Paper 164-25

Achieving Graphical Excellence With SAS/GRAPH®

Archer R. Gravely, UNC Asheville, Asheville, NC

Abstract

This paper will integrate Tufte's (1983) principles of graphical excellence with the use of SAS/GRAPH to produce plots and bar charts. The application of these principles will enable users to create presentations that communicate complex ideas with clarity, precision, and efficiency while avoiding the pitfalls of "chartjunk", graphical decoration, unintentional "optical art", and other forms of bad graphical design. Guidelines for choosing among tables, graphs, and text as presentation devices are also discussed. Graphical excellence with SAS/GRAPH requires the use of customization features such as NOTE and AXIS statements. Examples of customized plots and bar charts that illustrate the principles of graphical excellence are provided. This paper is based on the book *Your Guide to Survey Research Using the SAS System* published by SAS Institute in 1998.

Introduction

The researcher has three main tools to convey the message of his or her work: words, tables, and graphs. Like a woodworker's chisels, saws, and planes, these tools are best used to perform distinct functions, but all are usually required in combination to produce a quality piece of work. Some general guidelines for the effective use of tables, graphs, and words are:

- Tables are the media of choice for conveying numerical *values*
- Graphs are best suited for depicting *relationships* among variables
- When small numerical differences have important implications, graphs tend to do a poor job of communicating the findings
- Tables do a much better job than words in comparing more than two numbers
- For reporting small data sets, tables are more effective than graphs
- Sentences are used to *summarize* information

Principles of Graphical Excellence

The graphical display of quantitative information is an essential tool for communicating key findings to readers and policy makers. Mahon (1977) makes the interesting point that the misuse or over-use of graphics is worse than to neglect them. Tufte's (1983, p.51) landmark work on the presentation of quantitative information provides a compelling set of principles to define graphical excellence:

- Well-designed presentation of interesting data - a matter of substance, of statistics, and of design
- Consists of complex ideas communicated with clarity, precision, and efficiency
- Gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space
- Nearly always multivariate
- Requires telling the truth about the data

Avoid Chartjunk and Graphical Decoration

It is an unfortunate fact that ugly, vibrating statistical graphics are all too common in the popular press, professional literature, and unpublished research reports. Tufte (1983, p. 107) coined the word *chartjunk* to describe poorly designed graphics that have the following characteristics:

- Use an overwhelming amount of ink to describe a few numbers
- Print label information vertically
- Use cross-hatching patterns that vibrate

- Attempt to describe differences with indecipherable graphical elements (e.g., pie charts, block charts, star charts)

Tufte makes the humorous point that chartjunk is often a form of “unintentional optical art”. Many chartjunk applications arise from the naive “look what I did with a computer” syndrome rather than a well thought-out concept of the data presentation. Without exception, the designers of graphical and presentation software (SAS included), have made it very easy for unwary users to produce chartjunk based on program defaults (e.g. , cross-hatched patterns) or easy to use procedures (e.g., pie charts, three-dimensional charts, block charts, bubble charts, etc.) that exemplify bad graphical design. Tufte (1983) provides us with a checklist of characteristics to describe *friendly* vs. *unfriendly* graphics.

Figure 1: Characteristics of User-Friendly vs. Unfriendly Graphics

Friendly	Unfriendly
Spell out words - avoid cryptic labels	Many abbreviations; require unnecessary reader effort to understand the intended message
labels are printed left to right	Vertical labels; words in several directions
Include messages to help reader understand data	Cryptic graphic information
Labels are placed on graphic so reader can instantly understand graphic elements	Make reader go back and forth between legend and graphic. Use cross-hatching and garish colors
Graphic entices reader - arouses curiosity	Graphic is chartjunk
Colors (if used) are sensitive to color blind readers. Blue can be distinguished from almost all other colors by most color deficient people	Use red and green for side by side contrasts
Type is easily readable and not overbearing	Type is clotted and overbearing
Type is upper/lower case, with serifs	Type is all upper case, sans serif

Customization Techniques

In developing graphical presentations of quantitative data, Bertin (1981) notes that we have eight visual variables for describing relationships or variation that the eye can perceive:

- X Y - the two dimensions of the plane
- Size
- Value
- Texture
- Color
- Orientation
- Shape

Graphical excellence is achieved through the thoughtful manipulation of these eight visual elements. The effective visual display of quantitative information with SAS/GRAPH (or any other graphics software), will almost always require users to modify program defaults to include: axis scaling, plot symbols, patterns, line characteristics, and legends. Customized graphics also require informed user control over graphic labeling, size, font, and color. We will illustrate the use of these customization features in creating bar charts with GCHART and plots with GPLOT.

