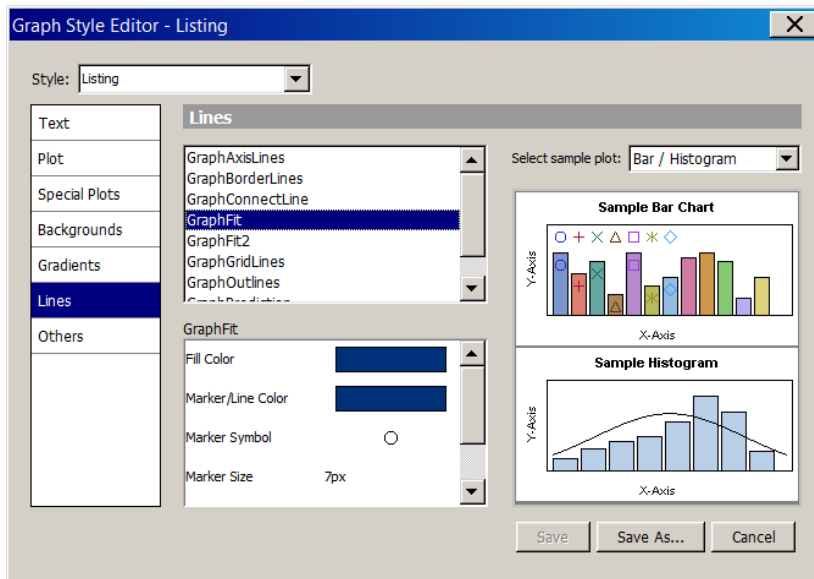
The names of the ODS styles can be accessed using the following program.

```
proc template;
list styles;
run;
```

The SAS System		
Listing of: SASHELP.TMPLMST		
Path Filter is: Styles		
Sort by: PATH/ASCENDING		
Obs	Path	Type
1	Styles	Dir
2	Styles.Analysis	Style
3	Styles.BarrettsBlue	Style
4	Styles.BlackPrint	Style
5	Styles.Daisy	Style
6	Styles.Default	Style
7	Styles.Dove	Style
8	Styles.Dtree	Style
9	Styles.EDefault	Style
10	Styles.FancyPrinter	Style
11	Styles.Festival	Style
12	Styles.FestivalPrinter	Style
13	Styles.Gantt	Style
14	Styles.GrayscalePrinter	Style
15	Styles.HTMLBlue	Style
16	Styles.Harvest	Style
17	Styles.HighContrast	Style
18	Styles.Journal	Style
19	Styles.Journal1a	Style
20	Styles.Journal12	Style
21	Styles.Journal2a	Style
22	Styles.Journal3	Style
23	Styles.Journal3a	Style
24	Styles.Listing	Style
25	Styles.Meadow	Style
26	Styles.MeadowPrinter	Style
27	Styles.Minimal	Style
28	Styles.MonochromePrinter	Style
29	Styles.Monospace	Style
30	Styles.Moonflower	Style
31	Styles.Netdraw	Style
32	Styles.NoFontDefault	Style
33	Styles.Normal	Style
34	Styles.NormalPrinter	Style
35	Styles.Ocean	Style
36	Styles.Pearl	Style
37	Styles.Plateau	Style
38	Styles.PowerPointDark	Style
39	Styles.PowerPointLight	Style
40	Styles.Printer	Style
41	Styles.Raven	Style
42	Styles.Rtf	Style
43	Styles.Sapphire	Style
44	Styles.SasDocPrinter	Style
45	Styles.SasWeb	Style
46	Styles.Seaside	Style
47	Styles.SeasidePrinter	Style
48	Styles.StatDoc	Style
49	Styles.Statistical	Style
50	Styles.valDark	Style
51	Styles.valHighContrast	Style
52	Styles.valLight	Style

PhUSE 2016

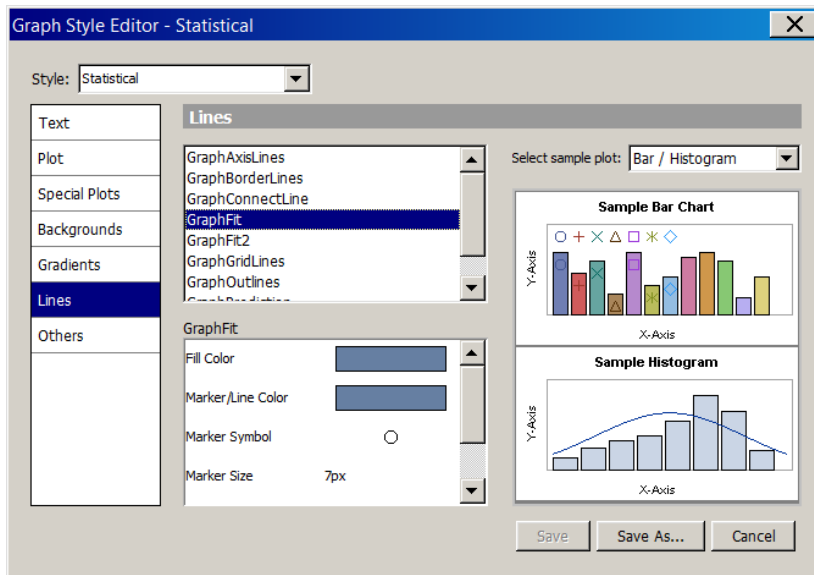
The names and characteristics of the ODS styles can be visualized via ODS Graphics Designer > Tools > Style Editor.



