

Building a Nested Range Bar Chart with SAS® Graph Template Language

Fan Pan, University of South Carolina; Wenjia Wang, University of South Carolina;
Jin Liu, University of South Carolina

ABSTRACT

PROC GCHART is an essential procedure within the SAS/GRAPH software to create bar charts. Besides the basic bar chart, GCHART is also able to produce grouped bar charts that visualize the comparison of values by categories. However, this procedure has a limitation regarding display of the variable with a range of values. PROC TEMPLATE, as a relative new statement that format tables and create different graphs, could be used to resolve the limitation of GCHART concerning the ranged values.

INTRODUCTION

A bar chart is a graphic construct with vertical or horizontal rectangular bars that present values with the length by proportion. Bar graphs are commonly chosen to display the comparison of measured values by discrete categories. Range bar chart has the advantages of depicting data with its beginning value rather than starting at a common zero point, and grouped bar chart enables the better visualization of the comparison across variables. This paper will present the choice of procedures that efficiently produce an innovative nested grouped bar chart, which combines the grouped bar chart and range bar chart.

RATIONALE FOR THE NESTED RANGE BAR GRAPH

When comparing the values across variables, the method of grouping bars is commonly used. The traditional ways of grouping bars are stacking and clustering. However, the grouped bar graphs could not represent a variable with a range of value. The nested range chart (generated by PROC TEMPLATE) on the right has more competency to show the max and min values of two or more variables. The purple bar represents one variable, and the pink bar represents another. Each bar displays the min value by the bottom side and max value at the top. Overlapping bars show the magnitude of the difference between the Max 1 and Max 2 or Min 1 and Min 2, whereas a “stacked” bar generated by PROC GCHART merely stacking the variables without distinguishing the two variables with the width of bars. Thus, the stacked range bar graph confuses the audience because it does not display the smallest and largest value within each variable. The code that generated the stacked range chart came from the Robert Allison's SAS code that created the single floating bar chart.

Figure 1

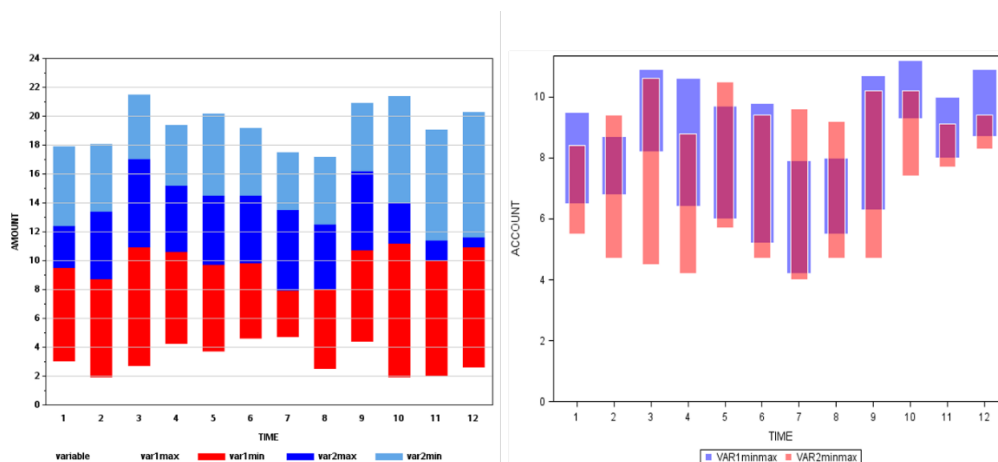


Figure 1. Visual Advantages of the Nested Range Bar Chart

SIMPLICITY OF PROGRAMMING

PROC TEMPLATE has more concise code for the data step. The data step below converted two variables to 4 columns Var1Min, Var1Max, Var2Min, and Var2Max with their min and max value:

```
data my_data;
input TIME VAR1min VAR1max VAR2min VAR2max;
datalines;
...
run;
```

