

Several Tips for Creating Graphs Using PROC SGPLOT & PROC TEMPLATE

Chunyan He, CARsgen Therapeutics

ABSTRACT

It always requires fancy outputs especially graphs in publication. In this paper, several SAS programming tips for making graphs more beautiful are introduced. For example, you created a linear plot using PROC SGPLOT for body temperature change over time for each subject at first. You're requested that adverse events of special interest occurred to each subject need to be marked in the plot in order to demonstrate the relationship of temperature change and adverse event occurrence. And concomitant medications of interest administrated to each subject need to be displayed in the graph to show the adverse event could be controlled by specified concomitant medications. It is difficult to still use PROC SGPLOT to create this plot in this case. We need to use PROC TEMPLATE instead to achieve all those requirements. Do you know how to make full use of PROC SGPLOT code you have and convert it to PROC TEMPLATE code quickly? How do you achieve all those requirements? This paper will give you answers.

INTRODUCTION

The SGPLOT procedure is one of Output Delivery System (ODS) graphics procedure, which can create one or more plots and overlays them on a single set of axes. Statements and options enable you to control the appearance of your graph and add additional features such as legends and reference lines. The SGPLOT procedure can create a wide variety of plot types, and can overlay plots together to produce many different types of graphs.

The SAS *Graph Template Language* (GTL) is an extension to the Output Delivery System (ODS) that enables you to create sophisticated graphics. The graphics produced by GTL are generated by template definitions that control the graph format and appearance and specify the variable roles to represent in the graph display. The graph can then be rendered by associating the templates with a data source.

We can use SGPLOT procedure to create most of graphs required for clinical summary report (CSR). However, for publication purpose, the CSR graphs might not be beautiful to better display the clinical analysis result. And we probably need to add some arrows and annotations, or customize the Y-axis label, etc. In this case it is difficult to still use SGPLOT procedure to achieve all those requests. Instead we need to use GTL to achieve all those requirements. It is kind of pity and waste of time to start writing GTL code from scratch. Could we make full use of SGPLOT code we already have and convert it to GTL code? The answer is of course yes. In this paper, the method how to convert SGPLOT code to GTL code quickly and several tips of using GTL to create fancy graphs are introduced by an example.

Here is the example dataset (Table 1) we are going to use for demonstration. We already created a line plot of body temperature change over time using PROC SGPLOT. Then we received request to mark adverse event (AE) of special interest and concomitant medications (CM) of interest administrated to each subject in the plot in order to demonstrate the relationship of temperature change and adverse event occurrence and show the adverse event could be controlled by specified concomitant medications. Based on request, we updated the datasets with AE information (Table 2).

Table 1 Original datasets (temp)

vsdy	temp
0	36.5
7	37
14	38
21	39

28	38
35	37.5
42	36.5
49	37
56	37

Table 2 Updated datasets with AE information

vsdy	temp	aedecod	eventday
0	36.5		.
1	.	Adverse event 1	1
5	.	Adverse event 2	5
7	37		.
14	38		.
15	.	Adverse event 3	15
21	39		.
28	38	Adverse event 4	28
35	37.5		.
42	36.5		.
49	37		.
56	37		.

CREATE GRAPH USING SGPLOT PROCEDURE

Below is our SGPLOT code we already have to create the line plot of temperature change over time.

```
proc sgplot data=temp;
    series x=vsdy y=temp/markers markerattrs=(symbol=circle size=5);
    xaxis label='Days after treatment';
    yaxis label='Temperature (C)';
run;
```

CONVERT SGPLOT CODE TO GTL CODE

We can add an option `tplout='t'` in PROC SGPLOT statement to easily convert SGPLOT code to GTL code and write GTL code to file t. Converted GTL code is as below,

```
proc template;
    define statgraph sgplot;
        begingraph / collation=binary;
        layout overlay / xaxisopts=( Label="Days after treatment"
            labelFitPolicy=Split type=linear )
            yaxisopts=( Label="Temperature (C)" type=linear );
        SeriesPlot X=vsdy Y=temp / primary=true display=(markers)
```

```

Markerattrs=( Symbol=CIRCLE Size=5);

endlayout;
endgraph;
end;
run;

