ABSTRACT

In clinical studies, researchers are often interested in the effect of treatment over time for multiple treatments or dosing levels. Usually, in a graphical report, the measurement of treatment effect is in the vertical axis and a second factor, such as time and visit number, in the horizontal axis. Multiple lines are displayed in the same figure; each line represents a third factor, such as treatment or dosing groups. It is critical that the line appearances (color, symbol and style) are consistent throughout the entire clinical report, as well as, across clinical reports from related studies.

Flavin and Carpenter (2004) showed that the GPLOT procedure, by default, did not guarantee consistency in line appearances. They provided macro and non-macro solutions to this problem. With the introduction of Statistical Graphics in SAS® version 9.2, there are multiple approaches to resolve this problem. In this paper, the author will cover the following topics:

- The Nature of SGPLOT (How are Line Attributes Assigned?)
- Re-visit the Line Inconsistency Problem by means of SGPLOT Procedure
- 5 Solutions to Resolve Line Inconsistency Issues
- The SGPANEL Procedure (as a Special Case of SGPLOT Procedure)

KEYWORDS
Legend, SGPLOT, GTL, SGPANEL, Index option, Attribute Map

INTRODUCTION

In the analysis of clinical data, often times only a subset of data is required. For instance, a treatment group may be excluded from an exploratory analysis. Since SAS assigns line attributes (line color, symbol and style) based on the available data, a subset of the original data may result in inconsistency in the line appearance among treatment groups. This can cause confusion and make the graph difficult to interpret. In this paper, the author illustrates several approaches to resolve this line inconsistency issue.

THE NATURE OF SGPLOT (HOW ARE LINE ATTRIBUTES ASSIGNED?)

To understand how line attributes are assigned, please consider Figure 1A: Average Score over Time for 4 Treatments: Placebo, Drug A, Drug B and Drug C. The result of each treatment is displayed as a line in the graph.

![Figure 1A: Average Score over Time for 4 Treatments](image)

Based on the legend, the order of Treatment Groups is based on the order of group variable (TRT) in the input data DRUG. This also affects how treatment groups are associated with Line Styles.

In GPLOT, the default order is ASCENDING order of the internal values of the group variable; in SGPLOT (SAS V9.2) the default is the order of group values in the INPUT DATA.
With the SG procedures, SYMBOLn statements in GPlot procedures are no longer relevant. One way to define line attributes is through the use of style GraphData1 from GraphData1 statements within PROC TEMPLATE. In addition, in SAS Version 9.2, the association of line attributes with treatment groups has been changed. For GPlot procedure, the association of a group variable with SYMBOLn statements is based on the internal values of the group variable in ascending order. In other words, the group with the lowest internal values is associated line attributes for SYMBOL1; second lowest is associated with SYMBOL2, etc. With SAS Version 9.3, one can use the following statement to ensure that the treatments are displayed in ascending order.

```sas
ods listing style=STYLES.MYSTYLE; * Use STYLES.MYSTYLE for the plot. *-
proc sgplot data=DRUG;
  series y=SCORE x=MTH / group=TRT markers;
keylegend/noborder title='Treatment: ';
run;
```
GROUP=variable and GROUPORDER = ascending | descending| data options in the SERIES statement specify the ordering of lines based on the value of the group variable. Since such feature is not available prior to SAS Version 9.3, SAS Version 9.2 users would have to sort the data based on the internal values of the group variables, if such is their desirable order to display the treatment groups. It is advisable to explicitly state the value for GROUPORDER options.

To be consistent with GPLOT default results, the author has sorted the data in the ascending order of the variable TRT and produced Figure 1B.

**Code for Figure 1B: Sort the data by Treatment Group. Then use SGPLOT procedure.**

```plaintext
*---- Sort the data by treatment group variable (TRT) so that lines are displayed ---*;
*---- by its internal values, instead of its order in the input dataset. ---*;
proc sort data=DRUG out=DRUG_BY_TRT;
   by TRT;
run;

*---- Produce Figure 1B based on the order of the internal value of TRT. ---*;
ods listing style=STYLES.MYSTYLE;
title 'Score Over Time by Treatment';
proc sgpplot data=DRUG_BY_TRT;
   series y=SCORE x=MONTH / group=TRT markers;
   keylegend/noborder title='Treatment: ';
run;
```

**Figure 1B:** After sorting by the group variable TRT, the order of treatments in the legend and the line attributes have been changed.

