

Charting the Basics with PROC GCHART

Perry Watts, Independent Consultant

Abstract

The bar chart is one of the most frequently requested graphs at all types of work sites: from banking to pharmaceuticals. Nevertheless, the SAS/GRAPH® GCHART procedure has a reputation for being confusing with the result that many users turn to Microsoft EXCEL for a quick solution to their graphics problems.

This tutorial presents the basic features of the GCHART procedure concentrating on the data structures that lend themselves to charting. Besides the VBAR statement with associated options for vertical bar charts, time will be spent on looking at how the PATTERN, AXIS, FORMAT and LEGEND statements can be used to build a good graph. Subgroup and Group charts are also discussed along with horizontal bar charts, pie, donut, and star charts.

Even if you have minimal experience with SAS/GRAPH software, you should come away from this presentation knowing how to quickly create a polished bar chart with GCHART.

Introduction: Sticking to the Basics

Every attempt is being made to *keep it simple*. Macros, annotate data sets and lengthy discussions about graphics statements, devices or catalog entries are not included in the paper. Also, the more complicated GREPLAY procedure only gets a passing reference when it is used to enhance a star chart.

Gratitude is extended to SAS Institute Inc. for granting permission to use the DEPT data set from *The How-To Book for SAS/GRAPH® Software* by Thomas Miron [4]. A slightly enhanced version of this data set is listed at the end of the paper.

SAS/GRAPH Statements

SAS/GRAPH procedures are supported by *statements* that affect how a graph is rendered. These statements are external to the PROCs and the ones used in this paper are global in scope. See Table 1 below.

Table 1. SAS/GRAPH statement definitions from [14], pp.161-162 that are used in this paper.

| Statement | Description |
|-----------|---|
| AXIS | Modifies the appearance, position, and range of values of axes in charts and plots |
| GOPTIONS | Submits graphics options that control the appearance of graphics elements by specifying characteristics such as ... fonts or text height. Graphics options can also temporarily change device settings. |
| LEGEND | Modifies the appearance and position of legends generated by procedures that produce charts, plots, and maps. |
| PATTERN | Controls the color and fill of patterns assigned to areas in charts, maps, and plots |
| TITLE | Add titles to graphics output. |

SYMBOL is omitted from the list, because it cannot be used in PROC GCHART.

The Bar Chart: For Discrete Data

Definition:

From Wikipedia:

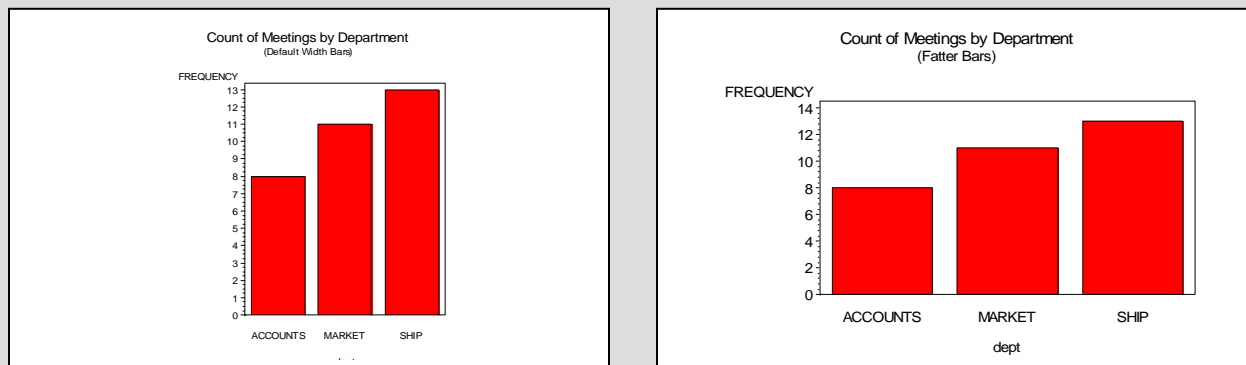
A **bar chart**, also known as a **bar graph**, is a chart with rectangular bars of lengths usually proportional to the magnitudes or frequencies of *what* they represent. Bar charts are used for comparing two or more values. The bars can be horizontally or vertically oriented. Sometimes a stretched graphic is used instead of a solid bar [7].

Missing from the definition is a description of the "what" that is being represented. For the bar chart that "what" is discrete nominal or ordinal data. Ranges, as we see later, should be reserved for histograms.

Also implicit in the Wikipedia definition is the fact that information is only conveyed by the length of the bars in a bar chart. Bar widths can be changed without altering the graph's message. The source code for the right-side panel Figure1 shows how easy it is to manipulate bar widths in PROC GCHART:

```
proc gchart data=N07.meetings;
  title2 "(Fatter Bars)";
  vbar dept/width=15;
run;
```

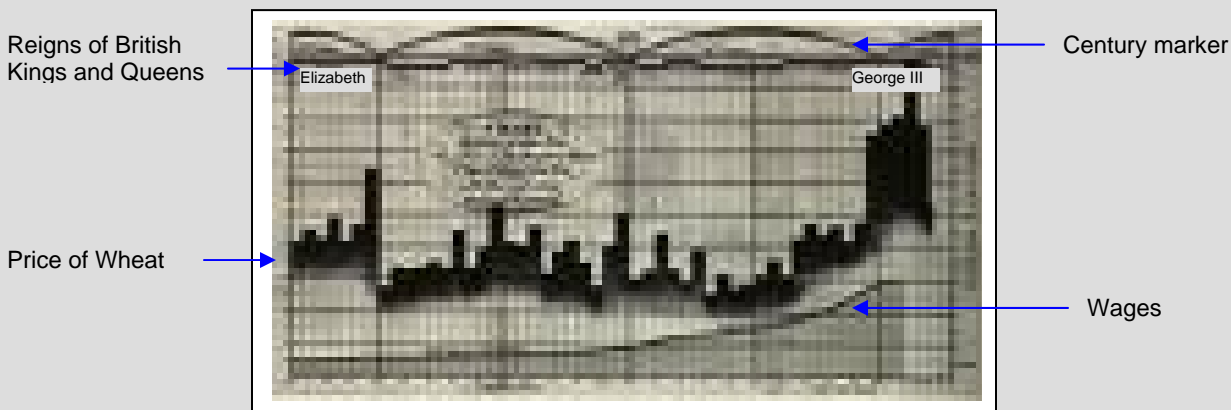
Figure 1. Two bar charts convey exactly the same information, because the width of the bars has no intrinsic meaning.



