Creating Trellis Graph Using SAS®
Jane Liao, Merck & Co., Inc.

ABSTRACT

Data visualization is an important first step in getting to know the data before any formal data analysis. It is
sometimes desirable to combine plots from different data sets on the same graph. Multi-panel graphs allow one to
view relationships between different variables in the data set. This can be done by overriding the conditioning
specifications for a single trellis graph. Trellis graphs allow one to view relationships between different variables in
the data set through conditioning. By using trellis graphics, one can view the data in a series of panels where
each panel contains a subset of the original data divided into intervals of the conditioning variables. The trellis
graph provides a level of detail in a multi-dimensional context with clarity and graphical brevity. In this report,
different techniques using SAS will be explored in creating a trellis graph.

INTRODUCTION

It is a common practice to visualize your data before a formal data analysis is completed. There are many
different reasons for doing this. One important reason is to detect patterns in the data and view the relationship
between different variables of interest so that an appropriate statistical model and analysis can be conducted.
Other reasons include outlier detection and variable distribution assessment. Finally, as technology progresses
the volume and complexity of data collected is increasing at an exponential speed making data visualization an
even more critical step before beginning analyses. There are many ways to visualize your data such as looking at
basic frequency distributions, creating tabular listings, and graphics development. This paper will focus on trellis
graph development by examining five different SAS procedures, CORR, GREPLAY, SGSCATTER, G PLOT, and
SGPANEL.

METHODS

1. The CORR procedure

The SAS CORR procedure [1] computes correlation coefficients between variables and can also create scatter
plots. The PLOTS=MATRIX option in the PROC CORR statement uses ODS output graphics to display a scatter
plot matrix for variables listed in the VAR statement [2]. This is a nice way to view the relationship between
variables. Note, the graphics displayed with ODS are experimental in SAS version 9.1. Refer to Table 1 for the
main SAS code and Figure 1 for the resulting graphics.

| Table 1 |
| Code to Create Multiple Plots in One Page Using PROC CORR |
| ods graphics on; |
| proc corr data=dsplot noprint plots=matrix; |
| var var1 var2 var3 var4; |
| run; |
| quit; |
| ods graphics off; |
2. The GREPLAY procedure

The GREPLAY procedure [1] with the template function enables you to combine multiple graphs and place them in one page. There are several steps involved in using the GREPLAY procedure presented below [3].

- First, create each individual graph and store it in a graphics catalog.
- Second, create your own template (if a template is not already available).
- Third, replay the stored graphs and assign them to the sections as you designed.

First step: Create individual graphs using sample SAS code as displayed in Table 2 below.

Table 2 (First step for GREPLAY):
Code to Create Each Individual Plot Using PROC GPLOT

```
proc gplot data=dsplot;
  plot var1*var2 / anno=anno1 haxis=axis1 vaxis=axis2;
  plot var1*var3 / anno=anno2 haxis=axis1 vaxis=axis2;
  plot var1*var4 / anno=anno3 haxis=axis1 vaxis=axis2;
  plot var2*var3 / anno=anno4 haxis=axis1 vaxis=axis2;
  plot var2*var4 / anno=anno5 haxis=axis1 vaxis=axis2;
  plot var3*var4 / anno=anno6 haxis=axis1 vaxis=axis2;
run;
quit;
```

Second step: Create a template with the TDEF statement in PROC GREPLAY. In this example, six plots are displayed on one page. A template with two rows and three columns is designed as illustrated in Table 3. Corresponding SAS code is included in Table 4.
Table 3
Design your template

<table>
<thead>
<tr>
<th>Title (7)</th>
<th>ulx, uly</th>
<th>urx, ury</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (plot)</td>
<td>ulx, uly</td>
<td>urx, ury</td>
</tr>
<tr>
<td>llx, lly</td>
<td>lrx, lry</td>
<td></td>
</tr>
<tr>
<td>2 (plot)</td>
<td>ulx, uly</td>
<td>urx, ury</td>
</tr>
<tr>
<td>llx, lly</td>
<td>lrx, lry</td>
<td></td>
</tr>
<tr>
<td>3 (plot)</td>
<td>ulx, uly</td>
<td>urx, ury</td>
</tr>
<tr>
<td>llx, lly</td>
<td>lrx, lry</td>
<td></td>
</tr>
<tr>
<td>4 (plot)</td>
<td>ulx, uly</td>
<td>urx, ury</td>
</tr>
<tr>
<td>llx, lly</td>
<td>lrx, lry</td>
<td></td>
</tr>
<tr>
<td>5 (plot)</td>
<td>ulx, uly</td>
<td>urx, ury</td>
</tr>
<tr>
<td>llx, lly</td>
<td>lrx, lry</td>
<td></td>
</tr>
<tr>
<td>6 (plot)</td>
<td>ulx, uly</td>
<td>urx, ury</td>
</tr>
<tr>
<td>llx, lly</td>
<td>lrx, lry</td>
<td></td>
</tr>
</tbody>
</table>