After sorting by group variable TRT, the default order of DATA is the same as the ASCENDING order of treatments, which matches the default order of PROC GPLOT.
THE PROBLEM

In the analysis of clinical data, researchers are often interested in only a subset of the data. What will happen to the SGPLOT output if one treatment, say, Drug B is excluded from analysis? In Figure 2, Drug B has been excluded. Note without Drug B, Drug A still has the lowest internal value, namely, ‘Drug A’. Hence, the line attributes for Drug A remain intact. However, Drug C and Placebo, having the second and third lowest internal values, take on the second and third line attributes, instead of the third and fourth line attributes as they did in the full analysis. Inconsistent line attributes is confusing and misleading. It makes interpretation of graphic results difficult for researchers! Please see Figure 2.

Figure 2: The same line inconsistency issues in the GPLOT procedure now re-appears in the output for SGPLOT procedure when Drug B is excluded.

**Code for Figure 2: SGPLOT procedure.**

```sas
*---- Subset data ordered by TRT to exclude Drug B. ---*;
Data DRUG_noB;
   set DRUG_BY_TRT (where=(TRT ne 'Drug B'));
run;

*---- Produce Figure 2 without Drug B. ---*;
*---- Order is still based on the internal values of TRT. ---*;
ods listing style=STYLES.MYSTYLE;
title 'Score Over Time by Treatment';
proc sgplot data=DRUG_noB;
   series y=SCORE x=MONTH / group=TRT markers;
   keylegend/noborder title='Treatment: '; run;
```

Flavin and Carpenter (2004) has addressed this issue with the GPLOT procedure. They have provided macro and non-macro solutions to the line inconsistency problem. So will the same solutions for the GPLOT procedure applicable to the problem in the SGPLOT procedure?
THE SOLUTIONS

5 solutions are provided below. The first 2 are analogous to the solutions suggested by Flavin and Carpenter (2004) to resolve this issue in GPLOT procedure. A third solution is suggested by the author and is also applicable to both GPLOT and SGPLOT procedures. Finally, the last 2 approaches take advantages of new features in Statistical Graphics for SAS Version 9.2 and beyond.

1. Use Dummy Data

Will Dummy Data Approach work for SGPLOT procedure? Remember in SAS 9.2, treatment groups are to be displayed according to its order of the input dataset. So it is important that after including the dummy dataset to the original input dataset, the combined dataset will retain the same DATA order for treatment. Despite the line attribute now appears to be consistent, the legend still displays non-existent treatment, Drug B. Hence the Dummy Data Approach still remains to be the less desirable approach! Please see Figure 3A, an output from Dummy Data Approach!

![Average Scores Over Time by Treatment](image)

Figure 3A: The Add Dummy Data Approach ensures line consistency. However, the excluded treatment group Drug B remains in the legend.

```sas
* Create DUMMY dataset so that all treatment groups are represented.*;
data DUMMY;
  length TRT $7.;
  TRT='Drug A';  MONTH=.; SCORE=.; output;
  TRT='Drug B';  MONTH=.; SCORE=.; output;
  TRT='Drug C';  MONTH=.; SCORE=.; output;
  TRT='Placebo';  MONTH=.; SCORE=.; output;
run;
*--------------------------------------------------------------------*
* Note: There is no need to sort DRUG3A because: *;
* As long as Drug A, Drug B, Drug C and Placebo are ordered first, *;
* second, third and fourth in INPUT data, sorting is not necessary. *;
* DUMMY dataset already has had the correct order. *;
*--------------------------------------------------------------------*
```

Drug B is not supposed to be in this figure and hence, it should not be in the legend! This can be confusing to readers!
**Code for Figure 3A: Add Dummy Data (Continued)**

DATA DRUG_WT_DUMMY;
   set DUMMY
       DRUG_noB;
run;