Controlling Axes Appearance

The creation of information rich yet user-friendly graphics, will almost always require users to modify the scaling, tick marks, and labeling of axes for plots and histograms. Tight user control over axis scaling and appearance is a critical step in creating plots and histograms based on the principles of graphical excellence. SAS/GRAPH provides two mechanisms for controlling the appearance of axes in PROC GPLOT and PROC GCHART. Both GPLOT and GCHART have built-in options for controlling some aspects of axis appearance and scaling. Full user control of axis definitions is achieved through use of the **AXIS statement**. Both of these methods for specifying axis appearance provide the ability to control value ordering, axis color, and the number of minor tick marks. It is important to note that you can use LABEL statements with GPLOT and GCHART to label axes, but this procedure will not allow you to control color, font, size, and text angle. However, *all* aspects of axis appearance can be controlled using the AXIS statement which can then be referenced as a GPLOT or GCHART option. The process of defining AXIS statements and referencing them with a GPLOT/GCHART option is analogous to defining value labels with PROC FORMAT and then assigning them to a variable with a FORMAT statement.

Axis Statement

The axis statement provides full user control over all dimensions of vertical and horizontal axes for graphic output generated with GPLOT and GCHART. The syntax of the Axis statement and its most commonly used options are as follows:

```
AXIS<1...99>
          <scale options>
          <appearance options>
          <tick mark options>
          <text options>;
```

Scale/Tick Mark Options

ORDER=(*value list*)
MAJOR=(*description*)|NONE
MINOR=(*description*)|NONE

Where
description can include the following elements:

COLOR=*tick-mark color*
HEIGHT=*n* where n is number of units
NUMBER=*n*
where *n* is number of tick marks
WIDTH=*n*
where *n* is width of line

Selected Axis Appearance Options

COLOR=*axis color*
WIDTH=*n*

Selected Axis Labeling Options

LABEL=(*text-description* | 'text')
VALUE=(*text-description*| 'text')

Where *text-description* can include the following elements:

ANGLE=*degrees*
COLOR=*text-color*
FONT=*font-name*
HEIGHT=*n*
JUSTIFY=LEFT | CENTER | RIGHT
ROTATE=*degrees*

Write Text in the Graphics Area with NOTES

Providing readers with helpful label information is an important step in creating well designed user-friendly graphics. TITLES and FOOTNOTES provide header information, but these procedures do not allow you to write text in the graphics output area. As a principle of graphical excellence, the use of LEGENDS to identify plot lines is to be *avoided* as it makes the reader work harder than necessary to understand the most important dimensions of the graph. The NOTE statement provides an easy to use procedure for writing text directly in the graphic area with full user control over font, size, color, and angle. The ability to use NOTE statements to control the placement and appearance of text is a prerequisite for achieving graphical excellence with SAS/GRAPH. NOTES can be used in both GCHART and GPLOT.

The form of the NOTE statement is as follows:

NOTE <*options*> '*text*';

Selected Appearance Options

COLOR=*color*
FONT=*font*
HEIGHT=*n<units>* (units can be IN | CELL | PCT)

Placement Options

JUSTIFY=LEFT | CENTER | RIGHT (default is LEFT)
MOVE=(*x <units>, y <units>*)

where:
x = horizontal position
y = vertical position

Angling and Character Rotation

ANGLE=*degrees* (angle of baseline of entire text string)
ROTATE=*degrees* (angle at which each character is rotated)

The primary difference between NOTE and TITLE/FOOTNOTE statements is the power of the NOTE **MOVE=(x,y)** option to position text anywhere in the graphics output area. TITLE and FOOTNOTE statements can write text at the top and bottom of the graphics output area respectively. However, to position text with MOVE, you must specify *x* (horizontal) and *y* (vertical) coordinates. The coordinates can be expressed as cells (CELLS), percentages (PCT), inches (IN), or centimeters (CM). The default NOTE coordinate system is GUNIT=CELLS and the size of the CELLS is determined by the VPOS and HPOS graphics options. The CELL system is based on the VPOS and HPOS values for your device type. For example, if your device has a VPOS of 40 (rows) and a HPOS of 100 (columns), then the vertical CELL range of your graphics area is 1-40 and the horizontal range is 1-100. Thus, with this cell based coordinate system, the exact center of the graphics output would be MOVE=(20,50). You may find it easier to work with a percentage based coordinate system or the combination of CELLS for titles/footnotes and percentages for NOTES. You can override the specified GOPTION GUNIT value or CELL default inside the MOVE = statement by specifying the desired measurement unit beside the *x* and *y* values.