```

CUSTOMIZE GRAPHS USING GTL CODE

STEP 1: ADD CONCOMITANT MEDICATIONS INFORMATION

We can use DRAWARROW and DRAWTEXT statement to add concomitant medications information.

```

**add arrows on the day of CM happens and specify CM name;

drawarrow x1=13 x2=18 y1=38.7 y2=38.6 /x1space=datavalue y1space=datavalue
x2space=datavalue y2space=datavalue arrowheadshape=filled
lineattrs=(color=red) ;

drawtext TEXTATTRS=(color=red family='arial' size=12) "CM Name 1" /
width=25 anchor=bottom x=13 y=38.7 xspace=datavalue yspace=datavalue;

drawarrow x1=30 x2=25 y1=38.6 y2=38.5 /x1space=datavalue y1space=datavalue
x2space=datavalue y2space=datavalue arrowheadshape=filled
lineattrs=(color=red) ;

drawtext TEXTATTRS=(color=red family='arial' size=12) "CM Name 2" /
width=25 anchor=bottom x=30 y=38.6 xspace=datavalue yspace=datavalue;

```

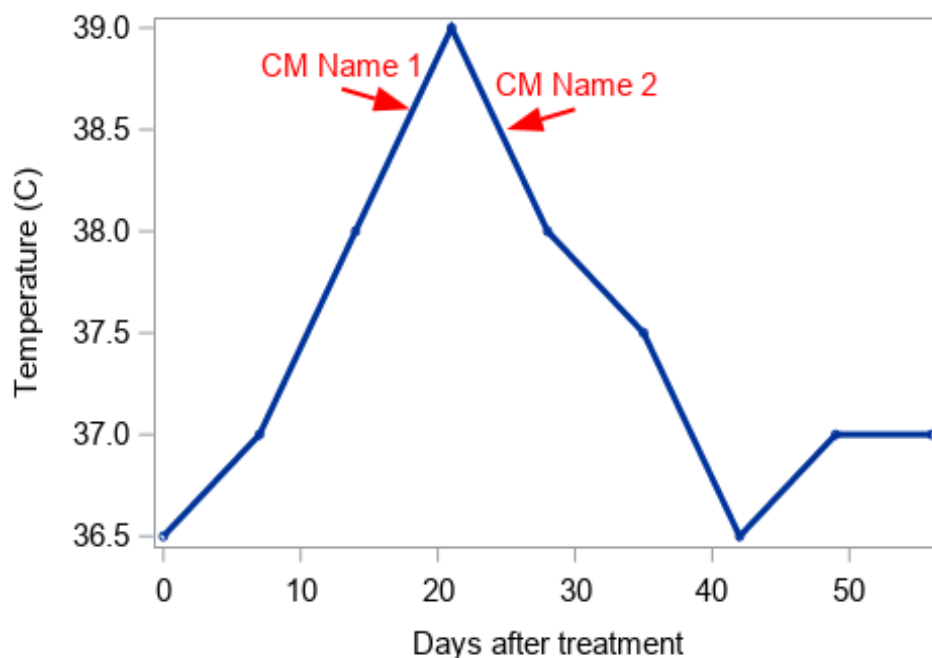


Figure 1. Figure with CM information

STEP 2: USE REFERENCELINE STATEMENT TO ADD ADVERSE EVENT INFORMATION

We can add REFERENCELINE statement to add AE information, SAS code is as below. However, there is an issue occurred that the adverse event label overlapped with X-axis tick value and X-axis label.

```
**add adverse event;  
referenceline x=eventday / curvelabel=aedecod curvelabellocation=outside  
curvelabelposition=min lineattrs=(color=green pattern=dot thickness=3) ;
```