*--- Generate Figure 3A. ---*
*--- STYLES.MYSTYLE as defined in Code for Figure 1A. ---*
ods listing style=STYLES.MYSTYLE;
title 'Average Score Over Time by Treatment';
proc sgplot data=DRUG_WT_DUMMY;
   series y=SCORE x=MONTH / group=TRT markers;
   keylegend/noborder title='Treatment: ';
run;

2. Re-map Line Attributes

Flavin and Carpenter (2004) has provided detailed examples on how to dynamically modify the SYMBOLn statements to ensure line consistency when using the G PLOT procedure. Their approach relies heavily on %SCAN to re-map the treatment attributes to the correct treatment. The same approach can be used to modify the style GraphData n from GraphData statements within the TEMPLATE procedure. Here, the author illustrates how this can also be accomplished by means of a macro named %lineattr. Instead of %scan, CALL SYMPUT will be used.

Figure 3B: 3 remaining treatments are represented by lines consistent in appearance as those seen in Figure 1B. Legend reflects only the lines displayed in the figure.

Note: Even without Drug B, the 3 other treatments still have the same line attributes. This is the goal of this paper!
**Code for Figure 3B: (Re-map Line Attribute Approach)**

*- Establish a LOOKUP dictionary for line attribute for each drug. -*;

data LOOKUP;
    input ID & $7. LINESTYLE : 3. COLOR : $6. SYMBOL : $12.;
    cards;
    Drug A  1 red  circle
    Drug B  2 blue square
    Drug C  3 green diamond
    Placebo 4 orange circlefilled
;
run;

*- Define LINEATTR macro to create Style Template with line attribute. -*;

%macro lineattr (indata= /* Enter Input Data Set Name. */
,lookup=LOOKUP /* Enter Lookup Data Set. */
,idvar=ID /* ID variable in Lookup Data Set. */
,groupvar=TRT /* Enter Group Variable. */
,style=STYLES.MYSTYLE /* Enter Style name to be output */
,parent_style=STYLES.MYSTYLE2 /* Enter Parent Style. */
)/des='To create line attributes in Style Template';

%local i;

*- Get the line attribute for treatment groups from LOOKUP. ---*;

*** Note: only line attributes for the treatment groups in the dataset ---*;

*** specified in &indata will be kept. ---*;

proc sql noprint;
    create table LINEATTR as
    select distinct L.* , D.&GROUPVAR
    from &indata D
    left join &LOOKUP L
    on D.&GROUPVAR = L.&IDVAR
    order by D.&GROUPVAR;
quit;

*- Assign value for the line style, colors and symbol of each treatment. -*;

data _null_; set LINEATTR;
    call symput('LINESTYLE'||strip(put(_n_, 3.)), LINESTYLE);
    call symput('COLOR'||strip(put(_n_, 3.)), COLOR);
    call symput('SYMBOL'||strip(put(_n_, 3.)), strip(SYMBOL));
run;

*- Re-define your line attributes in a new style. ---*;

proc template;
    define style &STYLE;
    parent=&parent_style;
    %do i=1 %to &sqlobs;
        style GraphData&i from GraphData&i/linestyle=&LINESTYLE&i
        contrastcolor=&COLOR&i markersymbol="&SYMBOL&i";
    %end;
    %end;
run;
%mend lineattr;

*** Invoke %lineattr to create STYLES.MYSTYLES2 with the correct line styles. ---*;

*** Generate line plot using this new style. ---*;

%lineattr(indata=DRUG_noB, style=STYLES.MYSTYLES2);
ods listing style=STYLES.MYSTYLES2;
title h=20pt 'Score Over Time by Treatment';
proc sgplot data=DRUG_noB;
    series y=SCORE x=MONTH / group=TRT markers;
    keylegend/noborder title='Treatment: ':'
run;
3. Re-Code Internal Values of Group Variable

In Figure 1B, Placebo is listed last in the legend because its internal value ‘Placebo’ is behind the internal values of other treatments, namely, ‘Drug A’, ‘Drug B’ and ‘Drug C’. If Placebo, instead of Drug B, has been excluded from that figure, the line appearance for the remaining groups would have been unaffected. Hence, a third solution is to re-code so that the treatment groups to be excluded would have higher group order than those for the groups that are to stay. When using PROC GPLOT, the group order is the internal value of the group variable. For PROC SGPLOT in SAS Version 9.2, the group order is based on the order of the group variable in the input data. As an example, please consider the following code which can also generate Figure 3B.

