

Using SAS/IntrNet™ to Teach PROC GCHART Coding

Wing K Chan, The Pennsylvania State University, University Park, Pennsylvania
David C. Steven, The Pennsylvania State University, University Park, Pennsylvania

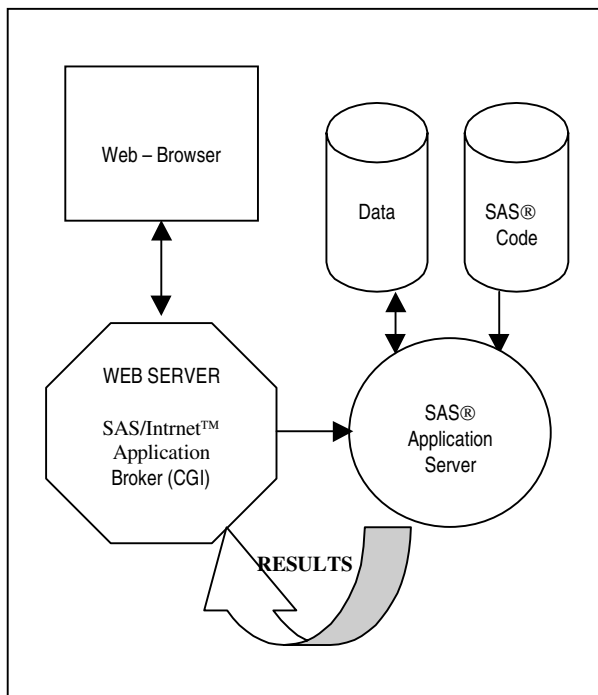
ABSTRACT

Teaching information technology poses a challenge for both university professors and corporate trainers. Time and/or resource constraints can impede learning opportunities for students. Through web-based learning, students can work independently outside the classroom, yet maintain a relatively high degree of interaction or feedback. In this paper, we present a web-based interactive learning application employing SAS/IntrNet™ to enable students to learn PROC GCHART fundamentals using just a web browser on a networked machine. A student using this program is able to experiment with sophisticated SAS codes simply by “pointing and clicking.”

INTRODUCTION

This application employs the SAS/IntrNet Application Dispatcher. The Application Dispatcher allows input from HTML forms to be passed to a SAS Application Server via a CGI broker program developed by SAS Institute. The Application Server receives parameters from the HTML form, including the specification of which SAS program to run. It then processes the request, and returns the results to the web browser. Figure 1 depicts the flow of communication in a typical SAS/IntrNet™ Application Dispatcher application.

Figure 1.



We have created an interactive web-based environment in which a student learning PROC GCHART can select options to build a customized graph. Students receive immediate results from the SAS code they build from selections on an HTML form page, including the completed SAS code and the PROC GCHART output. The application enables students to experiment with PROC

GCHART statements and observe how different statements affect the results without being distracted by the overhead resulting from making typing mistakes or syntax errors. Figure 2 displays a web page generated by this application.

DEMO APPLICATIONS AND CODE

Live working demonstrations and complete SAS code for this and similar SAS/IntrNet™ applications are available for public access from:

<http://cac.psu.edu/beatnic/papers/sugi/>

PROC GCHART

PROC GCHART can produce a variety of charts, including vertical and horizontal bar charts, block charts, pie charts, and star charts. These charts may be specified for frequency counts, cumulative frequency counts, percentages, cumulative percentages, sums, and means. The variety of graphs that can be generated is virtually limitless. Here, we concentrate only on two types of charts: the vertical and horizontal bar charts, and on just three statistics: means, sums, and frequency counts.

PROC GCHART options available for modifying chart appearance number in the dozens. Some options change a graph's visual characteristics, such as the *percent* and *slice* statements. Some alter the statistics displayed, for example, the *sumvar* statement. Others subset graphs into groups, such as the *groupvar* statement. Students learning the GCHART procedure may become overwhelmed by all these options. Using this application, a student may experiment with PROC GCHART options from any web-capable computer and generate valid SAS code with output to learn how the procedure works when supplied different valid options. The application guides the student along by restricting the procedure options and syntax to valid cases only.