The specification of text labels positioned with coordinates may seem tedious, but with a little practice, this important task can be accomplished relatively quickly with only a couple of trial and error attempts. In assigning move=(x,y) coordinates, it is helpful to understand that MOVE=(50 pct, 50 pct) will always represent the *exact center* of your graphic and you can mentally adjust initial values from this reference point. Example NOTE statements will be illustrated in the examples to follow.

Bar Charts

PROC GCHART is used to create vertical and horizontal histograms or bar charts, pie charts, block charts, and star charts. Histograms are useful graphic devices for showing group differences and frequency distributions. Based on the principles of graphical excellence, pie charts, block charts, and star charts are *not* recommended for presenting quantitative data. Bertin (1981) and Tufte (1983, p. 178) argue convincingly that tables are far superior to *dumb* pie charts and the only thing worse than a single pie chart is a group of pie charts where:

> ... the viewer is asked to compare quantities located in spatial disarray both within and between pies ... given their low data-density and failure to order numbers along a visual dimension, pie-charts should *never* be used.

GCHART can be used to graph both numeric and character variables to show frequency counts, percentages, sums, and means. The bars can be arranged in groups and individual bars can also be segmented to represent group information. As a general rule, horizontal histograms are easier to read than vertical bar charts for three reasons.

- It is easier for the eye to make vertical comparisons of horizontal bar charts than to scan horizontally across a vertical bar chart.
- Horizontal bar charts have more room for label information than vertical bar charts.
- SAS/GRAPH, by default, provides a statistical table for horizontal bar charts to report frequencies, cumulative frequencies, percents, and cumulative percentages

Basic Bar Chart

We will illustrate the use of GCHART to generate a basic horizontal histogram or bar chart to show frequency counts and percentages with the following survey question.

Figure 2: Survey Example for Bar Charts

14. How important is it to you to take part in

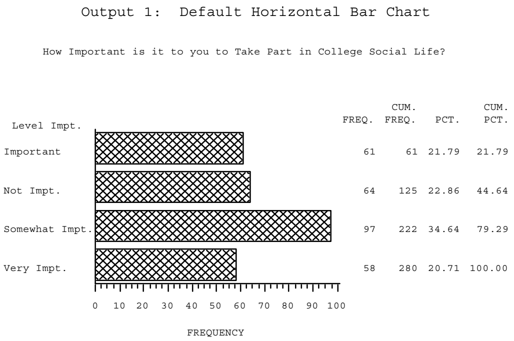
college life beyond attending class?

1 Not Important
2 Somewhat Important
3 Important
4 Very Important

In the SAS code to follow note that the HBAR statement is used to: (1) request a horizontal bar chart; and (2) identify the variable for which bars are created to represent each value. For example, HBAR SEX would create a bar chart with two bars: one for males and one for females. By default, SAS will create a histogram to show a frequency distribution of the variables specified on the HBAR/VBAR statement.

SAS Code for Basic Horizontal Bar Chart

```
libname db ‘c:\files\book\stud exp svy';

proc format;
  value $q14f '1' = 'Not Impt.'
              '2' = 'Somewhat Impt.'
              '3' = 'Important'
              '4' = 'Very Impt.';

proc gchart data = db.survey96;
hbar q14;
label q14 = 'Level Impt.';
format q14 $q14f. ;
title1 'Output 1:  Default Horizontal Bar
Chart';
title3
'How Important is it to you to Take Part
 in College Social Life?';
run;
quit;
```



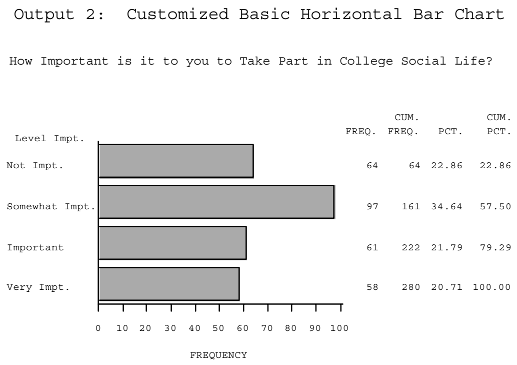
The basic horizontal bar chart shown in **Output 1** is easy to generate as it only requires only two GCHART statements. The PROC GCHART defaults, however, do not produce an aesthetically pleasing graphic that exemplify the principles of graphical excellence. The graph shown in **Output 1** would be improved by:

- Eliminating the vibrating cross-hatching patterns used with each bar
- Reducing the unnecessary detail on the horizontal response axis
- Ordering the bars by level of importance rather than the default alphabetical order of the bar labels.