**Code for Figure 3B (Using Re-code Internal Values Approach)**

```sas
*--- Define formats for re-coding group variable, TRT. ---*;
proc format;
  invaluenotr "Drug A" = 1
  "Drug B" = 4
  "Drug C" = 2
  "Placebo" = 3;
value TRTF 1 = "Drug A"
    2 = "Drug C"
    3 = "Placebo"
    4 = "Drug B";
run;

*-- Re-coded the value from TRT to TRT2 so that treatment groups to be dropped --*;
*--- have higher internal values and will be listed last. ---*
data DRUG_RECODED;
  set DRUG_noB;
  TRT2=input(TRT, intrtf.);
  label TRT2="Treatment: ";
  format TRT2. TRTF.;
run;

*--- Sort the data based on the internal value of the re-coded variable TRT2. ---*;
proc sort data=DRUG_RECODED out=DRUG_RECODED;
  by TRT2;
run;

* Re-order line attributes to match internal values of the group variable, TRT2.*;
proc template;
  define style STYLES.MYSTYLE3;
    parent=STYLES.MYSTYLE;
    style GraphData1 from GraphData1/linestyle=1 contrastcolor=red
      markersymbol='circle';
    style GraphData2 from GraphData2/linestyle=3 contrastcolor=green
      markersymbol='diamond';
    style GraphData3 from GraphData3/linestyle=4 contrastcolor=orange
      markersymbol='circlefilled';
    style GraphData4 from GraphData4/linestyle=2 contrastcolor=blue
      markersymbol='square';
  end;
run;

*- Generate Figure 3B that excludes Drug B with Re-coded variable as the group -*;
*-- variable. ---*
ods listing style=STYLES.MYSTYLE3;
  title 'Score Over Time by Treatment';
  proc sgplot data=DRUG_RECODED;
    series y=SCORE x=MONTH/group=TRT2 markers;
    keylegend/noborder title='Treatment: ';run;
run;
```
Based on the Code for Figure 3B, the variable TRT has been re-coded to create another variable, TRT2, which has internal values of 1, 2, 3 and 4. Moreover, TRT2=1 represents Drug A; TRT2=2 represents Drug C; TRT2=3 represents Placebo and TRT2=4 represents Drug B. The legend in Figure 3B is displayed in the order of the input data for TRT2 (which is the same order as the internal value of TRT2) and displayed in TRTF. format. Style GraphData statements have been re-arranged so that line attributes for the treatments are consistent with the internal values of TRT2. This new style STYLES.MYSTYLE3 is used to create Figure 3B, an ideal graph based on Re-code Internal Value approach. Note that this method can also be applied to PROC GPLOT, in which case the SYMBOLn statements will be modified.

4. Use INDEX Option in Graph Template Language (GTL)

Graph Template Language (GTL) empowers SAS users with the ability to create high-quality sophisticated statistical graphics with relative easiness. While the SGPLOT procedure provides a quick and easy solution to single-cell plot, GTL is designed to create complex multi-cell graphics with overlay of multiple plots. With GTL, one first defines a graphic template to be compiled. Then the graph is rendered by means of the GRENDER procedure. For the line inconsistency issue, INDEX option in SERIESPLOT statement offers yet another solution.

Consider the code in the following box for this GTL approach. Users first have to create a variable, namely, IDVAL, to be specified in the INDEX options of SERIESPLOT statement. The values in this IDVAL variable are associated with the style GraphData statements in styles definition. This is a relatively painless approach to resolve the line consistency issue. Unfortunately, the INDEX option is not available in the SGPLOT procedure. If the GTL code appears complex and formidable, don’t panic. If a graph can be created by means of SGLOT procedure, one can easily use TMPLOUT options in SGPLOT to output the corresponding GTL code. Then simply add INDEX options in the SERIESPLOT statement to create the graph template. This template can then be rendered to create a graph without the line consistency issue and hopefully, without tears from the developer!