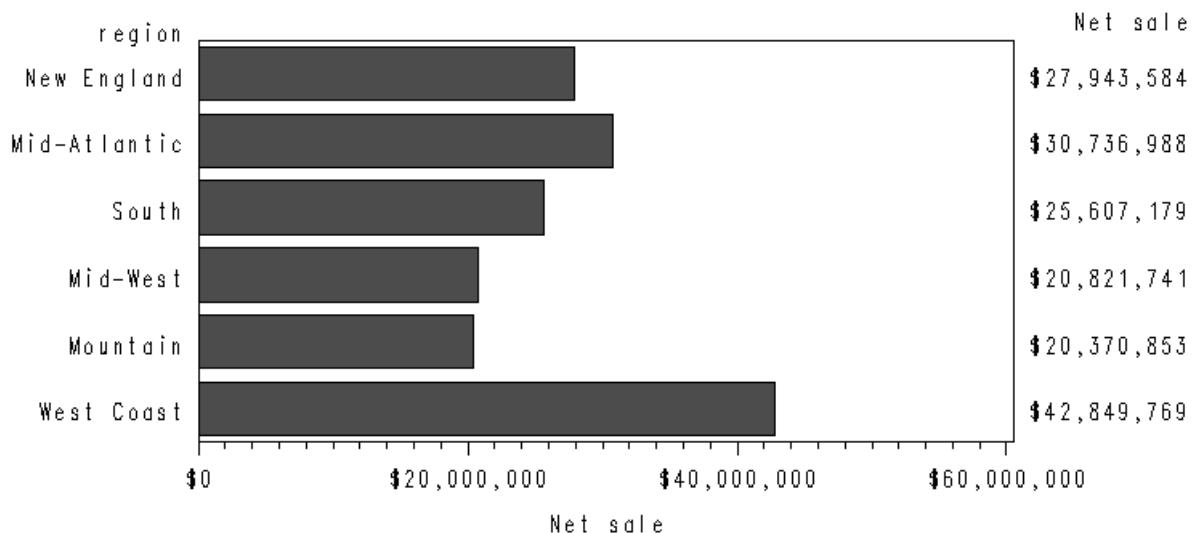
USING THE APPLICATION

The student is presented with an introductory page upon “arrival” at the PROC GCHART tutorial web site. On this page are instructions and links to the source code for the SAS/IntrNet™ application, along with a link to the data set used in the tutorial examples. A button labeled “GO” near the bottom of the introductory page will send a request to the SAS Application Dispatcher to return a page containing an initial PROC GCHART code sample and output. An example of this page, after selecting “Bar Charts” from the introductory page, is reproduced in Figure 2.

PROC GCHART statements are manipulated using selections from the HTML form on the right side of the page. If “Bar Charts” is selected from the introductory page, the student may choose from horizontal bar charts (HBAR) or vertical bar charts (VBAR). Otherwise, the PIE statement is presented for generating pie charts. To the right of the HBAR/VBAR statement selection list, the student may select from a list of analysis variables. Below that, s/he may decide whether to include a sumvar statement. If so, an additional variable and a summary statistic must be selected. To complete the process of building a PROC GCHART request, the student may then opt to subgroup the graph using either the GROUP or SUBGROUP statement. Options specific to generating pie charts are available for the case when “Pie Charts” is selected from the introductory page. When the student is satisfied with his or her

Figure 2.