History:

Possibly the first bar charts appeared in the *Commercial and Political Atlas* (London, 1786) by William Playfair [7]. Displayed in Figure 2 is a hazy reproduction of Playfair's chart entitled *CHART, Shewing at One View The Price of The Quarter of Wheat, & Wages of Labour by the Week --from-- The Year 1565 to 1824*. A more readable version of this graph appears in *The Visual Display of Quantitative Information* by Edward Tufte [5], p. 34.

Figure 2. An early bar chart by William Playfair compares the price of wheat to the wages for skilled laborers.



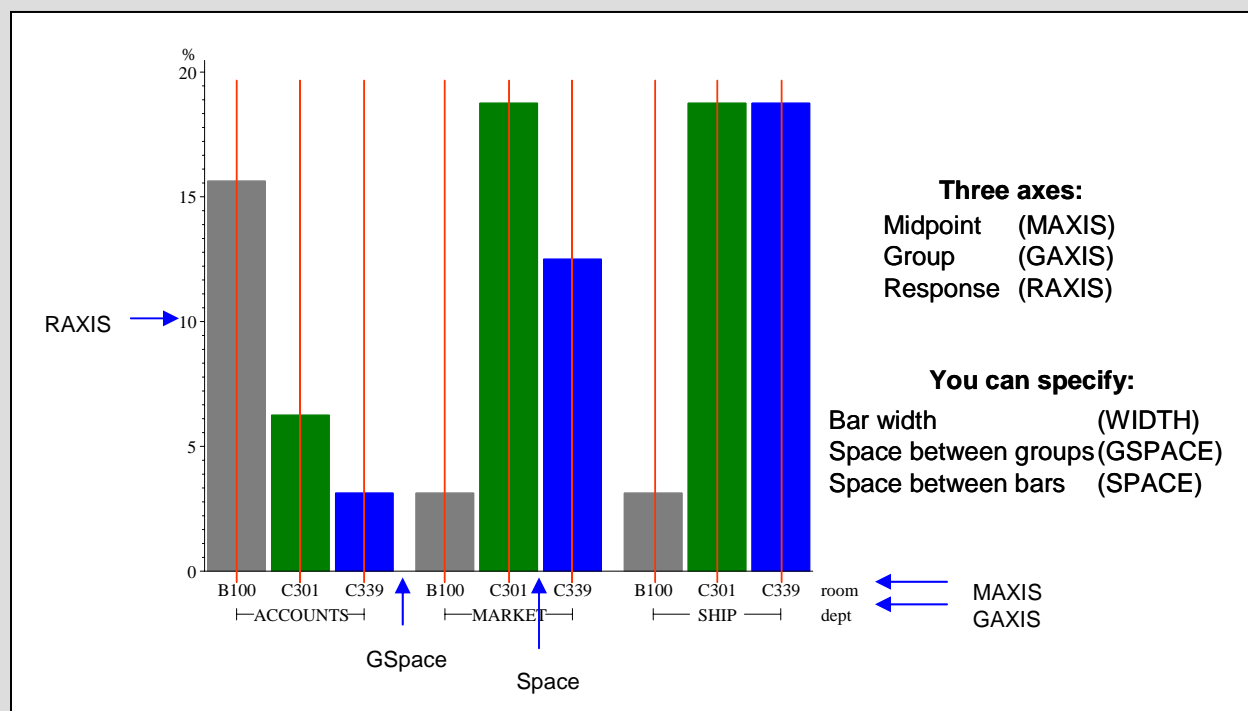
Playfair enhanced his chart by adding line plots for wages, reigns of monarchs and centuries. With these enhancements he anticipated the arrival of PROC GBARLINE in version 9 that produces an augmented vertical bar chart with a (line) plot overlay [15], p. 739. Nevertheless, the stand-alone bar chart with its simple structure has managed to endure over the centuries. Possibly its longevity can be attributed to its transparency. Bar charts convey brief messages quickly!

Bar Chart Structure: X-coordinates Don't Exist

The horizontal axis for x-coordinates in the GPLOT procedure is replaced by the midpoint (MAXIS) axis in GCHART. If needed, a hierarchical group axis (GAXIS) can be added. The only addressable coordinates in a SAS bar chart lie along the red vertical lines shown in Figure 3. Even the coordinates along the group axis are not

accessible to the programmer. This arrangement limits access to the graphic structure, but it makes it possible to easily change the values for WIDTH, SPACE and GSPACE that control the appearance of the bars.

Figure 3. GCHART replaces the horizontal axis in GPLOT with a midpoint axis. Default locations for the joint midpoint and group axes labels, *room* and *dept*, are also highlighted. Programmers have access only to coordinates along the red lines.



Procedure Syntax:

Only the options that are used in the paper are included here. For a complete listing, see the SAS/GRAPH Version 8 and Version 9 reference manuals [14], pp. 541-557 and [15], pp. 796-814. Copyright 1999, SAS Institute Inc., Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute Inc., Cary, NC

PROC GCHART<DATA=*input-data-set*>

VBAR | HBAR | VBAR3D *chart-variable(s)* </option(s)>

Where options following the forward slash (/) can include the following:

■ appearance options

CERROR=*error-bar-color*
COUTLINE=*bar-outline-color* | SAME
GSPACE=*group-spacing* (in cells)
LEGEND=LEGEND<1...99>
NOLEGEND
PATTERNID=GROUP | MIDPOINT | SUBGROUP
SHAPE=3D-*bar-shape* (HBAR3D and VBAR3D only)
SPACE=*bar-spacing* (in cells)
WIDTH=*bar-width* (in cells)
WOUTLINE=*bar-outline-width* (in pixels)

■ statistic options

CLM=*confidence-level*
ERRORBAR=BARS | BOTH | TOP
FREQ
FREQLABEL=*'column=label'* (HBAR only)
INSIDE=*statistic*
OUTSIDE=*statistic*
SUMLABEL=*'column=label'* (HBAR only)
SUMVAR=*summary-variable*
TYPE=*statistic*

■ axes options

DESCENDING
GAXIS=AXIS<1...99>
MAXIS=AXIS<1...99>
RAXIS=AXIS<1...99>

■ midpoint options

DISCRETE
GROUP=*group-variable*
MIDPOINTS=*value-list*
RANGE
SUBGROUP=*subgroup-variable*

