Using SAS/GRAPH® Software to Analyze Student Study Habits

Bill Wallace
Computing Services
University of Saskatchewan

Abstract

This paper describes the steps taken to create unusual vertical bar charts for a set of data using an ANNOTATE= data set and PROC GANNO.

Creating bar charts with the SAS/GRAPH® PROC GCHART is straightforward if the data, and the output desired, fall within PROC GCHART's capabilities. PROC GCHART creates vertical bar charts by summarizing one or more variables from a data set. The bars on the chart represent categories or discrete groups of one variable. The project described in this paper required one bar chart for each observation with the values of fourteen variables represented by bars. When proc gchart cannot create the chart required, ANNOTATE= data sets are good alternative.

Introduction

A faculty member presented the following problem: to graph the results of a standardized questionnaire on student study habits. He wished to create a visual profile of the study habits of each student based on their responses; responses fell into five categories or areas (A, B, C, D, E), two of which were discrete, the remaining three consisting of four subcategories each (Figure 3). In effect, the study profiles consisted of 14 bars, two of which stood alone (categories C and E), while the other three (A, B, and D) were displayed in groups of four. The categories and subcategories had been determined in previous studies by factor analysis.

There were two options open: 1: transform the data into a form PROC GCHART could handle or 2: create a custom bar chart with an ANNOTATE= data set and PROC GANNO. The simplest approach was to use PROC GCHART and the BY statement to create a separate graph for each student. The problem was, each bar on the chart had to represent the value of one variable instead of each bar representing a discrete category for one variable.

First the data was transposed using PROC TRANSPOSE, into a form PROC GCHART could use to create a bar chart.

When it was realized that the results from PROC GCHART were not adequate, an ANNOTATE= data set was created. Both methods and their limitations are described here.

The Questionnaire

Students in different years of a professional college were asked to respond to a 53-item questionnaire which used a 4-point scale. Two examples of items were:

- I usually set out to understand thoroughly the meaning of what I do.
- I like to play around with ideas of my own related to what I'm studying.

The scale options and values were:

0 = definitely disagree
1 = disagree with reservation
2 = agree with reservation
3 = definitely agree

Students' responses to groups of questions were combined mathematically to determine a mean score for each category and subcategory, e.g. combined responses to items 22 and 30 (variables 22 and 30) produced an individual's mean score for subcategory 1 of category A, while responses to items 10, 20, 29, and 52 produced a score for category C.

Creating the Data Set

Analysis of the data resulted in means for the 14 subcategories, thereby creating 14 new variables identified as A1-A4, B1-B4, C, D1-D4, and E. The original variables (the scores on the 53 items on the questionnaire) were dropped along with the variables sex, age, and year.

The following SAS® statements read in the raw data and created a data set consisting of the student number and fourteen new variables containing the means.
PROC GCHART

The structure of the data file initially prevented us from using PROC GCHART. The data had to be transposed so the values for each student were in one column instead of in fourteen separate variables of one observation. The code below transposed the data by student number, creating fourteen observations, for every observation in the original data set (i.e., one observation for each variable).

```
libname dir '[]';
proc sort data=dir.class91:
by stnum:
proc transpose data=dir.class91
out=dir.trans;
by stnum;
var al--e;
```

Below is one observation before and after using PROC TRANSPOSE.

---

```
stnum A1 A2 A3 A4 B1 B2 B3 B4 C D1 D2 D3 D4 E
793925 2.0 1.5 2.0 1.3 1.7 1.5 1.5 1.75 1.2 1.5 1.0 2.0 1.0 1.5
```

---

```
stnum _name_ col1
793925 a1 2.0
793925 a2 1.5
793925 a3 1.5
793925 a4 1.3
793925 b1 1.7
```

---

Name of Former Variable

Figure 1
After running PROC TRANSPOSE there were three variables; stnum containing the student number, _name_ containing the name of variable before transposing, and col1 containing the actual value.

With the data in a form PROC GCHART could use, we proceeded with the chart creation. The bars on the chart had to be labelled with the subcategory names (i.e., _name_), the height of a bar represented the value for a subcategory (i.e., col1) and each chart represented one student (i.e., stnum).