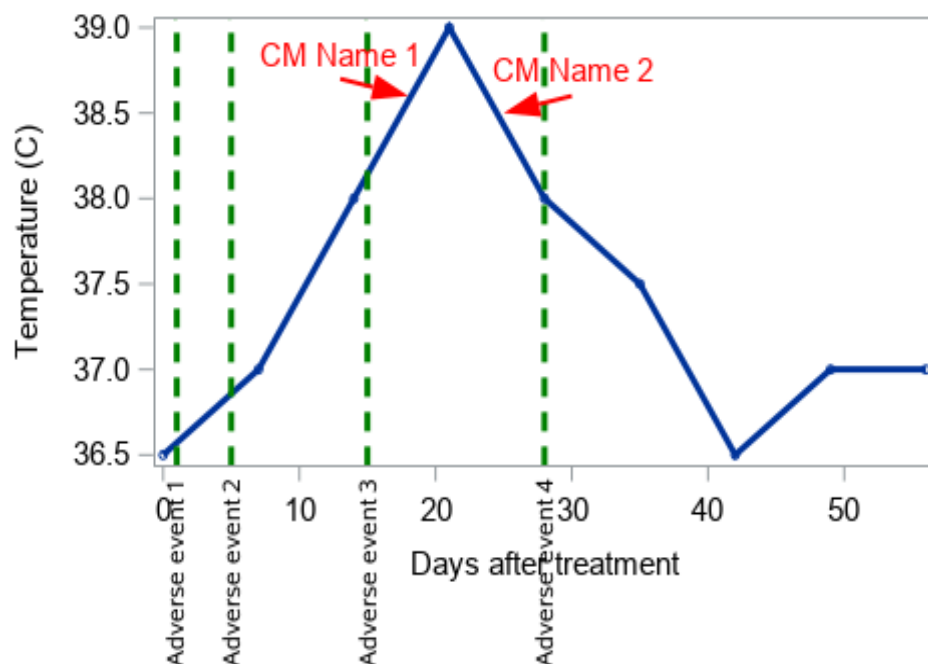


Figure 2. Figure with CM and AE information

STEP 3: USE DRAWTEXT STATEMENT TO DISPLAY ADVERSE EVENTS LABEL INSTEAD OF CURVELABEL OPTION IN REFERENCELINE STATEMENT

In order to fix above issue, we need to use DRAWTEXT statement to annotate adverse events. The DRAWTEXT statement can specify the AE label's location and label's angle of rotation which can make graph more beautiful. However, there is a new issue occurred that no sufficient space for adverse event labels under x-axis.

```
**adverse event annotation;  
drawtext textattrs=(color=green size=10) "Adverse event 1" /Y=10 x=-3  
xspace=datavalue yspace=graphpercent rotate=45 anchor=top width=50  
justify=center;  
drawtext textattrs=(color=green size=10) "Adverse event 2" /Y=10 x=0  
xspace=datavalue yspace=graphpercent rotate=45 anchor=top width=50  
justify=center;  
drawtext textattrs=(color=green size=10) "Adverse event 3" /Y=10 x=10
```

```
xspace=datavalue yspace=graphpercent rotate=45 anchor=top width=50
justify=center;

drawtext textattrs=(color=green size=10) "Adverse event 4" /Y=10 x=23
xspace=datavalue yspace=graphpercent rotate=45 anchor=top width=50
justify=center;
```