<p>Current Codes:</p> <pre> data electronic; infile 'electronic_data.dat'; input id \$ region quarter gsale return discount nsale; run; proc gchart data=electronic; format nsale dollar12. region region. Id \$product.; hbar region / sumvar=nsale discrete sum; run; quit; </pre>	<pre> proc gchart data=electronic; format nsale dollar12. region region. id \$product.; </pre> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <input type="text" value="vbar"/> </div> <div style="text-align: center;"> <input type="text" value="quarter"/> </div> <div style="text-align: center;">/</div> </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <input type="text" value="sumvar"/> </div> <div style="text-align: center;">=</div> <div style="text-align: center;"> <input type="text" value="gross sale"/> </div> <div style="text-align: center;"> <input type="text" value="sum"/> </div> </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <input type="text" value="group"/> </div> <div style="text-align: center;">=</div> <div style="text-align: center;"> <input type="text" value="region"/> </div> </div> <pre> run; quit; </pre> <div style="text-align: center; margin-top: 10px;"> <input type="button" value="submit"/> </div>
---	--



selections, s/he selects the SUBMIT button, and new results are generated.

SAS/IntrNet™ returns the appropriate PROC GCHART code on the left, based on the selections, a new form on the right for making new selections, and the resultant graph. This process can be repeated with different options as desired.

PROGRAM DESIGN AND LOGIC

In designing an application to meet this need, we focused on continuity and clarity. Continuity meant that students using the system should be able to retrace what s/he has done to generate a given graph without being distracted or sidelined by other potential problems, such as typographical errors or difficulty understanding the syntax for a given option or statement. To this end, the codes that the student "wrote" by clicking on the pull-down menus are stored and replayed along with the graph. The student may focus on the effect of different options, and may observe the correct syntax for any specific statement or option selected. This is made possible by the SAS Macro facility, and the put statements in the SAS program.

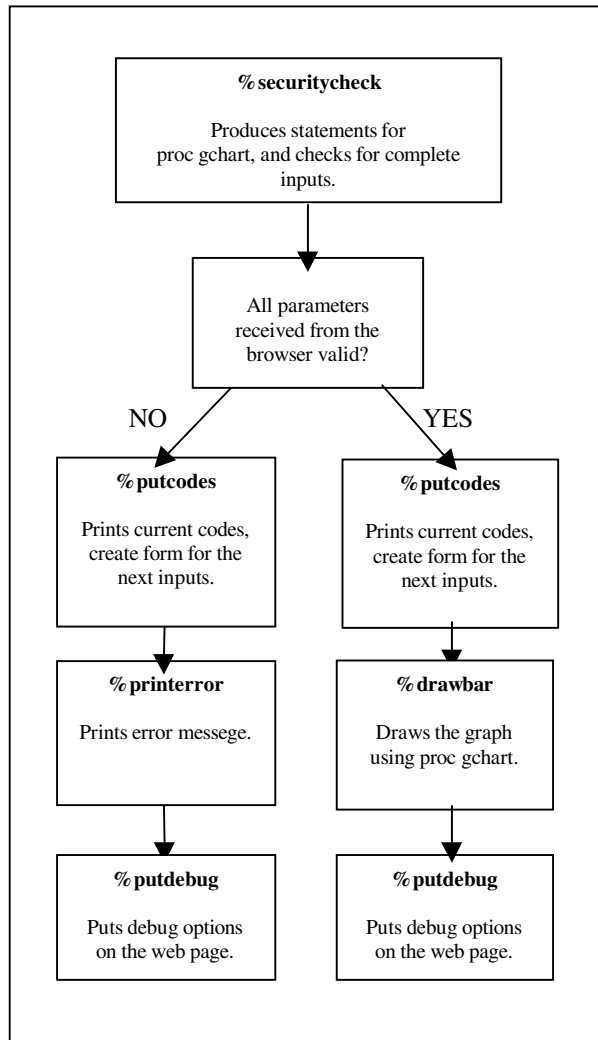
Clarity was achieved through the design of a user interface that is compact, easy to manage using minimal effort, and that generates results that are easy to find and to compare with the code that generated them. In this application, the code that generated the currently displayed graph are placed on the top left corner of the screen. Placed next to that is the HTML form for the making new coding selections. The HTML form mimics the PROC GCHART coding syntax. Some statements are editable, and others are permanent. Coding selection is performed through a pull-down menu. This eliminates typing and syntax and ensures that the Application Server is sent only acceptable, pre-defined selection parameters. Next, the graph generated by PROC GChart is placed in the middle of the screen, below the code and HTML form. Finally, debug options are made available as a selection near the bottom of the web page. This enables the user to view the SAS Log if s/he wishes to.