The following SAS statements were used to create the charts.

```sas
libname dir '\[]';
proc gchart data=dir.trans;
  vbar _name_ /sumvar=col1;
  by stnum;
```

The code created one graph with fourteen bars for each student. The problem with the resulting chart (see Figure 1) was each subcategory had the same pattern and the bars were not separated into their categories. To correct this problem, a group variable was created and the VBAR options, GROUP= and PATTERNID=GROUP were added to PROC GCHART.

The GROUP= statement did not work in desired manner. It divided the x axis in to groups, but it assumed the fourteen subcategories in the variable _name_ were members of each group. The result was five groups A, B, C, D, and E with fourteen subcategories repeated for each group on the x axis (See Figure 2). The bars appeared over their correct label in the category they were a member (i.e., bar for A1 over the axis label A1 in group A). However, white space appeared above the subcategories which were not members of the current category.

Creating the Annotate Data Set

After viewing the results from PROC GCHART we decided to use an ANNOTATE= data set to create the bar graphs. This gave us full control over the appearance of the graphs.

![Figure 2](image-url)
The graph was first laid out by hand to determine the graph area, bar widths, and a coordinate system. The variables xsys and ysys were set to three, specifying screen percent coordinate system. The lower left hand corner of the screen size was 0,0 and the top right hand corner was 100, 100 (percent).

The variable FUNCTION, in an ANNOTATE= data set defines what object is drawn or what operation is executed. The values assigned to FUNCTION to create the chart were:

• LABEL: positions text at the current x, y coordinates.
• MOVE: moves an imaginary pen on the screen to new x, y coordinates.
• DRAW: draws a straight line beginning at the last x, y coordinates and ending at the new x, y coordinates.
• BAR: draws a rectangle with one corner defined by the last x, y coordinates and the opposite corner defined by the new x, y coordinates.

The FUNCTION values MOVE, DRAW, and LABEL were used to create and label the axes. The categories were labelled along the x axis while the subcategories were labelled vertically at the top of the chart. The x and y axis were drawn starting 5% from the left and 5% from the right edge of the screen.

The next step was to create the bars. The first FUNCTION value was MOVE to set the bottom left corner of the bar. The second FUNCTION value was BAR. The width of the bar was set to four % (i.e., x=x+4) and the height was the value taken from the variable being plotted (e.g., y=spg*25). The mean was multiplied by 25 such that if a mean for a variable was 3, then instead of a bar with a height 3% of the screen we acquired a bar 75% of the screen. Each bar’s width was 4% of the screen with 1% separating each bar in the same category and 5% separating each category.

The SAS statements necessary to create one bar chart were executed once for each student.

libname mydir '[]';
goptions nodisplay noprincipal;
data mydir. anna;
length function $8 text $20;
retain xsys ysys '3' x 5 style 'xswissb' position '4' angle 0;
/* Read each students data */
set mydir.class91;
/* Draw & label Y axis and draw X axis */
position='4'; color='black';
function='move'; x=5; y=5; output;
function='label'; text='0'; output;
function='move'; x=5; y=5+(25*1); output;
function='label'; text='1'; output;
function='move'; x=5; y=5+(25*2); output;
function='label'; text='2'; output;
function='move'; x=5; y=5+(25*3); output;
function='label'; text='3'; output;
function='draw'; x=100; output;
/* Label study areas on X axis*/
Position='5';
style='simplex'; color='black';
function='move'; y=96; x=95; output;
function='label'; text=stnum; output;
function='move'; y=3; x=15; output;
function='label'; text='Category A'; output;
function='move'; y=3; x=39; output;
function='label'; text='Category B'; output;
function='move'; y=3; x=55; output;
function='label'; text='Category C'; output;
function='move'; y=3; x=72; output;
function='label'; text='Category D'; output;
function='move'; y=3; x=88; output;
function='label'; text='Category E'; output;
/* Draw Bars */
/* Multiply means by 25 to use area */
/* Category A bars */
function='move'; x=5; y=5; output;
function='bar'; style='solid';
color='blue'; x=x+4; y=y+(25*smi); output;
function='move'; x=x+1; y=5; output;
function='bar'; x=x+4; y=y+(25*spg); output;
function='move'; x=x+1; y=5; output;
function='bar'; x=x+4; y=y+(25*spa); output;
function='move'; x=x+1; y=5; output;
function='bar'; x=x+4; y=y+(25*spp); output;
/* Category B bars */
function='move'; x=x+5; y=5; output;
function='bar'; style='solid';
color='green'; x=x+4; y=y+(25*dmi); output;
function='move'; x=x+1; y=5; output;
function='bar'; x=x+4; y=y+(25*dpo); output;
/* Category C bars */
function='move'; x=x+5; y=5; output;
function='bar'; style='solid';
color='red'; x=x+4; y=y+(25*stmi);
output;