These deficiencies noted in **Output 1** are remedied in the SAS code to follow. The PATTERN statement will be used to override the default cross-hatching black and white with a solid gray color. An additional bar definition feature is the use of the COUTLINE = BLACK option to define the color outline of the gray bars. By default, SAS will order bar charts according to the alphabetical order of PROC FORMAT assigned labels or data values if a format is not specified. Bars can also be ordered with the ASCENDING or DESCENDING option on the HBAR/VBAR statement. In this case, none of these three ordering options will provide the desired bar order. The MIDPOINTS = *label* option is used on the HBAR statement to order the bars in the desired sequence. Note that the *label* values are the PROC FORMAT specified value labels. The horizontal response axis values can be controlled with the RAXIS= options on the HBAR/VBAR statement. We can also control the number of minor tick marks with MINOR=*number* option. Lastly, the size of the text in the TITLE statements is specified with the HEIGHT or H= parameter.

Enhanced Basic Horizontal Bar Chart

```
pattern1 c=grayaa v=s;

proc gchart data =db.survey96;
hbar q14/midpoints='Not Impt.'
         'Somewhat Impt.' 'Important'
         'Very Impt.' coutline=black
          raxis=0 to 100 by 10 minor=0;
label q14 = 'Level Impt.' ;
format q14 $q14f. ;
title1 h=2.5 'Output 2:  Customized Basic
Horizontal Bar Chart';
title3 h=1.5
'How Important is it to you to Take Part in
College Social Life?';
run;
quit;
```



Bar Chart of Groups

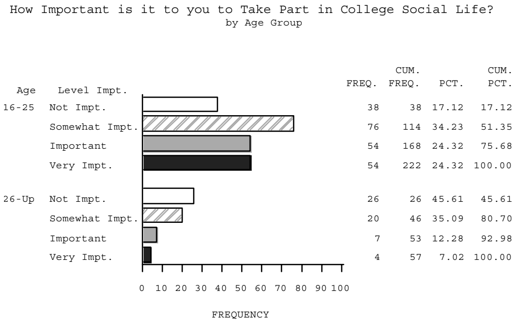
It is often of interest to generate a set of bars for two or more classification groups (race, sex, team, etc.) on the same graph. Bar charts are grouped using the **GROUP=** option on the VBAR/HBAR statement as shown in next example. It is important to note that SAS, by default, calculates percentages and cumulative frequency statistics for all groups *combined* rather than each group individually. Users will almost always want these statistics to be calculated for each group *separately* and this is accomplished with the **G100** option. With the G100 option, percentages will total 100% for each group separately. The **PATTERNID =** option is used to control the assignment of patterns to individual bars or groups. The **PATTERNID = MIDPOINT** option instructs SAS to change patterns for every bar. The **PATTERNID = GROUP** option value will change the pattern value when group value changes. To illustrate a bar chart with grouped data, the survey question shown in **Figure 2** will be graphed separately for two values of AGEGRP. The results are shown in **Output 3**.

SAS Code for Bar Chart of Groups

```
libname db 'c:\files\book\stud exp svy';

 proc format;
  value $q14f '1' = 'Not Impt.'
              '2' = 'Somewhat Impt.'
              '3' = 'Important'
              '4' = 'Very Impt.';

  value agegrp 16-25   = '16-25'
               26-high = '26-Up';

pattern1 c=grayff v=s;
pattern2 c=grayaa v=r3;
pattern3 c=grayaa v=s;
pattern4 c=gray22 v=s;

proc gchart data =db.survey96; where age > .;
hbar q14/group=age g100
 midpoints='Not Impt.' 'Somewhat Impt.'
'Important' 'Very Impt.'
 coutline=black raxis=0 to 100 by 10 minor=0
 patternid=midpoint;
label q14 = 'Level Impt.'
```

```
      age = 'Age';
format q14 $q14f. age agegrp. ;
title1 h=2.5 'Output 3: Horizontal Bar Chart
With Group Dimension';
title3 h=1.5
'How Important is it to you to Take Part in
College Social Life?';
title4 h=1.0 'by Age Group';
run;
quit;
```

Output 3: Horizontal Bar Chart With Group Dimension



Charting Multiple Response Items

PROC GCHART is designed to create histograms where the bars represent the values of a *single* variable. When reporting survey data for numerous items related to a common dimension, the use of multiple bar charts to report each item separately is not only tedious for the reader, but may obscure important relationships. A good technique for presenting this type of data is to transform the data so that multiple response variables can be captured as one variable and displayed on a single graphic. This type of data manipulation involves reshaping variables into observations. This type of presentation helps to achieve the *multivariate* principle of graphical excellence. To illustrate this process, we will reshape the four survey items shown in **Figure 3** and report them as a single dimension in the histogram shown in **Output 4**.