Style = LISTING / Category = LINES / Style element = GRAPHFIT

The GRAPHFIT style element is made of 6 graphical attributes (only 4 visible here)

The FILL COLOR graphical attribute is set to dark blue for the LISTING style



Style = STATISTICAL / Category = LINES / Style element = GRAPHFIT

The GRAPHFIT style element is made of 6 graphical attributes (only 4 visible here)

The FILL COLOR graphical attribute is set to pale blue for the LISTING style

PhUSE 2016

CHANGE THE CURRENT STYLE FOR AN ODS DESTINATION

It requires only the use of the STYLE= option on an ODS destination statement.

```
ods html style=journal
```

The journal style is applied to all output for that destination until you change or close the destination or start a new SAS session.

CREATE A NEW STYLE FROM AN EXISTING STYLE

```
proc template;
define style styles.TEST; /* New style 'TEST' */
parent = styles.LISTING; /* Existing style 'LISTING' */

style GraphData1 from GraphData1 /
lineStyle = 20
markerSymbol = 'square';

style GraphBackground from GraphBackground /
color = beige;

end;
run;
```

In the parent style 'LISTING',

- GRAPHDATA1 style element,
 - LINESTYLE graphical attribute is set to '1' (solid)
 - MARKERSYMBOL graphical attribute is set to 'circle'
- GRAPHBACKGROUND style element,
 - COLOR graphical attribute is set to 'white'

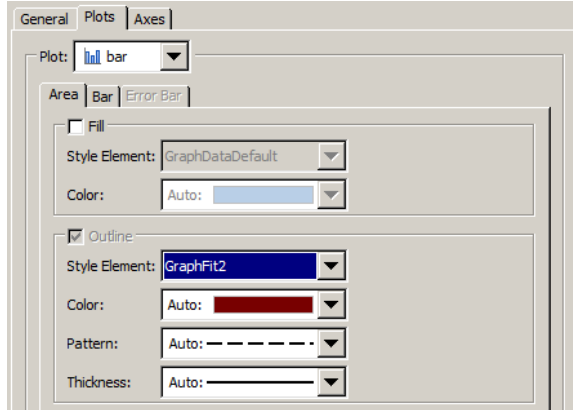
In the newly created style 'TEST',

- GRAPHDATA1 style element,
 - LINESTYLE graphical attribute is set to '20' (dash)
 - MARKERSYMBOL graphical attribute is set to 'square'
- GRAPHBACKGROUND style element,
 - COLOR graphical attribute is set to 'beige'

CHANGE A STYLE ELEMENT IN THE CURRENT STYLE

In the below example below, the default 'Graphfit' style element is changed by the 'Graphfit2' style element. As a consequence, the default graphical attributes:

- line color is changed from dark blue to red
 - line pattern is changed from solid to shortdash
- Specify a different style element with ODS Graphics Designer:



- Specify a different style element from within a SG procedure:

```
proc sgplot data=sashelp.class;
density height / lineattrs=graphfit2;
run;
```

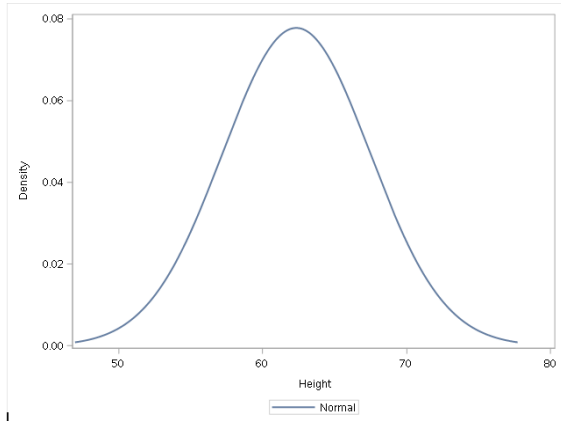
- Specify a different style element from within a GTL proc template:

```
proc template;
define statgraph style_element;
begingraph /;
<...>
scatterplot x=XVAR y=YVAR / lineattrs=graphfit2;
<...>
endgraph;
end;
run;
```

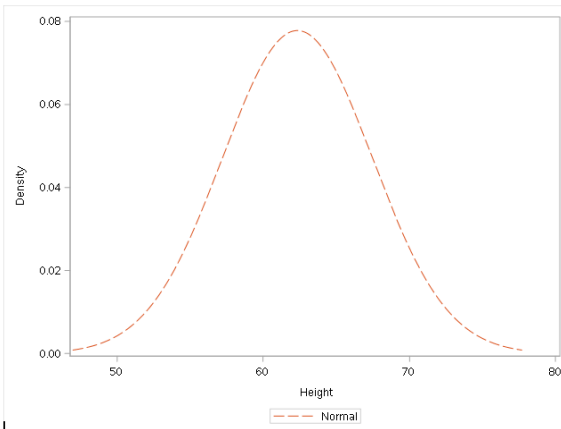
PhUSE 2016

- Change style element from 'Graphfit' to 'Graphfit2'

Style element Graphfit / line color = dark blue / line pattern = solid



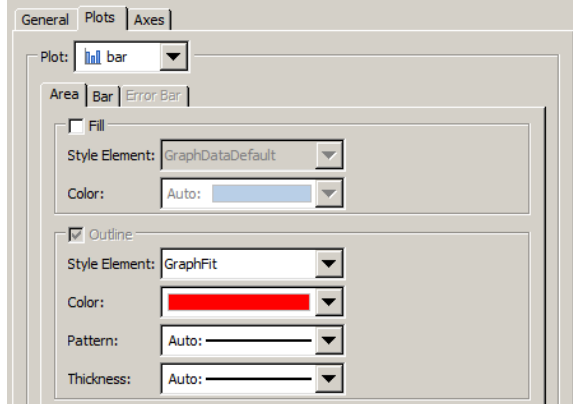
Style element Graphfit2 / line color = red / line pattern = shortdash



CHANGE A GRAPHICAL ATTRIBUTE IN A CURRENT STYLE ELEMENT

In the below example below, the default graphical attributes 'color' is changed from dark blue to red.