/* Category D bars */
function='move'; x=x+5; y=5; output;
function='bar'; style='1'; x=x+4;
y=y+(25*pna); output;
function='move'; x=x+1; y=5; output;
function='bar'; x=x+4; y=y+(25*pg);
output;
function='move'; x=x+1; y=5; output;
function='bar'; x=x+4; y=y+(25*pi);
output;
function='move'; x=x+1; y=5; output;
function='bar'; x=x+4; y=y+(25*psm);
output;

/* Category E bars */
function='move'; x=x+5; y=5; output;
function='bar'; style='X3';
color='blue'; x=x+4; y=y+(25*DS);
output;
/* Label each bar */
position='6'; angle=90;
function='move'; x=x+9; output;
function='label'; text='B1'; output;
function='move'; x=x+5; output;
function='label'; text='B2'; output;
function='move'; x=x+5; output;
function='label'; text='B3'; output;
function='move'; x=x+5; output;
function='label'; text='B4'; output;
function='move'; x=x+9; output;
function='label'; text='C1'; output;
function='move'; x=x+9; output;
function='label'; text='C2'; output;
function='move'; x=x+5; output;
function='label'; text='C3'; output;
function='move'; x=x+5; output;
function='label'; text='C4'; output;
function='move'; x=x+5; output;
function='label'; text='D1'; output;
function='move'; x=x+5; output;
function='label'; text='D2'; output;

Figure 3
function='move'; x=x+5; output;
function='label'; text='D3'; output;
function='move'; x=x+5; output;
function='label'; text='D4'; output;
function='move'; x=x+9; output;
function='label'; text='E1'; output;

After the annotate data set was created, PROC GANNO created the graphs. The NAME= option on the PROC GANNO statement created a separate graph for each student. When a new student number was encountered in the annotate data set, PROC GANNO created a new graph.

The OPTIONS, NODISPLAY, and NOPROMPT, and the GOUT option on the PROC GANNO statement, placed the graphs in a graphics catalogue.

```
libname mydir '[]';
proc ganno annotate=mydir.anno
    name=stnum gout=mydir.graphcat;
```

The final step was to use PROC GREPLAY to create a template with sixteen panels and place a graph in each panel. The template was divided into sixteen even panels by breaking the x and y axis at 0, 25, 50, 75, and 100. This allowed us to view sixteen students at a time.

Conclusion

PROC GCHART is the preferred method of creating bar charts since it does most of the work for you. If the data you are working with does not fit PROC GCHART's capabilities, the next step would be to try altering the data's format. PROC TRANSPOSE should be considered whenever data is encountered that does not fit the requirements of a procedure. The combination of PROC TRANSPOSE and PROC GCHART produced a chart for each student with the fourteen variables. The appearance of the chart was not exactly what the faculty member wanted. The intended audience had to be able to view the different categories and subcategories quickly and easily. The GROUP= option on the VBAR statement did not create the individual groups of bars we wanted.

After encountering these difficulties with PROC GCHART, we turned to ANNOTATE= data sets. Programming ANNOTATE= data sets is more involved and takes longer, however, this is a trade off you must make if you require complete control over the appearance of the chart.

Some initial preparation can simplify the programming. Laying the chart out by hand on a grid facilitates the transfer to x, y coordinates. The next step was to determine the variables PROC ANNOTATE required.

The first annotate program you write might be confusing, however, once you understand the logic it's simply a matter of setting variables to their correct values in the correct order.

For more information contact the author at:
Bill Wallace
Computing Services
University of Saskatchewan
Physics 56
Saskatoon, Saskatchewan
S7N 0W0

(306)966-4823
Bitnet: wallace@sask.usask.ca

1. SAS and SAS/GRAPH are registered trademarks of SAS Institute Inc., Cary, NC, USA.