Figure 3: Survey Items for Reshaping Data Example

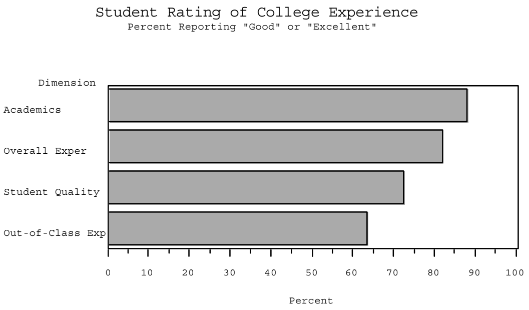
19.	How would you rate each of the following at UNCA?	Poor	Fair	Good	Excellent
a.	Your academic exper. . .	1	2	3	4
b.	Your out-of-class exper..	1	2	3	4
c.	Your overall experience	1	2	3	4
d.	The quality of students	1	2	3	4

In the following program, the four variables Q19A-Q19D will be represented by a single variable named DIM which will be assigned *values* to capture the original Q19A-Q19D *variable* names. A second new variable (PERCENT) will be created to report the percentage of respondents providing “Good” or “Excellent” responses to each of the four dimensions. PERCENT will be dichotomously scored (100/0) to compute this percentage using the mean statistic. Note the use of PROC TRANSPOSE to reshape the data set.

SAS Code for Reshaping Multiple Response Items

```
libname db 'c:\files\book\stud exp svy';

data survey;
   set db.survey96(keep=id q19a q19b q19c q19d);

proc sort data = survey; by id;

proc transpose data = survey out=reshape
     name=dim prefix=resp; by id;
     var q19a q19b q19c q19d ;

data final;
    set reshape;
    if resp1 in ('4','3') then percent = 100;
    else if resp1 in ('1','2') then percent = 0;

proc format;
    value $dim  'Q19A' = 'Academics'
                'Q19B' = 'Out-of-Class Exper'
                'Q19C' = 'Overall Exper'
                'Q19D' = 'Student Quality';

goptions reset=all dev=win ftext=zapf;

pattern1 c=grayaa v=s;

proc gchart data =final;
  hbar dim/sumvar=percent type=mean descending
  nostats coutline=black frame
  raxis=0 to 100 by 10 minor=1;
  label     DIM = 'Dimension'
        PERCENT = 'Percent';
format dim $dim. ;
title;
title1 h=1.5 'Output 5: Reshaping Data to Graph
Multiple Variables';
title3 h=1.3 'Student Rating of College
Experience';
title4 h=1.0 'Percent Reporting "Good" or
"Excellent"';
footnote;
run;
quit;
```

Output 4: Reshaping Data to Graph Multiple Variables



Subdivided Bar Charts

Subdivided bar charts allow you to subgroup *individual* bars into two or more segments to show group differences. As a graphics presentation device, the subdivision of bars with more than *two* segments is generally a bad idea as it is very difficult to interpret the spatial representation of three or more subgroups. However, segmented bar charts can be a useful technique under two conditions.

- The variables can be grouped into no more than two categories
- Interpretation of individual bar segments is of secondary importance to the total bar length.

Using these guidelines, the use of segmented bar charts can be a useful technique for describing differences among numerous groups, each represented by a single bar arranged in descending/ascending order. Vertical or horizontal reference lines can also be added to help the reader understand, at a glance, which groups are above or below the sample average or some other reference point.

In survey research applications, it is often advantageous to report four or five-point Likert type items as a combined “good/excellent” or “agree/strongly agree” single percentage measure rather than as mean scores or percentages for all scale values. With subdivided bars, the percent “good” and percent “excellent” can reported separately within each bar with the total bar length representing the combined “good/excellent” percentage statistic. We have found this to be a very effective presentation device for survey research data.

Subdivided bar charts are created with the SUBGROUP=*variable* option on the HBAR/VBAR statement. We will demonstrate this procedure with the survey question shown in **Figure 4**.