ANALYSIS OF THE SAS/INTRNET™ APPLICATION

The bulk of the application is comprised of three macros: %putcodes, %drawbar, and %securitycheck. There are other, minor macro functions that play a role in the final product. Additional coding is necessary for the complete SAS/IntrNet™ application.

Code for the macros is included in the Appendix. Figure 3 depicts the interrelation of these three macros.

Figure 3.



THE %SECURITYCHECK MACRO

When the application is initialized, or after the student selects the SUBMIT button, the parameters from the HTML form are sent to the SAS/IntrNet™ server. The server calls the designated program to handle the inputs. In this case, the %securitycheck macro is called upon first to validate the data, check for their completeness, and start generating SAS® statements in the form of macro variables. In this macro, we use the %let statement, and the logic statements %if ... %then ... %else extensively. There are two main options for the %securitycheck macro to take: (1) If all the input are valid and complete, then initiate the drawing process and reload the input form. However, if anything is wrong with the input, then (2) print an error message and print the input form for the user to enter again. Selected codes from this macro are detailed in the Appendix.

THE %PUTCODES MACRO

Although it is not on top of the logic flow diagram, the macro %putcodes in this application actually initiates the interaction between the program and the user. In this macro, the current codes are printed out. In addition, the form for the next round of HTML

inputs is created. This is done primarily through the use of the “put” statement. The put statement is a powerful tool in this case. As these statements are processed by the Application Dispatcher, they are converted into regular HTML statements that are understandable by web browsers. This adds tremendous flexibility, and enables the programmer to design customized web pages. Selected codes for the %putcodes macro are detailed in the Appendix.

THE %DRAWBAR MACRO

Assuming the student’s input clears the security check macro, this macro will be called to draw the charts using PROC GCHART. The %drawbar macro has two main parts. First, there are some ODS statements that opens and closes the Output Delivery System. Second is PROC GCHART with its statements options, which comprise the bulk of the %drawbar macro. These options are specified by the user, using the pull down menus in the HTML form shown in Figure 2. Parameters input from the HTML form are processed and passed onto the PROC GCHART statements using macro variables defined in the %securitycheck macro. Complete codes for the %drawbar macro, is detailed in the Appendix.

CONCLUSION

We have built a web-enabled SAS/IntrNet™ application that enables interactive self-taught learning of PROC GCHART. The application reduces frustration by focusing on the effect of PROC GCHART statements and options on the resulting output, and eliminating distractions due to typing mistakes and misunderstanding syntax requirements. This application is extensible to any of a number of other SAS® procedures and coding steps.

CONTACT INFORMATION

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

David C. Steven
 Graduate Education and Research Services
 The Center for Academic Computing
 The Pennsylvania State University
 University Park, PA 16802
 +1 814 865 0980
 +1 814 863 7049
 dcsteven@psu.edu
<http://cac.psu.edu/gears/>

APPENDIX

%SECURITYCHECK MACRO CODE SAMPLE

```

%macro securitycheck;
%let barchoice=%superq(barchoice);
%let barvar=%superq(barvar);
...
...
%if (&barvar=region or &barvar=quarter) %then %let discrete_statement= discrete;
%if &barvar=id %then %let discrete_statement= ;
%if &grouping=group %then %let groupstatement= group=&groupvar;
%if &grouping=subgroup %then %let groupstatement= subgroup=&groupvar;
%if &grouping=pick %then %let groupstatement= ;
...
...

%if (&barchoice=pick or %length(&barvar)=0 or %length(&sumvar)=0 or &sumvaraction=pick) %then
%do;
  %putcodes;
  %printerror;
  %putdebug;
%end;
%else %do;
  %if ( (&barchoice=hbar or &barchoice=vbar) and
    (&barvar=id or &barvar=region or &barvar=quarter) and
    (&sumvar=return or &sumvar=discount or &sumvar=gsale or &sumvar=nsale)) %then %do;
    %if( (&grouping=group or &grouping=subgroup) and
      (&groupvar^=id and &groupvar^=region and &groupvar^=quarter)) %then %do;
      %putcodes;
      %printerror;
      %putdebug;
    %end;
  %else %do;
    %putcodes;
    %drawbar;
    %putdebug;
  %end;
  ...
  ...
%end;
%end;

```

The %superq() function extracts the input from the HTML form.