Bar Chart TYPE and the Response Axis

First, more about GOPTIONS

To get graphics output, a GOPTIONS statement must be included in the code. Like SAS system OPTIONS, scope is global [4], p. 39. For a more comprehensive discussion about GOPTIONS see references [1], [3], [4], [14], [15], and [16]. Here is the code that will generate the first chart in the paper:

```
filename chrt1 'c:\N07\ChartBasics\Paper\EMF\chrt1.emf';
goptions device=emf gunit=pct rotate=landscape ftext="Arial";
title1 "Count of Meetings by Department";
goptions htext=13pt htitle=15pt gsfname=chrt1;
```

- **gsfname=chr1** GSF stands for **Graphics Stream File**. The argument points to an **.emf** file that can be inserted into PowerPoint or WORD. The same graph can be viewed in the Graphics Window of the enhanced editor by double-clicking on GCHART output listed in the RESULTS window.
- **device=emf** for Enhanced Windows Metafile is a device driver that is supported in SAS. EMF uses a vector format that produces a high-resolution graphic that can be resized without loss in quality [16],p.4. Raster or bitmap graphs such as *Playfair.jpg* displayed in Figure 2 are pixel-based which means the picture will degrade when it is enlarged.
- **gunit=pct** specifies the default unit of measure used with height specifications. Choices include CELLS (default), CM (centimeters), IN (inches), PCT (percentage of graphics output area), and PT (points). [14], p.350.
- **rotate=landscape** the graph's width is greater than its height.
- **ftext="Arial"** sets the font for all text in the graph. The argument is device dependent.
- **htext=13pt htitle=15pt** sets the default text and TITLE1 heights. A second GOPTIONS statement includes the HTEXT, HTITLE and GSFNAME clauses so that they can be altered for each graph that is generated in a multi-graph program. The changes made to the second GOPTIONS statement have no effect on the first.

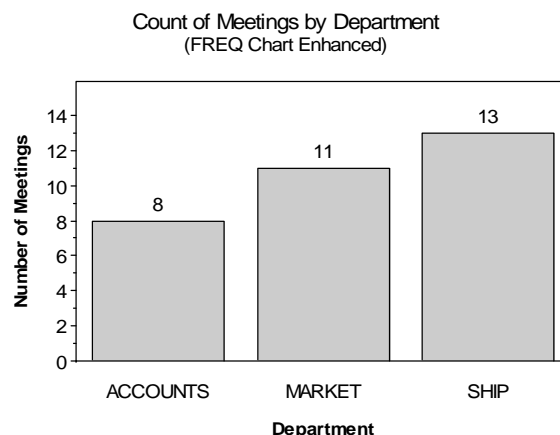
GOPTIONS statements do not appear elsewhere in the paper, but they can be found in the unabridged program on the NESUG CD. Code lines of lesser significance are also grayed out, and titles are diminished in size in the examples that follow.

Generating an Enhanced FREQ Chart (default TYPE)

Let's enhance the response axis (vertical RAXIS) with an axis statement that emboldens and rotates the label, assigns a meaningful number of minor ticks (1), and provides a range (0 to 14 by 2) that coincidentally leaves enough space for individual frequencies to be printed on top of the bars. In Figure 4 and many of the examples that follow, graphs are clipped and enlarged in WORD to make them more readable (**Format>Picture>Picture>Crop from>** and **Format>Picture>Size**).

Figure 4. An enhanced FREQ chart.

```
pattern1 color=grayCC;
axis1 label=(a=90 f="Arial/Bold"
"Number of Meetings") minor=(n=1)
order=(0 to 14 by 2);
axis2 label=
(f="Arial/Bold" "Department");
title1 "Count of Meetings by Department";
title2 '(FREQ Chart Enhanced)';
proc gchart data=N07.meetings;
vbar dept /
width=15
outside=freq
raxis=axis1
maxis=axis2
coutline=black
woutline=1;
run;
quit;
```



- **pattern1** The pattern statement sets the color and type of fill for all areas in a graph. The default type for a pattern fill is SOLID. Other options such as EMPTY and slanted hatch marks are also available. Because PATTERNID is not specified as a GCHART option, all the bars are colored the same. Also, while there is a COUTLINE option in GCHART, CBAR for assigning a uniform color to the bars doesn't exist.
- **axis1** The label for the response axis is rotated (a=90), and emboldened (f="Arial/bold"), provided with 1 minor tick mark that references an integer (n=1), and given a more expansive range (0 to 14 by 2). Note that AXIS1 is linked to PROC GCHART via the RAXIS option. For the label, formatting clauses (a= f=) must precede text assignments or they won't take effect.
- **outside=freq** displays the specified statistic (in this case FREQ) above the bar. The size of the text is the same as that used for the axis values, namely 13pt from GOPTIONS. This can become a problem when lots of bars are being displayed in a graph.
- **coutline=black woutline=1** the bars are set off with a black outline that is one pixel wide.

Generating Percent, Mean, and Sum Bar Charts

Few revisions are needed to go from the frequency chart depicted in Figure 4 to the percent, mean, and sum bar charts displayed in Figure 5. Repeated commands and titles are being removed from the code panel in Figure 5 to conserve space.

Figure 5. Percent, Mean and Sum bar charts.

Percent Chart

```
axis1 label=none order=(0 to 45 by 15)
minor=(n=2) value=('0%' '15%' '30%' '45%');
axis2 label=(f="Arial/Bold" "Department");
proc gchart data=N07.meetings;
vbar dept /
width=15 type=PCT inside=FREQ outside=PCT
raxis=axis1 maxis=axis2 coutline=black
woutline=1;
run; quit;
```

Mean Chart

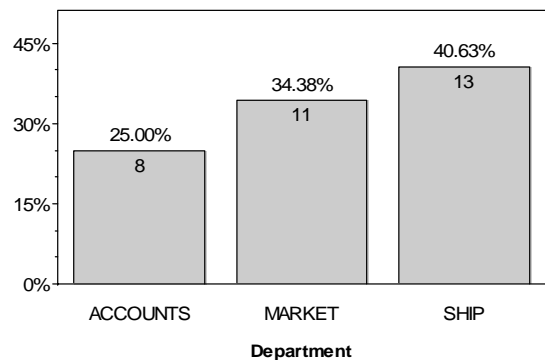
```
axis1 label=(f="Arial/Bold" "Hours") minor=NONE;
proc gchart ...;
vbar dept /
type=MEAN inside=FREQ outside=MEAN sumvar=Hours
run; quit;
```