Table 4 (Second step for GREPLAY)
Code to Create Template for PROC GREPLAY

```syntax
proc greplay igout=work.gseg tc=mytemp nofs;
tdef t2r3c
1/llx=0 lly=48 ulx=0 uly=95 lrx=33 lry=48 urx=33 ury=95
2/llx=33 lly=48 ulx=33 uly=95 lrx=67 lry=48 urx=67 ury=95
3/llx=67 lly=48 ulx=67 uly=95 lrx=100 lry=48 urx=100 ury=95
4/llx=0 lly=0 ulx=0 uly=48 lrx=33 lry=0 urx=33 ury=48
5/llx=33 lly=0 ulx=33 uly=48 lrx=67 lry=0 urx=67 ury=48
6/llx=67 lly=0 ulx=67 uly=48 lrx=100 lry=0 urx=100 ury=48
7/llx=0 lly=0 ulx=0 uly=100 lrx=100 lry=0 urx=100 ury=100
;template t2r3c;
```

Third step: Replay the stored graphs:
```
TREPLAY 1:gplot 2:gplot1 3:gplot2 4:gplot3 5:gplot4 6:gplot5 7:title;
```

The final plot created using PROC GREPLAY is shown in Figure 2.

Figure 2
Multiple Plot in One Page Using PROC GREPLAY
Tips when using GREPLAY procedure [4]:

- Clear the graph catalog to ensure the graphs have the names as you expected.

```
Table 4
Code to Clear Graph Catalog

proc greplay igout=work.gseg nofs;
delete _all_;
run;
quit;
```