Figure 4: Survey Example for Subdivided Bar Chart

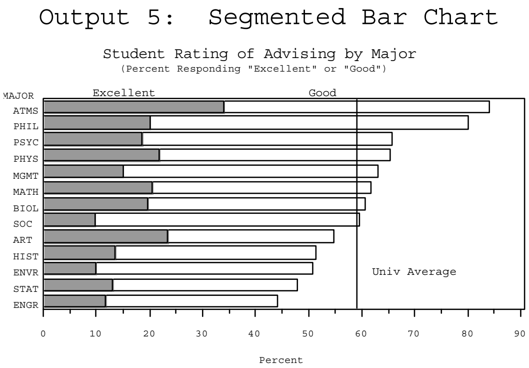
B01. How would you rate advising in your major?

1 Poor
2 Fair
3 Good
4 Excellent

As in the previous example, we will need to reshape the survey data into two new variables. The original “Excellent” or “Good” values will be transformed into a new variable called RESPONSE which has values of “G” or “E”. This RESPONSE variable will be used as the subgroup variable for segmenting the bars. We will compute the percentage of observations responding “good” or “excellent” by calculating the mean of a dummy coded (0/100) PERCENT variable. The PERCENT variable will be calculated and output separately for each value of RESPONSE (G/E). Since we are using the “mean” trick (Gravely, 1998) to calculate percentages, it is necessary to calculate these statistics prior to the PROC GCHART step and then use the SUM statistic in GCHART to display the results. By using the SUM statistic on a single summary measure for each bar, we are able to “trick” SAS into producing the desired graphic. The SAS code to follow will perform these operations with the resulting histogram shown in **Output 5**.

SAS Code for Subdividing Bars

```
libname library  ‘c:\files\book\rog assess svy';

data survey;
set library.progeval(keep=major qb01);
if qb01 = ' ' then delete;
if qb01 = '3' then percent= 100;
else if qb01 in ('1','2','4') then percent=0;
response = 'G';
output;
if qb01 = '4' then percent= 100;
else if qb01 in ('1','2','3') then percent=0;
response = 'E';
output;

proc summary data = survey nway;
  class  major response;
  var percent;
  output out = stats
  mean = ;

pattern1 c=gray99 v=s;
pattern2 c=grayff v=e;

proc gchart data = stats;
axis1 order = (0 to 90 by 10)
      minor = (n=1)
      label = ('Percent');
hbar major/sumvar=percent subgroup=response
          type=sum nostats
           descending vref=59 coutline=black
frame raxis=axis1
           nolegend;
note h=.9    move=(15,22) c=black 'Excellent'
             move=(50,22) c=black  'Good'
             move=(60,9)  c=black  'Univ
                  Average';
title1 h=2.5 'Output 5:  Segmented Bar Chart’;
title3 h=1.5
'Student Rating of Advising by Major';
title4 h=1.0
'(Percent Responding "Excellent" or "Good")';
run;
quit;
```



Note in **Output 5**, the use of the FRAME and NOSTAT options to frame the graphic area and suppress the default statistical table respectively. For this type of graph, the frequency and percentage statistics would take a lot of room and add little useful information. The FRAME option has the useful effect of keeping the reader's eye focused on the graphics space. We also chose to suppress the default legend as it tends to make the reader work harder than necessary to understand the graph. SAS NOTE statements are used to label the segments at the top of the first bar.

Plots

PROC GPLOT is used to create two-dimensional scatter and line plots with SAS/GRAPH. GPLOT has two major advantages over commonly used "point and click" spreadsheet/graphical software packages:

(1) GPLOT can easily create a second right-hand vertical axis
(2) SAS/GRAPH will automatically update the graphical display with new data by simply adding fresh observations to the data source file and re-executing the program.

GPLOT is an easy to use procedure for generating basic plots, but it requires knowledge of **SYMBOL**, **NOTE**, and **AXIS** statements to create user-friendly output that embody the principles of graphical excellence. We will illustrate the use of GPLOT to create three types of plots.

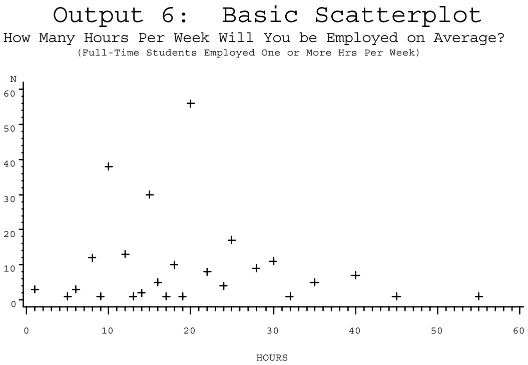
- Scatter plot (data points not connected)
- Single line plot (data points connected with a line)
- Multiple line plot (data points connected for more than one data series or group value)