Sum Chart

```
axis1 ... minor=(n=1);
proc gchart ...;
vbar dept /
type=SUM inside=MEAN outside=SUM sumvar=Hours
run; quit;
```

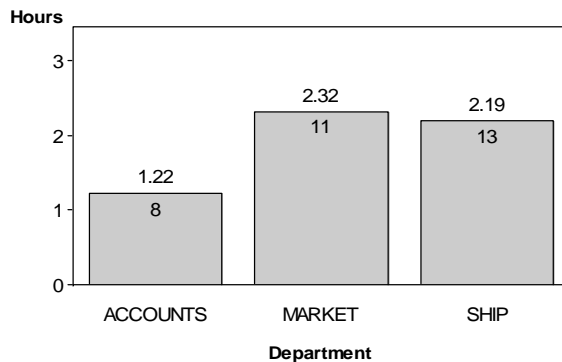
Percent Chart

Percentage of Meetings by Department
(Frequencies are listed INSIDE)



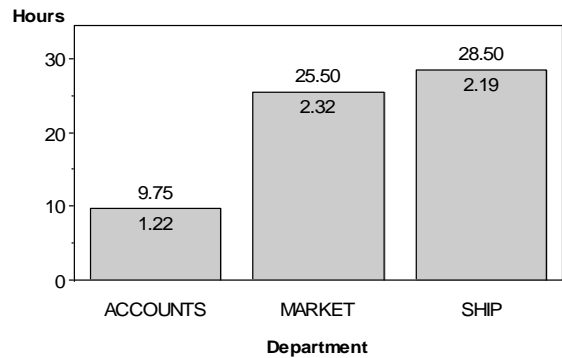
Mean Chart

Average Length of Meetings by Department
(Frequencies are listed INSIDE)



Sum Chart

Total Length of Meetings by Department
(Average Lengths are listed INSIDE)



- **Percent Chart** There is no need for a vertical axis label when the axis values contain percent signs. A range has been defined (0 to 45 by 15) so that the values are formatted properly with *Value=*. The argument to the TYPE option categorizes the chart. Note that OUTSIDE is also PCT to make the chart more readable. The INSIDE frequencies agree with the counts displayed in Figure 4.
- **Mean Chart** A mean must be taken of a numeric variable identified by the SUMVAR option. Remember that SUMVAR is short for summary-variable, not SUM as in addition. OUTSIDE is now the MEAN, and INSIDE again displays frequencies.
- **Sum Chart** Hours are now summed with SUMVAR, and OUTSIDE is SUM. This time MEAN that was OUTSIDE is moved to INSIDE.

OUTSIDE and INSIDE eliminate the need for grid lines or duplicate axes that tend to clutter a graph. However if the text extends significantly beyond the boundaries of the bars, it will automatically be removed from the graph.

VBAR and the Midpoint Axis

The variable assigned to VBAR (vertical bar) or HBAR (horizontal bar) always references the midpoint axis. The term *midpoint* is apt, since the value is center-justified under the bar for VBAR and half-way up a bar for HBAR. From Figure 3 it is recalled that the red lines are coincidental with the midpoint values and mark the only areas in the chart that can be referenced by the GCHART procedure. This positioning is ideal for discrete character data, but as we shall see, it can present challenges when continuous numeric variables are being charted.

Changing the Order of the Bars with MIDPOINTS= and DESCENDING options

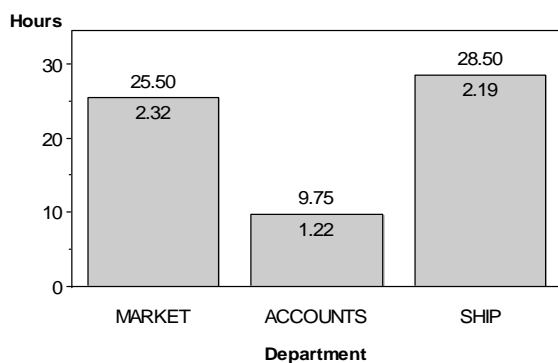
Bar order can be easily manipulated by selected options in the VBAR statement. By default, the bars are charted in alpha-numeric order. In Figure 6, instructions are given for changing that order. Applying a format to the variable DEPT can also be used to change bar order. See the source code on the NESUG CD for an example.

Figure 6. Changing bar order.

```
Midpoints=  
pattern1 color=grayCC;  
axis1 label=(f="Arial/Bold" "Hours") minor=(n=1);  
axis2 label=(f="Arial/Bold" "Department");  
proc gchart data=N07.meetings;  
  vbar dept /  
    width=15 type=SUM inside=MEAN outside=SUM sumvar=Hours  
    midpoints="MARKET" "ACCOUNTS" "SHIP"  
    raxis=axis1 maxis=axis2 coutline=black woutline=1;  
run; quit;  
Descending  
pattern1 color=grayCC;  
axis1 label=(f="Arial/Bold" "Hours") minor=(n=1);  
axis2 label=(f="Arial/Bold" "Department");  
proc gchart data=N07.meetings;  
  vbar dept /  
    width=15 type=SUM inside=MEAN outside=SUM sumvar=Hours  
    descending  
    raxis=axis1 maxis=axis2 coutline=black woutline=1;  
run; quit;
```

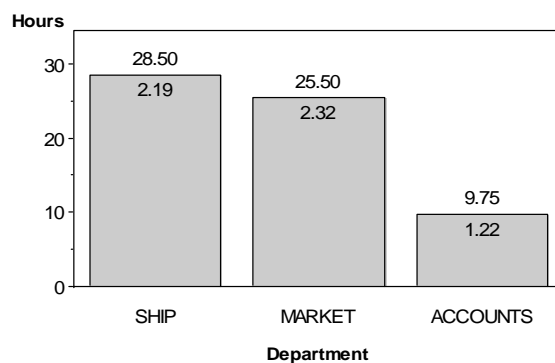
Order changed by the MIDPOINTS option

Total Length of Meetings by Department
(Order Determined by MIDPOINTS)



Order changed by the DESCENDING option

Total Length of Meetings by Department
(Order Determined by DESCENDING)