- Specify a different graphical attribute with ODS Graphics Designer:



- Specify a different graphical attribute from within a SG procedure:

```
proc sgplot data=sashelp.class;
density height / lineattrs=(color=red);
run;
```

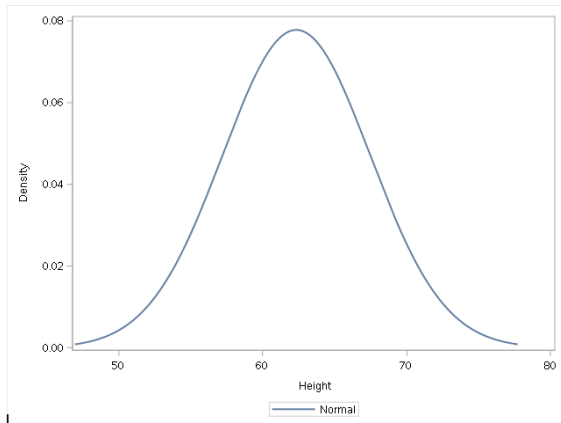
- Specify a different graphical attribute from within a GTL proc template:

```
proc template;
define statgraph app_attr;
begingraph /;
<...>
DensityPlot 'Height'n / lineattrs=(color=red);
<...>
endgraph;
end;
run;
```

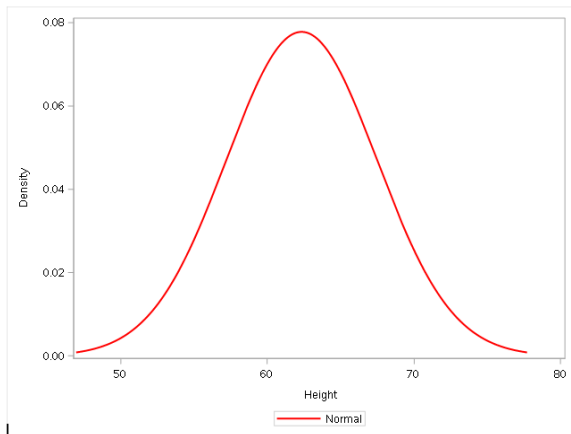
PhUSE 2016

- Change graphical attribute 'color' in the current Graphfit style element

Style element Graphfit / line color = dark blue



Style element Graphfit / line color = red



All the other graphical attributes from the Graphfit style element remain the same (solid line pattern for instance).

STYLIZING A SAS GRAPH USING THE ATTRIBUTE OPTIONS

DEFINITION

An attribute option allows the user to control a graph attribute from within a SG procedure or a GTL proc template. The option is introduced by a slash '/' and its attributes must be enclosed in parentheses and specified as a name=value pair.

```
needleplot x=x y=y / Lineattrs = (pattern=1 thickness=8);  
scatterplot x=x y=y / Markerattrs = (symbol=StarFilled);  
barchart category=x response=y / Fillattrs = (transparency=0.5);
```

SYNTAX IN A SG PROCEDURE

```
proc sgplot data=input ;  
  needle x=x y=y / Lineattrs=(pattern=1 thickness=8);  
run;
```

SYNTAX IN A GTL PROC TEMPLATE

```
proc template;  
  define statgraph wfp;  
    begingraph /;  
    <...>  
    needleplot x=x y=y / Lineattrs=(pattern=1 thickness=8);  
    <...>  
  endgraph;  
end;  
run;  
  
proc sgrender  
  data=input  
  template=wfp;  
run;
```