The %if statement compares the user inputs to the graphing requirements. If a PROC GCHART statement is necessary, the %let statement puts that into a macro variable for later use.

This is the main decision making section of the %securitycheck function. It determines whether the inputs are valid and complete. If so, it calls the %drawbar function to draw the graph, along with the %putcodes and %debug. If the inputs are not valid or incomplete, this function will still call the %putcodes and %debug function. However, it will call the %printerror function to let the user know that he/she has made a mistake.

%PUTCODES MACRO CODE SAMPLE

```

%macro putcodes;
ods html body=_webout (nobot dynamic)
path=&_tmpcat (url=&_replay)
rs=none;
data _null_;
  file _webout;
  %put &mpoint;
  put "<table border cellpadding=10>";
  put "<td>";
  put "<b> Current Codes:</b><br><br>";
  put "data electronic;<br>";
  put "&nbsp;&nbsp;&nbsp;";
  put "infile 'electronic_data.dat';<br>";
  put "&nbsp;&nbsp;&nbsp;";
  put "input id $ region quarter gsale return discount nsale;<br>";
  put "run;<br><br>";
  put "PROC GCHART data=electronic;<br>";
  put "&nbsp;&nbsp;&nbsp;";
  put "&barchoice &barvar /<br>";
  put "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;";
  put "&sumvar_statement <br>";
  put "quit;<br>";
  put "<td>";
  put "<br><br>";
  put "PROC GCHART data=electronic;<br>";
  put "<FORM ACTION='your SAS broker program URL'>";
  put "<INPUT TYPE='HIDDEN' NAME='_PROGRAM' VALUE='your SAS program name'>";
  put "<input type='hidden' name='_service' value='production'>";
  put "<select name='barchoice'>";
  put "<option value='hbar' &hbar_select> hbar";
  put "<option value='vbar' &vbar_select> vbar";
  put "</select>";
  put "run;<br>";
  put "quit;";
  put "<br><br><input type='submit' value='submit'>";
  put "</table>";
run;

```

These lines are necessary for the Output Delivery System to put HTML output on the web. Note: (nobot dynamic) option is used here so that no "trailer" is generated by the ODS. As a result, the output stream is not "closed," thus allowing other HTML outputs to follow

Here we have a _null_ data set with the put statements to put customized HTML outputs on the web page. Note: use ' to enclose the put statement, and use ` to enclose anything else that may require parenthesis's.

Variables that begin with "&" are macro variables. They are defined in the %securitycheck macro.

The HTML form is generated and the input statement sends the information to the SAS server for processing.

%DRAWBAR MACRO CODE SAMPLE

```
%macro drawbar;  
ods html body=_webout  
path=&_tmpcat (url=&_replay)  
rs=none;  
proc gchart data=wkc110.electronic;  
format nsale gsale discount return dollar12. region region. id $product.;  
goptions reset=all;  
goptions device=gif;  
&barchoice &barvar  
  /&sumvar_statement  
  &discrete_statement  
  &groupstatement  
  &sumstatement;  
run;  
quit;  
goptions reset=all;  
ods html close;  
%mend;
```

These are the ODS statements that put the resulting graph on the web page. Notice that no "dynamic" keyword is needed. This is the case because the ODS was not "closed" from the previous object.

The format statements and the goptions statements. These are added to enhance the appearance of the graph.

These are the options that the user specified using the HTML form. As mentioned above, these macro variables contained the actual SAS statements. They were produced in the %securitycheck function.