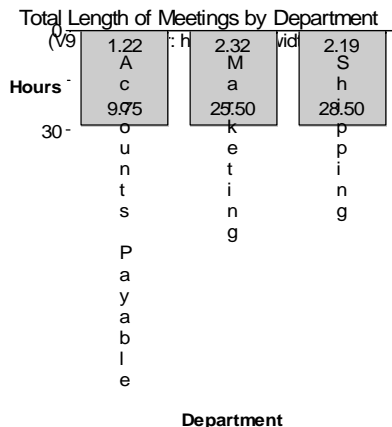


Coping with Text Overflow along the Midpoint Axis

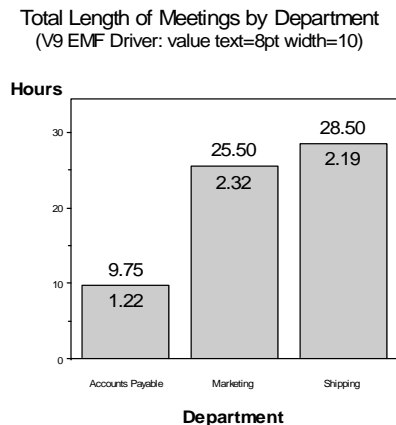
The length of the text for midpoint axis values is almost always limited by the width of the bars. In Figure 7, various methods are illustrated to show what can be done when text lengths exceed bar boundaries. The first panel shows the default SAS solution to this problem. The *Accounts* department has been extended by format to *Accounts Payable*. Relevant GCHART options are highlighted in the discussion that follows.

Figure 7. Dealing with midpoint value text that exceeds the width of chart bars.

Solution #1: Default, text goes to the next line

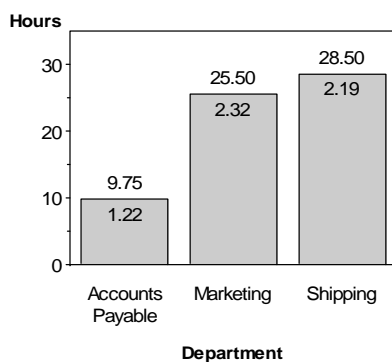


Solution #2: Reduce the Size of the Value Text



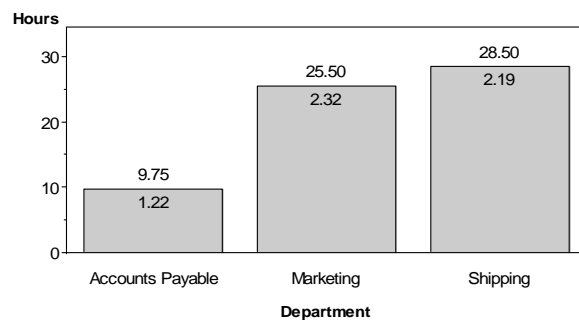
Solution #3: Use a SPLIT character in a Format

Total Length of Meetings by Department
(V9 EMF Driver SPLIT with a format: htext=13pt width=10)



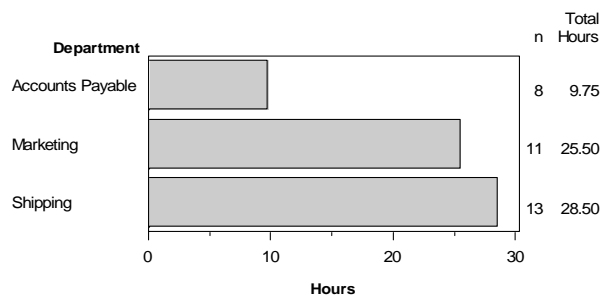
Solution #4: Increase the Width of the Bars

Total Length of Meetings by Department
(V9 EMF Driver: htext=13pt width=18)



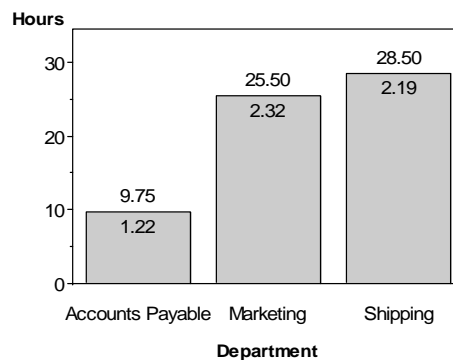
Solution #5: Use HBAR instead of VBAR

Total Length of Meetings by Department
(V9 EMF Driver HBAR: htext=13pt width=3.5)



Solution #6: Use the EMF Driver in V9 SAS

Total Length of Meetings by Department
(V9 EMF Driver: htext=13pt width=12)



- **Solution #1: Default, text goes to the next line.** A width of 10 cells cannot accommodate "Accounts Payable" when HTEXT is set to 13 points. The result is particularly atrocious when an EMF device is used in Version 9. For a more typical version of the default output, see Miron [4], p. 64.

```
/* Lengthen text with a format */
proc format;
  value $dptLfm 'ACCOUNTS'='Accounts Payable' 'MARKET'='Marketing'
    'SHIP'='Shipping';

goptions htext=13pt htitle=15pt gsfname=chrt11;
pattern1 color=grayCC;
title1 'Total Length of Meetings by Department';
title2 '(V9 EMF Driver: htext=13pt width=10)';
proc gchart data=N07.meetings;
  vbar dept /
    width=10 type=SUM inside=MEAN outside=SUM sumvar=Hours
    raxis=axis1 maxis=axis2 coutline=black woutline=1;
  format dept $dptLfm.;
run; quit;
```

- **Solution #2: Reduce the Size of the Value Text.** H(eights) or text size can be reset in the AXIS statement. Unfortunately, the text is unacceptably small in this example.

```
goptions htext=13pt htitle=15pt gsfname=chrt12;
title1 'Total Length of Meetings by Department';
title2 '(V9 EMF Driver: value text=8pt width=10)';
axis1 label=(f="Arial/Bold" "Hours") value=(h=8pt) minor=(n=1);
axis2 label=(f="Arial/Bold" "Department") value=(h=8pt);
proc gchart data=N07.meetings;
  vbar dept /
    width=10 type=SUM inside=MEAN outside=SUM sumvar=Hours
    raxis=axis1 maxis=axis2 coutline=black woutline=1;
  format dept $dptLfm.;
run; quit;
```