ILLUSTRATED EXAMPLE

- Original plot with default attributes: color = blue / thickness = 1

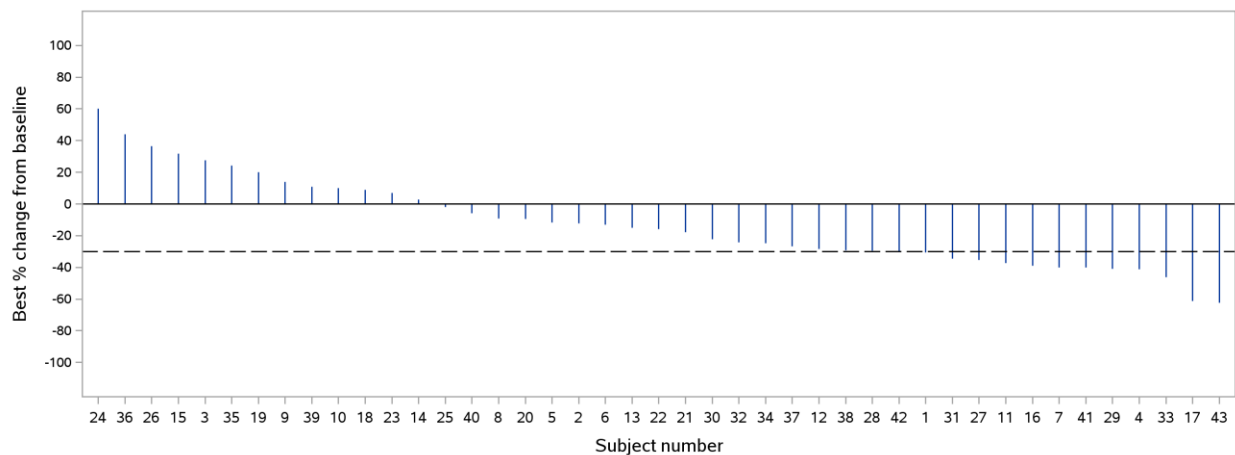
```
proc template;
define statgraph wfp;
begingraph /;
EntryTitle "Figure 1a - Best percentage change from baseline in criteria, by
treatment" /;
layout overlay/
xaxisopts=(Label="Subject number" LABELATTRS=(size=10) display=(label ticks
tickvalues) TICKVALUEATTRS=(size=8) type=linear linearopts=(tck_idx=
tickvalue=(1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43) viewmin= 0.5 viewmax=43.5
TICKVALUEFITPOLICY=ROTATETHIN))
yaxisopts=( Label="Best % change from baseline" LABELATTRS=(size=10) type=linear
TICKVALUEATTRS=(size=7)
linearopts=( tickvalue=( -100 -80 -60 -40 -20 0 20 40 60 80 100 ) viewmin=-120
viewmax=120 ) ) ;

needleplot X=x Y=y /Lineattrs=(pattern=1) name='wfp';

Entry halign=right TEXT /valign=top textattrs=(size=9);
ReferenceLine y=0 / clip=true Lineattrs=( Color=CX000000);
ReferenceLine y=-30 / clip=true Lineattrs=( Color=CX000000 PATTERN=4);
endlayout;
endgraph;
end;
run;

proc sgrender
data=input
template=wfp;
run;
```

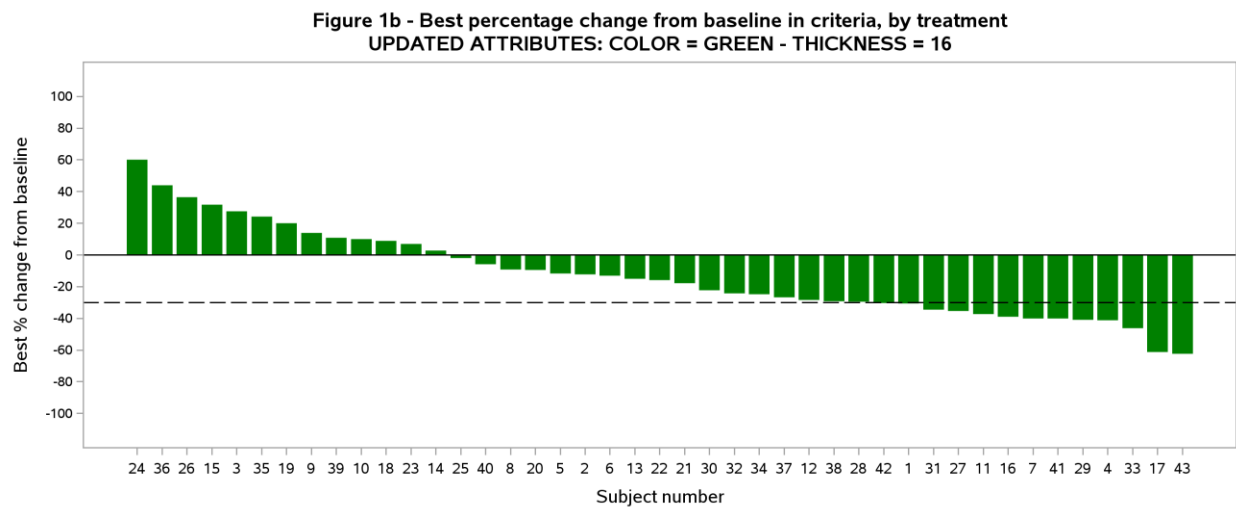
Figure 1a - Best percentage change from baseline in criteria, by treatment
DEFAULT ATTRIBUTES: COLOR = BLUE - THICKNESS = 1



PhUSE 2016

- Updated plot with new attributes: color = green / thickness = 16

```
proc template;
define statgraph wfp;
begingraph /;
<...>
needleplot X=x Y=y / Lineattrs=(pattern=1 color=green thickness=16);
<...>
endgraph;
end;
run;
```



⇒ The newly defined graphical attributes 'Color = green' and 'Thickness = 16' apply for ALL values in the plot.

PhUSE 2016

IN SUMMARY:

- ⇒ **WITH THE ATTRIBUTE OPTIONS, THE GRAPHICAL ATTRIBUTE DEFINED BY USERS IS APPLIED TO ALL VALUES IN THE PLOT. THEREFORE, USING ATTRIBUTE OPTIONS IS ACCEPTABLE IF NO GROUP VARIABLE IS INVOLVED BUT IS NOT RECOMMENDED OTHERWISE.**
- ⇒ **ACTUALLY, IF A GROUP VARIABLE IS INVOLVED, SAS TRIES ITS BEST TO DIFFERENTIATE THE DIFFERENT GROUPS BY CYCLING THROUGH THE GRAPHICAL ATTRIBUTES FROM THE ODS STYLE:**
 - if users forces 'symbol=StarFilled' SAS will apply different colors to differentiate the groups
 - if users forces 'color=Blue' SAS will apply different symbols to differentiate the groups
 - if users forces both 'symbol=StarFilled' and 'color=Blue', SAS can no longer differentiate the groups. All values on the plot will be represented by an identical Blue + StarFilled marker
- ⇒ **AS A CONSEQUENCE, IF A GROUP VARIABLE IS INVOLVED AND USERS WANT TO TAKE CONTROL OVER THE GRAPHICAL ATTRIBUTES, IT IS STRONGLY RECOMMENDED TO USE THE STYLEATTRS STATEMENT.**

STYLIZING A SAS GRAPH USING THE STYLEATTRS STATEMENT

DEFINITION

Styleattrs statement allows temporary editing of the default ODS style.