However, GCHART statement cannot make a direct comparison of the values. Thus a new variable “variable” is created to define the lower and upper edges of the ranges. It takes an additional data step with do loop to formatting the borders of values.

```
data my_data;
input TIME VAR1min VAR1max VAR2min VAR2max;
variable='var1min'; amount=VAR1min; output;
variable='var1max'; amount=VAR1max-VAR1min; output;
variable='var2min'; amount=VAR2min; output;
variable='var2max'; amount=VAR2max-VAR2min; output;
datalines;
...
run;
data my_data; set my_data;
length htmlvar $500;
if variable eq 'var1max' then do;
htmlvar=
'title=||quote(
'Time: '||trim(left(Time))||'0D'x||
'Max: '||trim(left(VAR1max))||'0D'x||
'Min: '||trim(left(VAR1min))
)||
' href="bar7_info.htm"';
end;
if variable eq 'var2max' then do;
htmlvar=
...
end;
run;
```

However, the complexity of preparation for the GCHART statement still could not produce ideal output. See the GCHART output in the left graph of Figure 1. Besides the conciseness, code of the PROC TEMPLATE is also much easy to change if the users want to generate other similar bar charts. See Appendix (on page 4) for the complete PROC TEMPLATE code.

CONCLUSION

PROC TEMPLATE is gaining popularity concerning making graphs. The paper demonstrated two advantages of using PROC TEMPLATE. First, the TEMPLATE statement is better at visualizing comparison across different variables than the GCHART statement. Second, the TEMPLATE statement is more efficient for data preparation as well as editing the code. Therefore, it will be easier for people who are not familiar with SAS language to apply.

REFERENCES

Haworth, Lauren. 2006. "PROC TEMPLATE: The Basics." Proceedings of the 31st Annual SAS Users Group Conference, South San Francisco, CA. Available at <http://www2.sas.com/proceedings/sugi31/112-31.pdf>.

Matange, Sanjay. "Bar Charts with Stacked and Cluster Groups." © SAS Institute Inc. 2013. Available at <https://blogs.sas.com/content/graphicallyspeaking/2013/09/07/bar-charts-with-stacked-and-cluster-groups/>.

Watts, Perry. 2013. "Building a Better Bar Chart with SAS® Graph Template Language." Proceedings of the 2013 Midwest SAS Users Group Conference. Elkins Park, PA. Available at <https://www.mwsug.org/proceedings/2013/HW/MWSUG-2013-HW05.pdf>

CONTACT INFORMATION

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

Fan Pan
University of South Carolina
Pan6@email.sc.edu

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.

APPENDIX: COMPLETE CODE FOR THE PROC TEMPLATE

```
/*SESUG sas code creating the two range bar chart with the borderline of the
white chart*/
proc template;
  define statgraph MYBARCHART;
    begingraph/datacontrastcolors=(white);
    layout overlay /
      CYCLEATTRS=TRUE
      xaxisopts=(label="TIME")
      yaxisopts=(label="ACCOUNT");

    BARCHART Y=VAR1max X=TIME /
      stat=SUM NAME="VAR1minmax" LEGENDLABEL="VAR1minmax"
      barwidth=0.65 barlabel=FALSE
      fillattrs=(color=blue TRANSPARENCY=0.5);

    BARCHART Y=VAR1min X=TIME /
      stat=SUM
      barwidth=0.65 barlabel=FALSE
      fillattrs=( color=white TRANSPARENCY=0);

    BARCHART Y=VAR2max X=TIME/
      stat=sum NAME="VAR2minmax" LEGENDLABEL="VAR2minmax"
      barwidth=0.45 barlabel=FALSE
      fillattrs=(color=red TRANSPARENCY=0.5);

    BARCHART Y=VAR2min X=TIME /
      stat=sum
      barwidth=0.45 barlabel=FALSE
      fillattrs=(color=white TRANSPARENCY=0);
      DISCRETELEGEND "VAR1minmax" "VAR2minmax";
    endlayout;
    endgraph;
  end;
run;
ods listing style=listing;
proc sgrender data=my_data template=MYBARCHART;
run;
```