- **Solution #3: Use a SPLIT character in a Format.** SPLIT was introduced in Version 8 SAS. Therefore, an example of its use cannot be found in Miron's book.

```
proc format;
  value $dptSfm 'ACCOUNTS'='Accounts Payable' 'MARKET'='Marketing'
    'SHIP'='Shipping';

goptions htext=13pt htitle=15pt gsfname=chrt13;
title1 'Total Length of Meetings by Department';
title2 '(V9 EMF Driver SPLIT with a format: htext=13pt width=10)';
axis1 label=(f="Arial/Bold" "Hours") minor=(n=1);
axis2 label=(f="Arial/Bold" "Department") split="~";
proc gchart data=N07.meetings;
  vbar dept /
    width=10 type=SUM inside=MEAN outside=SUM sumvar=Hours
    raxis=axis1 maxis=axis2 coutline=black woutline=1;
  format dept $dptSfm.;
run; quit;
```

- **Solution #4: Increase the Width of the Bars.** This solution works well if there are few bars in the chart.

```
goptions htext=13pt htitle=15pt gsfname=chrt14;
title1 'Total Length of Meetings by Department';
title2 '(V9 EMF Driver: htext=13pt width=18)';
axis1 label=(f="Arial/Bold" "Hours") minor=(n=1);
axis2 label=(f="Arial/Bold" "Department");
proc gchart data=N07.meetings;
  vbar dept /
    width=18 type=SUM inside=MEAN outside=SUM sumvar=Hours
    raxis=axis1 maxis=axis2 coutline=black woutline=1;
  format dept $dptLfm.;
run; quit;
```


- **Solution #5: Use HBAR instead of VBAR.** While larger text is accommodated, statistics labels are "all or nothing" for an HBAR chart. If SUMS are desired, FREQS must also be displayed. The FREQLABEL and SUMLABEL options improve the appearance of the graph. Since there are fewer vertical than horizontal CELLS, WIDTH has to be reduced.

```

goptions htext=13pt htitle=15pt gsfname=chrt15;
title1 'Total Length of Meetings by Department';
title2 '(V9 EMF Driver HBAR: htext=13pt width=3.5)';
axis1 label=(f="Arial/Bold" "Hours") minor=(n=1);
axis2 label=(f="Arial/Bold" "Department") ;
proc gchart data=N07.meetings;
  HBAR dept /
    width=3.25 type=SUM sumvar=Hours
    freqlabel='n' sumlabel='Total Hours'
    raxis=axis1 maxis=axis2 coutline=black woutline=1;
  format dept $dptLfm.;
run; quit;

```

- **Solution #6: Use the EMF Driver in V9 SAS.** The EMF driver in Version 9 will allow text to extend a short distance beyond the bar borders. This facility does not extend to older drivers such as CGM.

```

goptions device=emf gunit=pct rotate=landscape ftext="Arial";
goptions htext=13pt htitle=15pt gsfname=chrt16;
pattern1 color=grayCC;
axis1 label=(f="Arial/Bold" "Hours") minor=(n=1);
axis2 label=(f="Arial/Bold" "Department");
title1 'Total Length of Meetings by Department';
title2 '(V9 EMF Driver: htext=13pt width=12)';
proc gchart data=N07.meetings;
  vbar dept /
    width=12 type=SUM inside=MEAN outside=SUM sumvar=Hours
    raxis=axis1 maxis=axis2 coutline=black woutline=1;
  format dept $dptLfm.;
run; quit;

```

The Segmented or Subgroup Bar Chart also relies exclusively on the Midpoint Axis

Each bar in a bar chart can be subdivided into segments that reflect the proportion of a subgroup's contribution to the height of a bar. Color is assigned to the segments via a PATTERN statement, and a LEGEND links color assignment to subgroup affiliation. A legend is automatically created for a subgroup chart unless NOLENGEND is specified. In Figure 8, the updated chart shows where departmental meetings are being held.

Figure 8. Dealing with text that exceeds the width of chart bars.

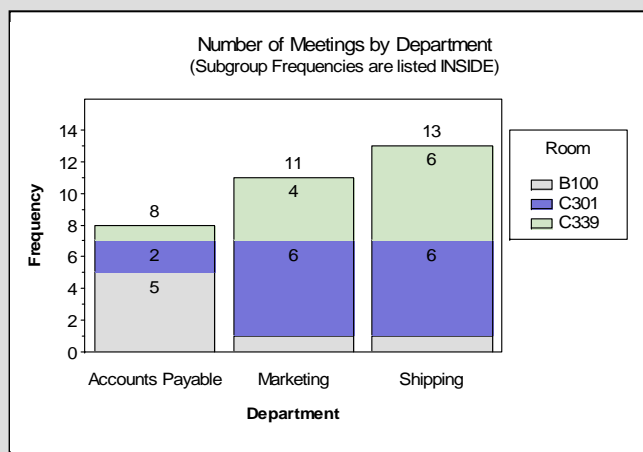
```

pattern1 color=graydd;
pattern2 color=VLIB; /* very-light-blue*/
pattern3 color=PALG; /* pale-light-green*/
legend1
  across=1 shape=bar(3,2)
  label=("Room" position=(top center))
  cborder=black position=(top outside right)
  offset=(-7,-7);

axis1 label=(a=90 f="Arial/Bold" "Frequency")
  order=(0 to 14 by 2) minor=(n=1);
axis2 label=(f="Arial/Bold" "Department");

title1 'Number of Meetings by Department';
title2 '(Subgroup Frequencies are listed INSIDE)';
proc gchart data=N07.meetings;
  vbar dept /
    width=15 type=FREQ
    outside=FREQ inside=Freq subgroup=room
    legend=legend1 raxis=axis1 maxis=axis2
    coutline=black woutline=1;
  format dept $dptLfm.;
run; quit;

```



- **pattern1... pattern3** Pattern statements define colors (in alphabetical order) for room numbers that reference subgroups in the bar chart. Light colors have been defined so that the counts are visible. When the frequency equals 1, the number is removed from the chart, because the referenced band height is too narrow. If HTEXT were set to 10pt all numbers would be displayed, but then the axes and label text would be too small.
- **Legend1** Here is where GUNITS=PCT from the GOPTIONS statement comes in handy. Units are automatically defined as PCT in the SHAPE and OFFSET legend options. A legend with one column is created with ACROSS=1. ROOM is center-justified over the numbers within the legend, but the legend itself is initially placed *top, outside* (the graphics plotting area) and to the *right*. OFFSET is then used to move the legend an additional 7 percent down and to the left.
- **inside=freq outside=freq** Subgroup and midpoint frequencies are displayed in the graph. Even though the value '1' is removed from the plot, it can be inferred by subtracting sum of the subgroup frequencies from the midpoint frequency. Nevertheless a graph with missing associated frequencies is not of publication quality. A solution to this problem involving ANNOTATE is described in *Building a Better Bar Chart with SAS/GRAPH® Software* [6]. With ANNOTATE, text size is not restricted by subgroup height.