**Code for Figure 3B (Using INDEX Option in Graph Template Language (GTL) Approach)**

```sas
data DRUG_IDVAL;
set DRUG_noB;
  *--- Create variable IDVAL to associate the group variables with ---*
  *--- line attributes defined in STYLES.MYSTYLE. ---*
  if TRT='Drug A' then IDVAL=1;
  else if TRT='Drug B' then IDVAL=2;
  else if TRT='Drug C' then IDVAL=3;
  else if TRT='Placebo' then IDVAL=4;
run;

proc template;
  *--- Template SGPLOT was obtained from TMPLOUT options in PROC SGPLOT. ---*
  *--- INDEX options in the SERIESPLOT statement has been added manually. ---*
  define statgraph SGPLOT;
begingraph;
  EntryTitle "Score Over Time by Treatment" / textattrs=(size=2pct);
  layout overlay;
    SeriesPlot X=MONTH Y=SCORE / 
      primary=true group=TRT index=IDVAL display=(markers) 
      LegendLabel="Average Score" NAME="SERIES";
    DiscreteLegend "SERIES" / Location=Outside Title="Treatment: ";
    Border=false;
  endlayout;
endgraph;
end;
run;

*--- Use PROC SGRENDER to render the graph from the stored graphics ---*
*--- template SGPLOT. ---*
ods listing style=STYLE.MYSTYLE;
proc sgrender data=DRUG_IDVAL template=SGLOT;
run;
```
5. Use of SG Attribute Map (Available in SAS Version 9.3)

SAS Institute released the 9.3 version of its product on July 11, 2011, its 35th Anniversary Date. This new release has a powerful feature to ensure consistency in line appearance --- SG Attribute Maps. With SG Attribute Map dataset, users specify ID variable values, group variable values and line attributes in one dataset. This dataset and its ID variables are then referred in the SGPLOT procedure. As an illustration, the author provides the following code to demonstrate yet another effortless method to produce Figure 3B in the brave new world of SAS Version 9.3. In event that one needs to create more sophisticated graphics, the SG Attribute Map feature is also available in GTL.

**Code for Figure 3B (Using SG Attribute Map Approach)**

```sas
*-- Create Attribute Map Data Set named MYATTRMAP. --*;
data MYATTRMAP;
  retain ID "MYID";
    markercolor : $6.;
cards;
  Drug A  1  red   circle   red
  Drug B  2  blue  square  blue
  Drug C  3  green diamond green
  Placebo 4  orange circlefilled orange
;run;
*-- Reference to MYATTRMAP to define the line attributes. --*;
ods listing;
title 'Score Over Time by Treatment';
proc sgplot data=DRUG_noB dattrmap=MYATTRMAP;
  series y=SCORE x=MONTH / group=TRT attrid=MYID marker;
  keylegend/noborder title='Treatment: '; run;
```

THE SGPANEL PROCEDURE

In early phase clinical trials, researchers are often interested in various lab test results of individual subjects over time. For example, in diabetics studies, the measurement of HbA1c of individual subjects and their body mass index (BMI) are critical to the study. Such results can be nicely presented in a panel of graphic cells, such as Figure 4A.

**Figure 4A**: Lines are inconsistent with Figure 1B. Treatment attributes are based on order of BMI, Subject ID and Treatment in input data.

**Figure 4B**: Lines are consistent with Figure 1B after applying the INDEX=options in GTL.
In Figure 4A, Placebo group happens to be the first group in the ascending order of BMI and Subject ID; hence, it has adopted the first line attributes defined in STYLES.MYSTYLE. Subsequent groups are assigned line attributes in the same manner. For multiple-cell graph like Figure 4A, each cell can be thought of having multiple missing groups. Sorting the input data in the order of the panel variables (BMI and SUBJID) and group variable (TRT) may not have any consequence on line attribute assignment. Therefore, in order to maintain consistency in line appearances, it is critical that one of the 5 methods discussed earlier in this paper is applied. More modifications are required for the first 3 methods since now more variables are needed to identify the treatment attributes. One has to take into consideration the values of BMI and SUBJID, as well as, the TRT group variable in this example. Hence, the author would recommend SAS programmers to use the last 2 methods. As an example, Method 4 has been applied to convert a multi-cell plot that shows line inconsistency (Figure 4A) to one without this problem (Figure 4B).