- View the available SAS defined templates to see what is automatically made available to you. This can be done by running the code included in Table 5 below. The available templates will then be displayed in your SAS log window – refer to Table 6. If you don't see a template you desire, then a new template will need to be created.

```
Table 5
Code to View Template in SAS

proc greplay nofs
   tc=sashelp.templt;
list tc;
run;
quit;
```

```
Table 6
Template from SASHELP.TEMPLT

NOTE: Catalog members: SASHELP.TEMPLT

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>H2</td>
<td>1 BOX LEFT, 1 BOX RIGHT</td>
</tr>
<tr>
<td>H2S</td>
<td>1 BOX LEFT, 1 BOX RIGHT (WITH SPACE)</td>
</tr>
<tr>
<td>H3</td>
<td>3 BOXES ACROSS (HORIZONTALLY)</td>
</tr>
<tr>
<td>H3S</td>
<td>3 BOXES ACROSS (WITH SPACE)</td>
</tr>
<tr>
<td>H4</td>
<td>4 BOXES ACROSS (HORIZONTALLY)</td>
</tr>
<tr>
<td>H4S</td>
<td>4 BOXES ACROSS (WITH SPACE)</td>
</tr>
<tr>
<td>L1R2</td>
<td>1 BOX LEFT, 2 BOXES RIGHT</td>
</tr>
<tr>
<td>L1R2S</td>
<td>1 BOX LEFT, 2 BOXES RIGHT (WITH SPACE)</td>
</tr>
<tr>
<td>L2R1</td>
<td>2 BOXES LEFT, 1 BOX RIGHT</td>
</tr>
<tr>
<td>L2R1S</td>
<td>2 BOXES LEFT, 1 BOX RIGHT (WITH SPACE)</td>
</tr>
<tr>
<td>L2R2</td>
<td>2 BOXES LEFT, 2 BOXES RIGHT</td>
</tr>
<tr>
<td>L2R2S</td>
<td>2 BOXES LEFT, 2 BOXES RIGHT (WITH SPACE)</td>
</tr>
<tr>
<td>U1D2</td>
<td>1 BOX UP, 2 BOXES DOWN</td>
</tr>
<tr>
<td>U1D2S</td>
<td>1 BOX UP, 2 BOXES DOWN (WITH SPACE)</td>
</tr>
<tr>
<td>U2D1</td>
<td>2 BOXES UP, 1 BOX DOWN</td>
</tr>
<tr>
<td>U2D1S</td>
<td>2 BOXES UP, 1 BOX DOWN (WITH SPACE)</td>
</tr>
<tr>
<td>V2</td>
<td>1 BOX UP, 1 BOX DOWN</td>
</tr>
<tr>
<td>V2S</td>
<td>1 BOX UP, 1 BOX DOWN (WITH SPACE)</td>
</tr>
<tr>
<td>V3</td>
<td>3 BOXES STACKED VERTICALLY</td>
</tr>
<tr>
<td>V3S</td>
<td>3 BOXES STACKED VERTICALLY (WITH SPACE)</td>
</tr>
<tr>
<td>WHOLE</td>
<td>ENTIRE SCREEN TEMPLATE</td>
</tr>
</tbody>
</table>
3. **SGSCATTER procedure in SAS 9.2**

The **SGSCATTER** procedure in **SAS 9.2** [1] creates a paneled graph of scatter plots for multiple combinations of variables, depending on the plot statement that you used. You can use options to overlay fit plots and ellipses on your scatter plots. This functionality can be implemented with the following three statements [5]:

- The **MATRIX** statement, which creates a scatter plot matrix, and has options for computed ellipses and diagonal plots.
- The **COMPARE** statement, which creates a shared axis panel of scatter plots, and has options for fits and computed ellipses.
- The **PLOT** statement, which creates a panel of independent scatter plots, and has options for fits and computed ellipses.

Three examples are included below: Table 7a utilizes the **MATRIX** statement and the resulting graph (Figure 3) is similar to the plot from **PROC CORR**. The second example uses the **COMPARE** statement (refer to Table 7b) and the last example uses the **PLOT** statement (refer to Table 7c). The resulting graphs are similar from both examples (see Figure 4).

<table>
<thead>
<tr>
<th>Table 7a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code to Create Multiple Plot Using <strong>PROC SGSCATTER</strong></td>
</tr>
<tr>
<td><strong>proc sgscatter</strong> data=dsplot;</td>
</tr>
<tr>
<td><strong>matrix</strong> var1 var2 var3 var4;</td>
</tr>
<tr>
<td><strong>run</strong>;</td>
</tr>
<tr>
<td><strong>quit</strong>;</td>
</tr>
</tbody>
</table>

**Figure 3 (Graph from Table 7a)**

Trellis Scatterplot from **PROC SGSCATTER** (V9.2 only)

VAR1

VAR2

VAR3

VAR4
Table 7b
Code to Create Multiple Plot Using PROC SGSCATTER

```sas
proc sgscatter data=dsplot;
  compare x=var1 y=(var2 var3 var4);
run;
quit;
```

Figure 4 (Graph from table 7b and 7c)

Table 7c
Code to Create Multiple Plot Using PROC SGSCATTER (The plot is very similar to Figure 4)

```sas
proc sgscatter data=dsplot;
  plot var1*(var2 var3 var4) /column=1;
run;
quit;
```

4. GPLOT procedure

The GPLOT SAS procedure can be used to create a graph with multiple panels. However data manipulation is required before calling the procedure. This paper presents one example to create the Mean Plot with Error Bar for multiple panels by treatment (e.g., one panel for each treatment group as showing in Figure 5). In order to create this graph, data manipulation steps are needed and they are shown in Table 8a and 8b.

<table>
<thead>
<tr>
<th>TRT</th>
<th>parm</th>
<th>newvar_Mean</th>
<th>newvar_StdErr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>38.37</td>
<td>3.05</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>60.90</td>
<td>2.86</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>52.70</td>
<td>2.08</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>32.02</td>
<td>3.49</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>43.14</td>
<td>2.44</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>61.27</td>
<td>2.34</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>50.71</td>
<td>2.22</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>40.08</td>
<td>3.46</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>47.18</td>
<td>2.81</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>61.22</td>
<td>2.47</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>52.99</td>
<td>2.03</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>43.99</td>
<td>4.06</td>
</tr>
</tbody>
</table>
A proc print of data from Table 8b
(Only show treatment 0 and 1)

<table>
<thead>
<tr>
<th>TRT</th>
<th>parm</th>
<th>xvar</th>
<th>plotvar</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>35.32</td>
<td>Low</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>38.37</td>
<td>Mean</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>41.42</td>
<td>High</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
<td>58.04</td>
<td>Low</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
<td>60.90</td>
<td>Mean</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
<td>63.77</td>
<td>High</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>3</td>
<td>50.62</td>
<td>Low</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>3</td>
<td>52.70</td>
<td>Mean</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>3</td>
<td>54.78</td>
<td>High</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>4</td>
<td>28.53</td>
<td>Low</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>4</td>
<td>32.02</td>
<td>Mean</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>4</td>
<td>35.51</td>
<td>High</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>40.70</td>
<td>Low</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>43.14</td>
<td>Mean</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>45.58</td>
<td>High</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>6</td>
<td>58.93</td>
<td>Low</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>6</td>
<td>61.27</td>
<td>Mean</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>6</td>
<td>63.61</td>
<td>High</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>48.49</td>
<td>Low</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>50.71</td>
<td>Mean</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>52.92</td>
<td>High</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>8</td>
<td>40.08</td>
<td>Mean</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>8</td>
<td>43.54</td>
<td>High</td>
</tr>
</tbody>
</table>

Table 8b: Code to get mean and high/low value for error bar