Bar Chart GROUP and the Group Axis

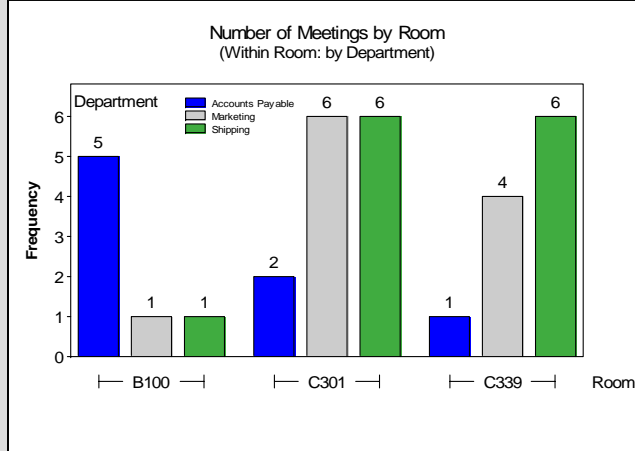
Groups can be accommodated in a bar chart, but then the number of bars is typically increased from m (#mid-points) to mXg (mX #group members). Bar widths decrease and midpoint axis text is even more constricted. A vertical arrangement of midpoint axis values is almost always guaranteed. A relatively simple solution presented in Figure 9 replaces the midpoint axis display with a legend. The appearance of a legend in a group chart involves some changes in options statements described below.

Figure 9. A group bar chart with a legend. mXg translates to 3 departments X 3 rooms for 9 bars total.

```
pattern1 color=blue; pattern2 color=graycc;
pattern3 color=CX40AC40; /* a shade of green */
legend1
across=1 shape=bar(3,2)
label=("Department" h=11 pt position=left j=c)
position=(top inside left) mode=share
value=(h=9pt 'Accounts Payable' 'Marketing'
'Shipping');

axis1 label=(a=90 f="Arial/Bold" "Frequency")
order=(0 to 6 by 1) minor=NONE offset=(,8pct);
axis2 label=none value=none;
axis3 label=("Room");

title1 'Number of Meetings by Room';
title2 '(Within Room: by Department)';
proc gchart data=N07.meetings;
vbar dept /
width=15 type=FREQ outside=FREQ
group=Room legend=legend1 subgroup=dept
raxis=axis1 maxis=axis2 gaxis=axis3
coutline=black woutline=1;
run; quit;
```



- **legend1** Additional features are highlighted. It is possible, for example, to change text heights in a legend. In this instance, the label "Department" is set to 11 points and values are reduced to 9 points. In addition, the label is moved to the left to take advantage of the bar placement in the graph. MODE is also set to SHARE in order to expose the value 5 over the first bar.
- **axis1** references the response axis. A relatively large offset of 8 percent accommodates the legend. There is no corresponding offset at the base of the graph, because the comma in (,8pct) serves as a place holder. Using an offset is better than increasing the axis range from 6 to 7. With an artificial maximum the response axis would be compromised.
- **axis2** references the midpoint axis. The axis must be declared in order to be hidden. Defaults generate values!
- **axis3** references the group axis. While the location for the label is fixed, "ROOM" is changed to "Room".
- **width=15** causes log message **WARNING: There was not enough room to use the specified width of 15 for the bars** to be generated. Also SAS resets the width of the bars internally along with values to SPACE (between the bars) and GSPACE (between the groups).

- **group=Room legend=legend1 subgroup=dept** Using these three options generates a group bar chart with a legend. The most curious of the options is SUBGROUP=DEPT. There is no subgroup in the chart, and DEPT is a midpoint not a subgroup variable. Also missing from the code is PATTERNID=MIDPOINT that would assign colors by DEPT. SAS will not insert a legend unless PATTERNID=SUBGROUP which happens when SUBGROUP=DEPT [14], p. 551.

Group Bar Charts Don't have to be Hierarchical

Data structures are not hierarchical when group affiliations are being charted for individuals. For example, William from SASHELP.Class can't be both male and female!

| Obs | Name | Sex | Age | Height |
|-----|---------|-----|-----|--------|
| 1 | Alfred | M | 14 | 69.0 |
| 2 | Alice | F | 13 | 56.5 |
| 3 | Barbara | F | 13 | 65.3 |
| 4 | Carol | F | 14 | 62.8 |
| 5 | Henry | M | 14 | 63.5 |
| 6 | Janet | F | 15 | 62.5 |
| 7 | Jeffrey | M | 13 | 62.5 |
| 8 | Judy | F | 14 | 64.3 |
| 9 | Mary | F | 15 | 66.5 |
| 10 | Ronald | M | 15 | 67.0 |
| 11 | William | M | 15 | 66.5 |

The NOZERO option is used to generate the non-hierarchical bar chart in Figure 10.

Figure 10. The SASHELP.class data set is used to plot heights for students ranging in age from 13 to 15 years.

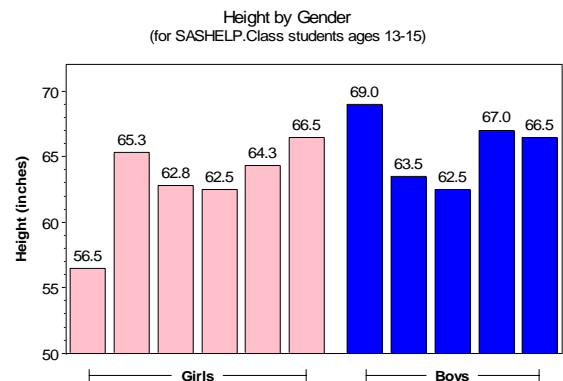
```

pattern1 color=pink; pattern2 color=blue;

axis1
  label=(a=90 f="Arial/Bold" "Height (inches)")
  order=(50 to 70 by 5) minor=(n=4);
axis2 label=NONE value=NONE;
axis3 label=none
  value=(f="Arial/Bold" "Girls" "Boys");

title1 'Height by Gender';
title2 '(for SASHELP.Class students ages 13-15)';
proc gchart data=N07.sasClass;
  vbar Name /
    width=5 type=SUM sumvar=Height outside=sum
    Group=sex patternID=group NOZERO
    raxis=axis1 maxis=axis2 gaxis=axis3
    coutline=black woutline=1;
run; quit;