Basic Scatter Plot

To illustrate the creation of a basic scatter plot using the PROC GPLOT defaults, we will plot the frequency distribution of employment hours (HOURS) for college students enrolled full-time (CREDHRLD >= 12). Before we can plot this distribution, we must first calculate the frequencies by summing the number of observations for each value of CREDHRLD. In this example, we will accomplish this task by creating a counter variable N with a value of 1 for each observation. PROC SUMMARY will then be used to sum N for each value of CREDHRLD. The last step will be to use PROC GPLOT to graph the SUMMARY output data set to produce the desired graph. The plot results are shown in **Output 6**.

SAS Code for Basic Scatter Plot

```
libname svy ‘c:\files\book\new student svy';

data newsvy;
  set svy.db96(keep=hours credhrld);
if credhrld >= 12;
n = 1;

proc summary data = newsvy nway;
  class hours;
  var n;
  output out = stats
  sum = ;

proc gplot data = stats;
plot n*hours;
title1 'Output 6:  Basic Scatterplot';
title2 'How Many Hours Per Week Will You be
Employed on Average?';
title3 '(Full-Time Students Employed One or More
Hrs Per Week)';
run;
quit;
```



Enhanced Scatter Plot

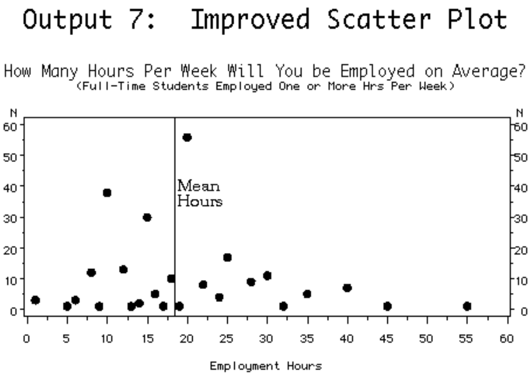
To illustrate the power of SAS/GRAPH to achieve graphical excellence, we will enhance the basic scatter plot shown in **Output 6** by customizing the following elements:

- Plot symbol (SYMBOL statement)
- Vertical and horizontal axis scaling (AXIS statement)
- Axis labeling (AXIS statement)
- Horizontal reference line (HREF)
- Placement of explanatory text in graphics output area (NOTE)
- Right hand vertical axis (PLOT2)
- Framing of graphics output area (FRAME)

These customization features are illustrated using the previous example. The data step commands are the same but the SAS/GRAPH enhancements are shown in bold print. The improved scatter plot is shown in **Output 7**.

SAS Code for Enhanced Scatter Plot

```
symbol1 c=gray00 v=diamond;

proc gplot data = stats;
axis1 order=(0 to 60 by 5)  minor=(number=1)
      label=('Employment Hours');
axis2 order=(0 to 60 by 10) minor=(number=1);
plot  n*hours/frame href=18.4 haxis=axis1
              vaxis=axis2;
plot2 n*hours/vaxis=axis2;
note h= 1.2  f=zapf
move=(33 pct,49 pct) 'Mean’
move=(33 pct,45 pct) ‘Hours';
title;
title1 'Output 7:  Improved Scatter Plot';
title2 'How Many Hours Per Week Will You be
Employed on Average?';
title3 '(Full-Time Students Employed One or More
Hrs Per Week)';
run;
quit;
```



Multiple Line Plot

Separate plot lines can be generated for each value of a group variable on the same graph with the following plot statement:

PLOT *vertical-var*horizontal-var***=group-var**;

To distinguish among the different group variable values, separate SYMBOL statements are needed to define plot symbols, color, and line characteristics. We will build on the previous example by adding a new variable (TSTUDENT) to illustrate this process. TSTUDENT has two values (transfer/freshmen) and the SAS code to follow will generate a separate line plot to show mean hours of employment over time for each TSTUDENT category. Note the modification of PROC SUMMARY to include TSTUDENT on the CLASS statement to calculate the summary plot variables.