```sas
data mstd;
  set anal;
  length label $4;
  plotvar=newvar_Mean-newvar_StdErr;
  label='Low'; output;
  plotvar=newvar_mean;
  label='Mean'; output;
  plotvar=newvar_Mean+newvar_StdErr;
  label='High'; output;
run;
```

In order to have the graph with three panels instead of one panel, a new variable XVAR is created in the next step.

```sas
data mstd ;
  set mstd;
  xvar=4*trt+parm;
run;
```

Figure 5
Multiple Plot in One Page Using PROC GGPLOT with Manipulated Data

Mean Plot with Error Bar
5. **SGPANEL procedure in SAS 9.2**

The SGPANEL procedure in version SAS 9.2 [1] is an enhanced graphic procedure compared to previous SAS versions. The SGPANEL procedure is designed to produce paneled graphs based on conditioning variables. The SGPANEL procedure allows the user to combine a variety of plot and chart types into more elegant overlays with a lot of flexibility. An example is shown in Table 9. The SGPANEL procedure supports the following plot types [5]:

- Basic plot: scatter, series, step, band, and needle
- Fits and Confidence: loess, regression, and penalized B-spline
- Distribution: horizontal and vertical box plots, histograms, normal curves, and kernel density estimates
- Categorization: dot plots, horizontal and vertical bar charts, and horizontal and vertical line charts

<table>
<thead>
<tr>
<th>Table 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code to Create Multiple Plot Using PROC SGPANEL</td>
</tr>
</tbody>
</table>

```sas
proc sgpanel data=sg;
panelby trt /columns=3 rows=1;
label trt='Treatment Group';
scatter x=parm y=newvar_mean / YERRORLOWER=lowv yerrorupper=highv;
COLAXIS label='' values=1 to 4 by 1);
rowaxis label='Mean of Value' values=(20 to 70 by 10);
format parm tcsg.;
run;
quit;
```

<table>
<thead>
<tr>
<th>Figure 6 (Graph from Table 9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple Plot in One Page Using PROC SGPANEL</td>
</tr>
</tbody>
</table>

![Graph showing mean plots with error bars for different treatment groups.](image-url)

Mean Plot with Error Bar done by SGPANEL procedure (SAS Version 9.2)

<table>
<thead>
<tr>
<th>Treatment Group = 0</th>
<th>Treatment Group = 1</th>
<th>Treatment Group = 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAR1</td>
<td>VAR2</td>
<td>VAR3</td>
</tr>
<tr>
<td>VAR4</td>
<td>VAR1</td>
<td>VAR2</td>
</tr>
<tr>
<td>VAR3</td>
<td>VAR4</td>
<td>VAR1</td>
</tr>
</tbody>
</table>
CONCLUSION

This paper presented five SAS procedures to aid your data visualization techniques. SAS version 9.2 includes new procedures to make the creation of trellis graphs easier compared to previous SAS versions. Through the use of PROC CORR, GREPLAY, SGSCATTER, G PLOT, and SGPANEL you can achieve the creation of trellis graphics.

REFERENCES


TRADEMARK INFORMATION

SAS, SAS/GRAPH and SAS/STAT software are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.

ACKNOWLEDGMENTS

I would like to thank Amy Gillespie from Merck & Co., Inc. encouraging me to materialize this paper and reviewing the paper and providing me valuable feedback.

CONTACT INFORMATION

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

Author Name: Jane Liao
Company: Merck & Co., Inc.
Address: 351 North Sumneytown Pike
City, State ZIP  North Wales, PA 19454-1099
Work Phone: 267-305-7514
Fax:
Email: jane_liu2@merck.com