Datacolors / datacontrastcolors / datalinepatterns / datasymbols are options for the styleattrs statement.

When producing a graph where a group variable is involved:

- By default, SAS access the ODS style associated to the open ODS destination and read its attributes in sequential order. For instance, if ODS destination is HTML, SAS access the 'HTMLBlue' style and render the group 1 values in dark blue, the group 2 values in dark red and the group 3 values in dark green.
- With the Styleattrs statement, users are able to hardcode the sequence of the attributes directly in the code, for instance group 1 values in dark green, group 2 values in dark red and group 3 values in dark blue.

The group variable is define using the Group= options.

Warning 1: ODS graphics must be on for SAS to know the style attached to the open ODS destination and then modify it accordingly.

Warning 2: For SAS to execute the sequence of attributes hardcoded in the Styleattrs statement, the attrpriority option must be turned off (attrpriority=none).

SYNTAX IN A SG PROCEDURE

```
Ods graphics / attrpriority=none;
proc sgplot data=input ;
styleattrs dataContrastColors = ( CXADD8E6 CX98FB98 CXF08080 )
/* LightBlue PaleGreen LightCoral */
dataSymbols = ( CIRCLE SQUARE );
needle x=x y=y / group=trt_grp;
run;
```

SYNTAX IN A GTL PROC TEMPLATE

```
proc template;
define statgraph wfp;
begingraph / dataContrastColors=( CXADD8E6 CX98FB98 CXF08080 )
/* LightBlue PaleGreen LightCoral */
dataSymbols=( CIRCLE SQUARE );
<...>
needleplot x=x y=y / group=trt_grp;
<...>
endgraph;
end;
run;

Ods graphics / attrpriority=none;
proc sgrender
data=input
template=wfp;
run;
```

ILLUSTRATED EXAMPLE

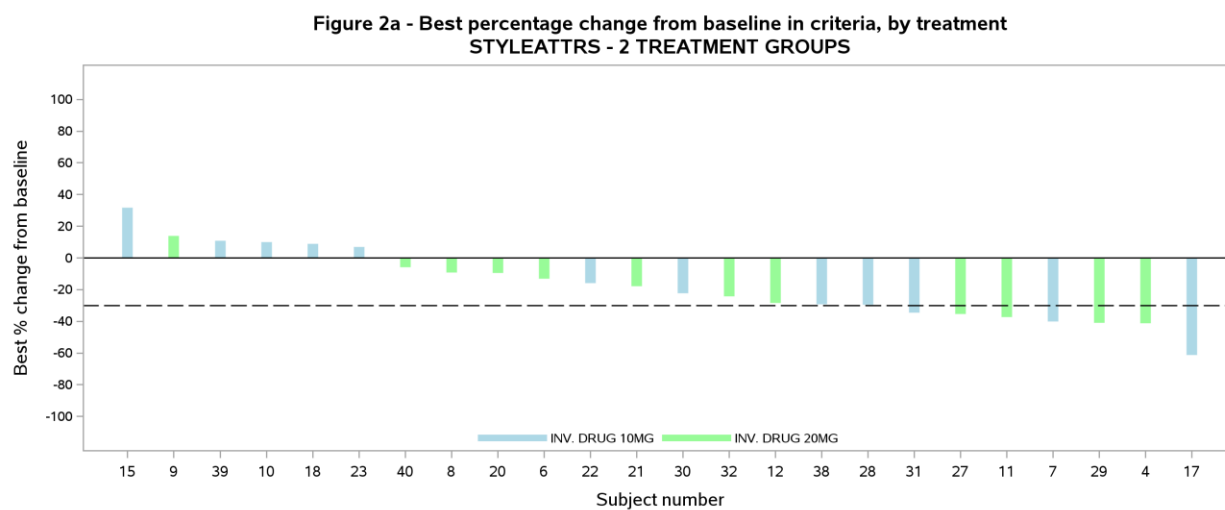
```

proc template;
define statgraph wfp;
begingraph / dataContrastColors=( CXADD8E6 CX98FB98 CXF08080 )
/* LightBlue PaleGreen LightCoral */
dataSymbols=( CIRCLE SQUARE );
<...>
needleplot x=x y=y / group=trt_grp;
<...>
endgraph;
end;
run;

Ods graphics / attrpriority=none;
proc sgrender
data=input
template=wfp;
run;