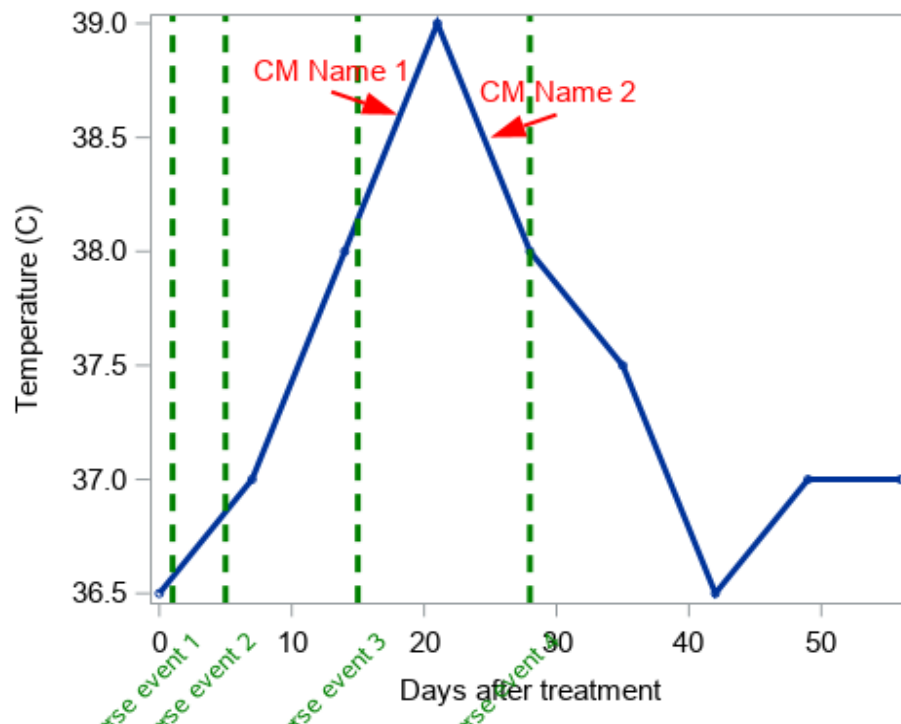


Figure 3. Updated Figure with CM and AE information

STEP 4: USE PAD OPTION TO ADD EXTRA SPACE AT THE BOTTOM OF THE GRAPH

We can use PAD option in LAYOUT OVERLAY statement to add extra space at the bottom of the graph to display adverse events' labels.

```
**add pad options to add extra space at the bottom of the graph;
layout overlay / pad=(bottom=15pct)
xaxisopts=( Label="Days after treatment"
labelFitPolicy=Split type=linear )
yaxisopts=( Label="Temperature (C)" type=linear );
```

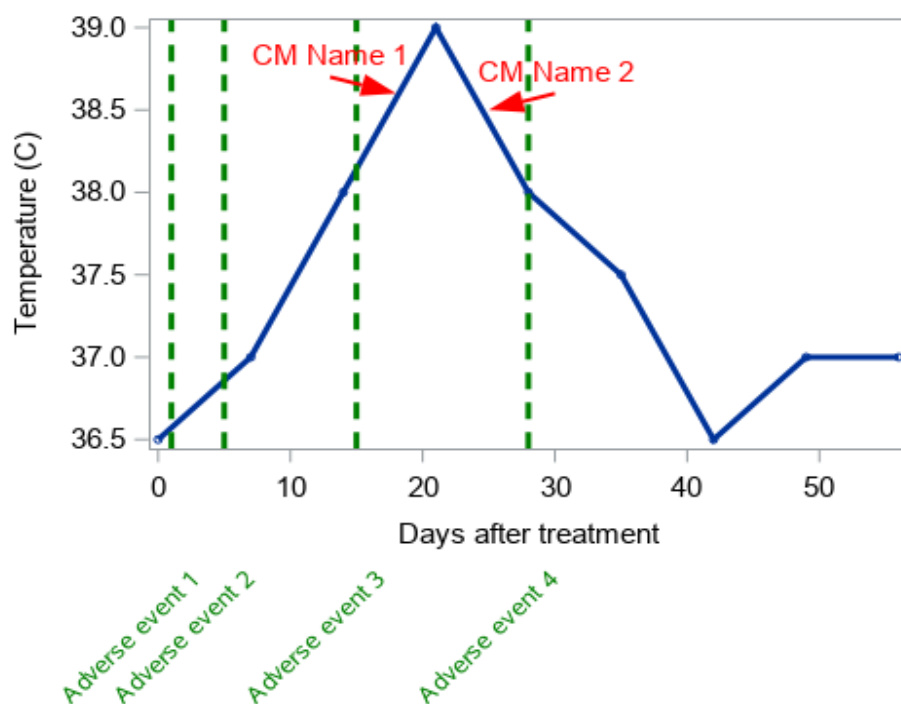


Figure 4. Final figure

CONCLUSION

When you have had SGPLOT code and want to customize your figure using GTL or when you're in dilemma that you, as a starter of GTL, have no idea how to start creating complicated graphs using GTL. You can make full use of SGPLOT code you already have or start creating graphs using SGPLOT procedure first, and then use TMPLOUT option to convert SGPLOT code to GTL code quickly. Then you can modify GTL code to further customize your figure. REFERENCELINE, DRAWARROW, DRAWTEXT statements and PAD option in GTL makes creating graph more flexible in our daily work.

REFERENCES

SAS® 9.4 Graph Template Language: Reference, Fifth Edition

<http://support.sas.com/documentation/cdl/en/grstatgraph/65377/HTML/default/viewer.htm#titlepage.htm>

SAS® 9.4 ODS Graphics: Procedures Guide, Sixth Edition

<https://documentation.sas.com/?docsetId=grstatproc&docsetTarget=titlepage.htm&docsetVersion=9.4&locale=zh-CN>

CONTACT INFORMATION

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

Name: Chunyan He

Enterprise: CARsgen Therapeutics

Work Phone: +86 21 64501828-8560

E-mail: chunyanhe@carsgen.com
Web: <http://www.carsgen.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. Any brand and product names are trademarks of their respective companies.