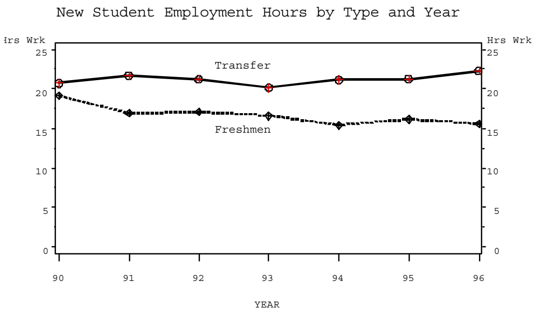
As a principle of graphical excellence, we almost never use legends to identify for the reader which plot lines represent each group value. Simply put, legends makes the reader work harder than necessary to understand the most important elements of the graph. The **NOLEGEND** option is used on both the PLOT and PLOT2 statements to suppress the printing of default legend. The **NOTE** statement is used to write text on the graphic to label each plot line with the proper group label. The NOTE statement provides us with full control over placement, color, font, angle, and size of text labels needed to make user-friendly graphics.

SAS Code for Multiple Line Plot

```
libname svy ‘c:\files\book\:new student svy';

data newsvy;
  set svy.db90
      svy.db91
      svy.db92
      svy.db93
      svy.db94
      svy.db96;
if credhrld >= 12;

proc summary data = newsvy nway;
  class year tstudent;
  var hours;
  output out = stats
  mean = ;

symbol1 v=diamond c=gray00 i=join line=2 w=4;
symbol2 v=circle  c=gray00 i=join line=1 w=4;

proc gplot data = stats;
axis1 order=(0 to 25 by 5) minor=(number=1)
      label=('Hrs Wrk');
 plot hours*year=tstudent/frame vaxis=axis1
                         nolegend;
 plot2 hours*year=tstudent/vaxis=axis1
                          nolegend;
 note move = (40,19) c=gray00 h=1.2 'Transfer'
      move = (40,13) c=gray00 h=1.2
        'Freshmen';
title1 'Output 8:  Multiple Line Plot';
```

```
title2 'New Student Employment Hours by Type and
Year';
run;
quit;
```

Output 8: Multiple Line Plot



Conclusions

Edward Tufte in his most recent work (1997) notes that “...clarity and excellence in thinking is very much like clarity and excellence in the display of data. When principles of design replicate principles of thought, the act of arranging information becomes an act of insight”. This paper has attempted to integrate the principles of graphical excellence with the use of SAS/GRAPH to generate bar charts and histograms. There are many computer graphics “point and click” software programs that are probably easier to use than SAS/GRAPH, but none can match its comprehensiveness and ability for users to control all aspects of the graphical presentation. The use of default values in all graphical software packages will usually create chartjunk. The key to achieving graphical excellence with SAS/GRAPH is to use AXIS and NOTE statements to control the two dimensions of the plane and to place explanatory text in the graphics output area to enhance readability. A more complete treatment of this topic is provided in Gravely’s 1998 *Your Guide to Survey Research Using the SAS System* published by SAS Institute.

References

Bertin, Jacques. *Graphics and Graphic Information Processing*. Berlin; New York: de Gruyter, 1981.

Carpenter, Arthur L. and Charles E. Shipp. *Quick Results With SAS/Graph Software*. Cary, NC: SAS Institute Inc., 1995.

Corning, Betsy. "Designing and Producing Effective Graphs With SAS/GRAPH Software: Using Graphic Elements Effectively." *Observations*, 1994, Vol 3, N2, pp-73-86.

Corning, Betsy. "Designing and Producing Effective Graphs With SAS/GRAPH Software: Presenting Data Effectively." *Observations*, 1994, Vol 3, N4, pp-73-86.

Corning, Betsy. "Designing and Producing Effective Graphs With SAS/GRAPH Software: Making Graphs Informative." *Observations*, 1994, Vol 3, N3, pp-3-17.

Ehrenberg, A.S.C., "Rudiments of Numeracy". *Journal of the Royal Statistical Society (A).* 1977, **140**, pp. 277-297.

Gravely, Archer R. *Your Guide to Survey Research Using the SAS System*. Cary, NC: SAS Institute Inc., 1998.

Mahon, B.H., "Statistics and Decisions: The Importance of Communication and the Power of Graphical Presentation". *Journal of the Royal Statistical Society (A).* 1977, **140** pp. 298-323.

SAS Institute, Inc. *SAS/GRAPH Software Reference*, Vol. 1 & 2, Version 6, First Edition. Cary, NC: 1990.

Tufte, Edward R. *The Visual Display of Quantitative Information*, Cheshire, Conn: Graphics Press, 1983.

Tufte, Edward R. *Visual Explanations: Images and Quantities, Evidence and Narrataive.* Cheshire, Conn: Graphics Press, 1997.

Author Contact

Archer R. Gravely
Office of Institutional Research -CPO 1460
UNC Asheville
One University Heights
Asheville, NC 28804-8503
email: gravely@unca.edu
phone: (828) 232-5118