```

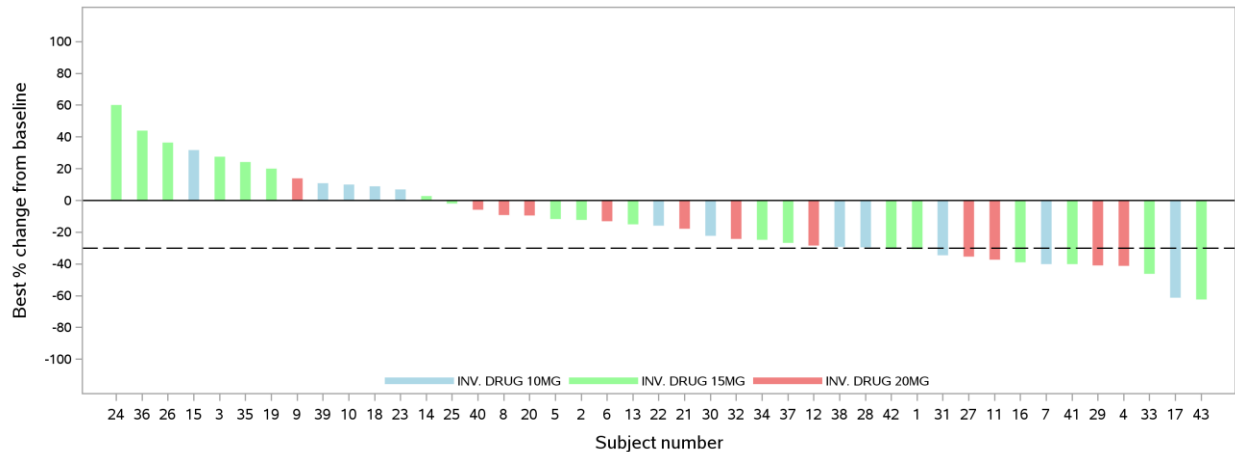
- Original plot. The input dataset shows subjects enrolled into the:
 - INV. DRUG 10MG group
 - INV. DRUG 20MG group



PhUSE 2016

- Updated plot. The input dataset now shows subjects enrolled into the:
 - INV. DRUG 10MG group
 - INV. DRUG 15MG group
 - INV. DRUG 20MG group

Figure 2b - Best percentage change from baseline in criteria, by treatment
STYLEATTRS - 3 TREATMENT GROUPS



- ⇒ The color assignment is based on the sequence of color hardcoded into the styleattrs statement: first treatment group in the dataset represented in light blue, second treatment group in the dataset represented in pale green, third treatment group in the dataset represented in light coral and so on.
- ⇒ In the first plot, subjects enrolled into the INV. DRUG 20MG group (second treatment group in the dataset) are represented in pale green (second color listed in the styleattrs statement).
- ⇒ In the first plot, subjects enrolled into the INV. DRUG 20MG group (third treatment group in the dataset) are represented in light coral (third color listed in the styleattrs statement).
- ⇒ The plot rendering is dependent from the group variable in the input dataset (sorting order, missing/existing values)

PhUSE 2016

IN SUMMARY

- ⇒ **WHENEVER A GROUP VARIABLE IS INVOLVED AND USERS WANT TO DIFFERENTIATE THE DIFFERENT 'BY' GROUPS IN THE PLOT, THE ATTRIBUTE OPTIONS MUST NOT BE USED AND THE STYLEATTRS STATEMENT SHOULD BE CONSIDERED.**
- ⇒ **WITH THE STYLEATTRS STATEMENT, USERS ARE ABLE TO HARDCODE A DESIRED SEQUENCE OF GRAPHICAL ATTRIBUTES DIRECTLY INTO A SG PROCEDURE OR A GTL PROC TEMPLATE.**
- ⇒ **THESE ATTRIBUTES ARE APPLIED TO THE INPUT DATA AS PER THE FOLLOWING RULE:**
 - First graphical attribute ⇔ Group variable, first value in the dataset
 - Second graphical attribute ⇔ Group variable, second value in the dataset
 - Third graphical attribute ⇔ Group variable, third value in the dataset
- ⇒ **USING THE STYLEATTRS STATEMENT ALLOWS USERS TO STYLIZE MANY ASPECTS OF A SAS GRAPH IN AN INTUITIVE AND QUICK MANNER. HOWEVER THE RESULT IS DEPENDENT FROM THE INPUT DATASET.**
- ⇒ **FOR USERS TO TAKE FULL CONTROL OVER THEIR SAS GRAPH, IT IS STRONGLY RECOMMENDED TO CREATE AND USE AN ATTRIBUTE MAP DATASET.**

STYLIZING A SAS GRAPH USING AN ATTRIBUTE MAP DATASET

DEFINITION

An attribute map dataset is a specific dataset holding all the required information to associate graphical attributes to values from the input dataset.

This dataset must include the following 2 reserved variables:

- ID = attribute map identifier. ID identifies each attribute map. An attribute map dataset can contain more than one attribute map. This capability enables users to apply different attribute maps to different group variables in a plot. The example below shows 2 attribute map: 'trt' for treatment and 'resp' for response.
- VALUE = group value. This variable lists the different possible values for a define group variable. The example below shows that the treatment group variable can take 3 different values in the input dataset: 'INV. DRUG 10MG', 'INV. DRUG 15MG' and 'INV. DRUG 20MG'. The response group variable can take 5 different values in the input dataset: 'CR', 'PR', 'SD', 'PD' and 'UNK'.

Beside ID and VALUE, users need to define the attribute variables of interest such as FILLCOLOR=, LINECOLOR=, LINEPATTERN=, MARKERSYMBOL, and so on. The example below shows that the different values for the treatment variable will be differentiated in the plot by using different line colors.