**Code for Figure 4A (To Create Multi-Cell Graph with Based on Order in Input Data)**

```sas
ods listing style=STYLES.MYSTYLE;
title h=20pt 'HbA1c (%) by Month';
proc sgpanel data=HBA1C
tmplout='c:\wuss_2011\Figure4B.sas';
panelby BMI SUBJID / layout=panel columns=3 rows=3 colheaderpos=top rowheaderpos=right uniscale=row start=topleft spacing=5;
series x=MONTH y=HBA1C/group=TRT name='HBA1C'
    lineattrs=(pattern=solid thickness=3) markers;
keylegend 'HBA1C' noborder position=bottom title="Treatment: ";
colaxis values=(0 3 6 9 12) label='Month';
rowaxis label='HbA1c (%)';
run;
```

**Code for Figure 4B (To Create Multi-Cell Graph with Consistency in Line Appearance)**

```sas
*--- Define a template named GSPANEL. ---*
proc template;
define statgraph SGPANEL;
dynamic _xviewmin_ _xviewmax_ _yviewmin_ _yviewmax_;
dynamic _panelnumber_ _byline_;
beigingraph / designwidth=640 designheight=640;
EntryTitle "HbA1c (%) by Month" / textattrs=( size=20pt);
layout gridded / rowgutter=5;
    layout datapanel classvars=( BMI SUBJID) / sparse=false
        includeMissingClass=false rowgutter=5 columngutter=5
        rowDataRange=unionall columnDataRange=union panelNumber=_panelnumber_
        cellHeightMin=50px cellWidthMin=50px start=TopLeft columns=3 rows=3
        rowAxisOpts=( display=all altdisplay=all label="HbA1c (%)" type=auto
                      linearopts=( viewmin=_yviewmin_ viewmax=_yviewmax_ ) )
        columnAxisOpts=( display=all altdisplay=all label="Month" type=auto
                        linearopts=( tickvaluelist=( 0 3 6 9 12 ) viewmin=0 viewmax=12 ) );
    layout prototype / _SGPROC;
    SeriesPlot X=MONTH Y=HBA1C / primary=true display=(markers)
        Group=TRT INDEX=IDVAL Lineattrs=( Pattern=1 Thickness=3)
        LegendLabel="HbA1c" NAME="HBA1C";
    endlayout;
LegendLabel="HbA1c" NAME="HBA1C";
endgraph;
end;
```

Note: HbA1c dataset contains the following variables: SUBJID, TRT, BMI, MONTH, HBA1C and a derived variable IDVAL, which is based on the value of TRT as seen in “Code for Figure 3B (Using INDEX Option in Graph Template Language (GTL) Approach)".
CONCLUSION

In the analysis of clinical data, often times only a subset of data is of interest. Special attention has to be taken in order to guarantee that the line appearance for the treatments stay intact in subset analysis. In this paper, the author explores 5 different approaches to ensure line consistency. These methods include (1) adding dummy observations to input data, (2) re-mapping line attributes, (3) re-ordering of group variable internal values, (4) use of INDEX option in GTL and (5) use of Attribute Map in statistical graphics procedures. Except for the first method of adding dummy data, the remaining 4 approaches all achieve the same desirable results. So the choices should be based on the easiness of usage/maintenance and the SAS release currently available in your organization.

REFERENCES


ACKNOWLEDGMENTS

The author was inspired to explore on this topics after a brief conversation with Mr. Shawn Hopkins, of Seattle Genetics. Thank you so much, Mr. Hopkins, for sharing your insight and expertise. She is grateful to Ms. Susan Morrison, Ms. Lelia McConnell and Ms. Marcia Surratt of SAS Technical Support for their assistance. Last but definitely not least, she would like to extend her sincere gratitude to Ms. Justina Flavin and Mr. Art Carpenter for their SUGI 2004 paper and to Mr. Carpenter for his valuable feedback on this paper. Any errors and oversight in this article are the sole responsibility of the author.

CONTACT INFORMATION

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

Alice Monica Cheng
E-mail: alice_m_cheng@yahoo.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.