SYNTAX TO CREATE AN ATTRIBUTE MAP DATASET

The most commonly used method for creating an attribute map dataset is via a data step using the 'datalines' statement:

```
data attrds;
infile datalines delimiter=' ' missover;
length ID $8 VALUE $40 linecolor $12 markersymbol $40 ;
input ID VALUE linecolor markersymbol;
datalines;
trt,INV. DRUG 10MG,LightBlue
trt,INV. DRUG 15MG,LightCoral
trt,INV. DRUG 20MG,PaleGreen
resp,CR, ,starfilled
resp,PR, ,star
resp,SD, ,circle
resp,PD, ,trianglefilled
resp,UNK, ,hash
;
run;
```

	ID	VALUE	linecolor	markersymbol
1	trt	INV. DRUG 10MG	LightBlue	
2	trt	INV. DRUG 15MG	LightCoral	
3	trt	INV. DRUG 20MG	PaleGreen	
4	resp	CR		starfilled
5	resp	PR		star
6	resp	SD		circle
7	resp	PD		trianglefilled
8	resp	UNK		hash

Each observation defines the attribute(s) for a particular value of a group variable e.g. Observation 1 indicates that each value from the input dataset where treatment = 'INV. DRUG 10MG' will be rendered in a light blue color on the plot.

PhUSE 2016

SYNTAX TO ACCESS AN ATTRIBUTE MAP DATASET FROM WITHIN A SG PROCEDURE

```
proc sgplot data=input dattrmap= attrds;  
  needle x=x y=y / group=trt_grp attrid=trt;  
run;
```

SYNTAX TO ACCESS AN ATTRIBUTE MAP DATASET FROM WITHIN A GTL PROC TEMPLATE

```
proc template;  
  define statgraph wfp;  
    begingraph /;  
    <...>  
    needleplot x=x y=y / group=trt_grp;  
    <...>  
  endgraph;  
end;  
run;
```

```
proc sgrender  
  data=input  
  template=wfp  
  dattrmap=attrds;  
  dattrvar trt_grp="trt";  
run;
```

ILLUSTRATED EXAMPLE

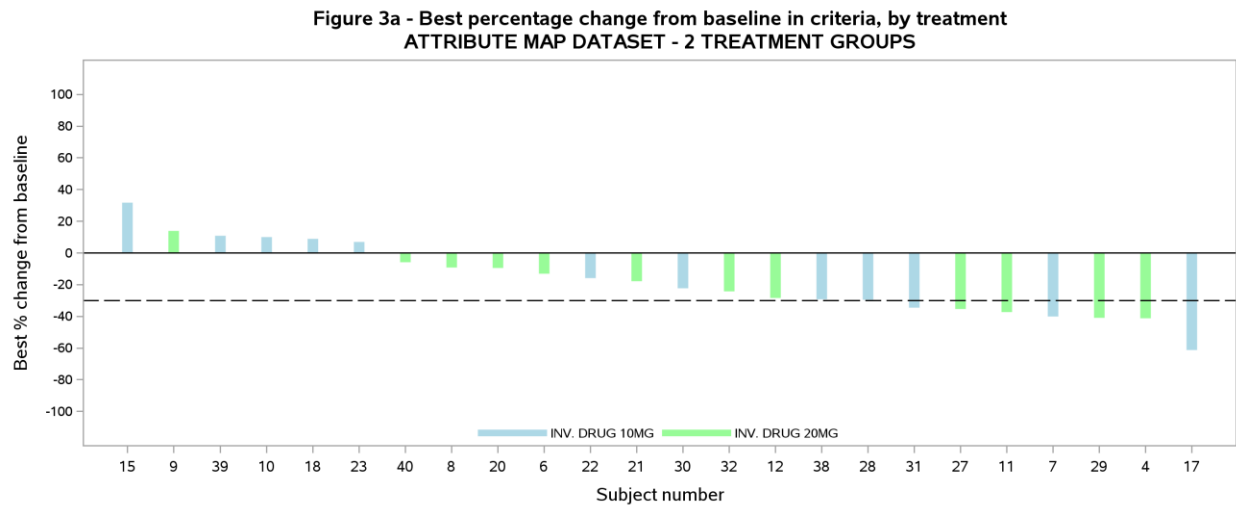
```

proc template;
  define statgraph wfp;
    begingraph /;
    <...>
    needleplot x=x y=y / group=trt_grp ;
    <...>
  endgraph;
end;
run;

proc sgrender
  data=input
  template=wfp
  dattrmap=attrds;
  dattrvar trt_grp="trt";
run;

```

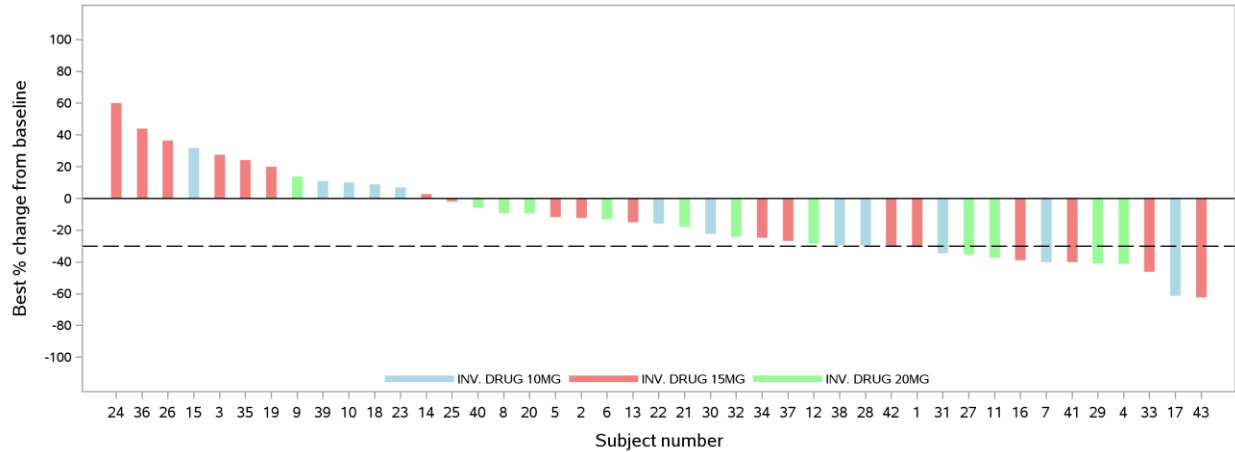
- Original plot. The input dataset shows subjects enrolled into the:
 - INV. DRUG 10MG group
 - INV. DRUG 20MG group



PhUSE 2016

- Updated plot. The input dataset now shows subjects enrolled into the:
 - INV. DRUG 10MG group
 - INV. DRUG 15MG group
 - INV. DRUG 20MG group

Figure 3b - Best percentage change from baseline in criteria, by treatment
ATTRIBUTE MAP DATASET - 3 TREATMENT GROUPS



- ⇒ The color assignment is based on the attribute map dataset: INV. DRUG 10MG is assigned to color light blue, INV. DRUG 15MG is assigned to color light coral and INV. DRUG 20MG is assigned to color pale green.
- ⇒ Whether the input dataset has 2 values for the 'by' variable (subjects enrolled into the INV. DRUG 10MG group + INV. DRUG 20MG group) or 3 values for the 'by' variable (subjects enrolled into the INV. DRUG 10MG group + INV. DRUG 15MG group + INV. DRUG 20MG group) does not impact the color assignment in the plot.
- ⇒ The plot rendering is independent from the group variable in the input dataset.

PhUSE 2016

IN SUMMARY

- ⇒ **WHENEVER A GROUP VARIABLE IS INVOLVED AND USERS WANT TO CONTROL DISPLAY OF THE DIFFERENT 'BY' GROUPS IN THE PLOT, THE CREATION OF AN ATTRIBUTE MAP DATASET SHOULD BE CONSIDERED.**
- ⇒ **AN ATTRIBUTE MAP DATASET ENABLES USERS TO HARDCODE A LIST OF GRAPHICAL ATTRIBUTES, HARDCODE A LIST OF POSSIBLE VALUES FOR THE 'BY' VARIABLE AND PAIR THEM ONE BY ONE IN A FIXED MANNER:**
 - Graphical attribute X ⇔ Group variable, value A
 - Graphical attribute Y ⇔ Group variable, value B
 - Graphical attribute Z ⇔ Group variable, value C
- ⇒ **THE GRAPHICAL ATTRIBUTES ASSIGNMENT WILL ALWAYS BE THE SAME, IRRESPECTIVE OF THE INPUT DATASET.**
- ⇒ **THEREFORE, WHILE ATTRIBUTE OPTIONS AND STYLEATTRS STATEMENT ARE TECHNIQS TO BE USED ON A PLOT BY PLOT BASIS ONLY, THE ATTRIBUTE MAP DATASET IS THE RECOMMENDED APPROACH WHEN USERS NEED TO PRODUCE SEVERAL GRAPHICS ON A PROJECT/STUDY AND WANT TO ENSURE THAT THE SAME ATTRIBUTES WILL BE USED CONSISTENTLY ACROSS ALL PLOTS.**

CONCLUSION

Looking at a few SAS versions back, it used to be quite tedious to control graphical attributes, creating for instance an ANNOTATE dataset, or most recently using the TEMPLATE procedure to create new tailored ODS styles. With SAS 9.2, it became easier to stylize a graph. Several options are available to users:

- A quick and easy manner to proceed is to use an attribute option (e.g. lineattrs) from within the SG procedures.
- For graphs where a group variable is involved and users need to distinguish each group in the plot, it is recommended to abandon the attribute options to the favor of the Styleattrs statement.
- Finally, users are even given the possibility to externalize the graphical attributes information into a dedicated dataset (Attribute Map Dataset) in order to take full control over the graph appearance.

REFERENCES

1. Amadeus Software 'Fundamental ODS Graphics' course. <http://www.amadeus.co.uk/index.php>.
2. <http://support.sas.com/documentation/cdl/en/odsug/67921/HTML/default/viewer.htm#n1ooja9ez1hlrt1s2bilfpywp03.htm>
3. <http://support.sas.com/documentation/cdl/en/grstatproc/67909/HTML/default/viewer.htm#p1dt33l6a6epk6n1chtynsgsigit.htm>
4. <http://support.sas.com/documentation/cdl/en/grstatproc/67909/HTML/default/viewer.htm#p0teue0ztycuidn14cciruv038ur.htm>

ACKNOWLEDGMENTS

Thank you to Nassim Sleiman and Mathias Lebreton for their contributions to this paper.

CONTACT INFORMATION

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

Jerome Lechere
Senior Statistical Programmer
Oncology Biometrics and Data Management
Novartis Pharma AG
Postfach CH-4002 Basel
Switzerland
Work Phone: +41 61 69 61232
Email: jerome.lechere@novartis.com
Web: www.novartis.com

Brand and product names are trademarks of their respective companies.