```



- **pattern1 pattern2** reference 'F' and 'M' in alphabetical order from variable, SEX. Group order can be changed for *hierarchical* group bar charts by inserting an ORDER option into the group axis declaration. However, with NOZERO, re-ordering is not allowed. So girls must come before boys.
- **axis1** By starting the response axis at 50 inches, minor differences are emphasized. Thus it can be seen that the heights for both genders are roughly the same. The only exceptions are a single short girl and a tall boy.
- **axis2** The midpoint axis hidden, because we don't need to know *who* is being measured.
- **axis3** Group axis *values* are being emboldened to match the *label* for the response axis.
- **vBar Name** declares NAME as the midpoint variable.
- **width=5** is wider than the default width and easily accommodates OUTSIDE text at 15 points. When WIDTH is reduced to 2 cells, the OUTSIDE values disappear. (A NOTE is written to the log).
- **type=SUM sumvar=Height** Since the data are in summary format (one record per midpoint value), type becomes SUM. If FREQ were charted, all the bars would have a height of 1. MEAN, however, works just the same as SUM.
- **group=sex patternID=group** The group variable is defined and its values are linked to the pattern statement via PATTERNID. Remember, since PATTERNID is *not* SUBGROUP, no legend is generated.
- **NOZERO** suppresses the display of a bar when its frequency is zero (i.e. nonexistent). Thus the 'William' bar is excluded from the group labeled GIRLS. If NOZERO were not applied to this chart, space for 22 bars would be allocated in the chart.

Bar Chart Embellishments

Generating Error Bars

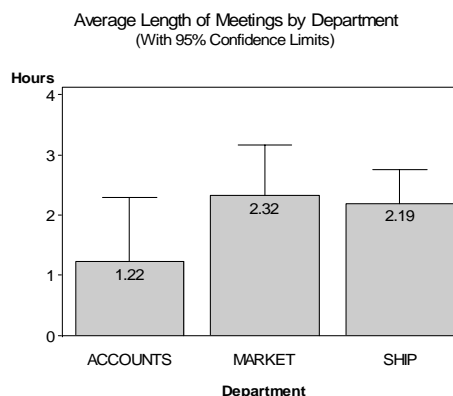
Error bars can be easily added to a bar chart when SUMVAR is set to MEAN or PCT. Algorithms for calculating error bar heights from raw data are described in the SAS/GRAPH manual [14], p.547. In Figure 11, error bars are displayed for average meeting times per department.

Figure 11. Generating Error Bars in PROC GCHART.

```
pattern1 color=grayCC;

axis1 label=(f="Arial/Bold" "Hours")
  minor=NONE;
axis2 label=(f="Arial/Bold" "Department");

title1 'Average Length of Meetings by Department';
title2 '(With 95% Confidence Limits)';
proc gchart data=N07.meetings;
  vbar dept /
    width=15 type=MEAN inside=MEAN
    errorbar=TOP clm=95 sumvar=Hours
    raxis=axis1 maxis=axis2
    coutline=black woutline=1;
run; quit;
```



- **type=MEAN inside=MEAN sumvar=Hours** Error bars are being generated for the means of a continuous variable. TOP error bars and INSIDE statistics can be displayed together.
- **errorbar=TOP clm=95 coutline=black woutline=1** Besides TOP, BOTH (tops and bottoms) and BARS shown on page 595 in [14] are available for selection. Error bars with 95% confidence limits are being plotted. Finally, error bars and bar outlines are configured the same with *coutline* and *woutline*.

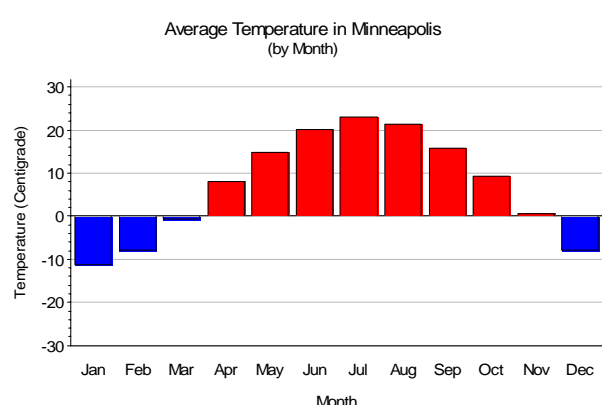
Generating Min/Max Charts

The method for assigning colors in Figure 12 comes from an unpublished CD entitled *Robert Allison's SAS/Graph Examples!* In Allison's program, COL2.SAS, colors are assigned by the SUBGROUP option. This is the second time a SUBGROUP assignment is altered to introduce added functionality into GCHART. Earlier we saw how SUBGROUP could be used to include a legend into a group chart, and here whole bars are being assigned colors in a midpoint chart by referencing a binary variable that ascertains if the temperature is above or below zero.

Figure 12. Generating a Min/Max chart in PROC GCHART.

```
pattern1 color=Blue;
pattern2 color=Red;
axis1 label=(a=90 'Temperature (Centigrade)')
  minor=(n=4) order=(-30 to 30 by 10);
axis2 label=('Month');
title1 'Average Temperature in Minneapolis';
title2 '(by Month)';

proc gchart data=n07.MplsWeather;
  vbar month / discrete noframe
    width=5 type=sum sumvar=TC
    subgroup=AboveZeroYvN nolegend
    raxis=axis1 maxis=axis2
    autoref clipref cref=graybb
    coutline=black woutline=1;
  format month monthfmt.;
run; quit;
```



- **discrete** The unformatted values for month are numbers ranging from 1 to 12. Up to now all midpoint values have been character. For numbers, GCHART internally calculates midpoints from the data range. What DISCRETE does is to instruct GCHART to omit the calculation and use the raw values instead.
- **subgroup=AboveZeroYvN nolegend** The variable AboveZeroYvN contains an 'N' when the temperature is below zero and a 'Y' when it is above zero. Thus PATTERN1 is set to blue (cold) and PATTERN2 is red (hot). Since a temperature cannot be both above and below zero, all the bars are assigned single colors.
- **autoref clipref cref noframe** Reference lines are used to make it easier to determine the temperature at given months. CLIPREF places the lines behind the bars, CREF colors them light gray so that they aren't too prominent, and NOFRAME gets rid of the default frame around the plotting region.

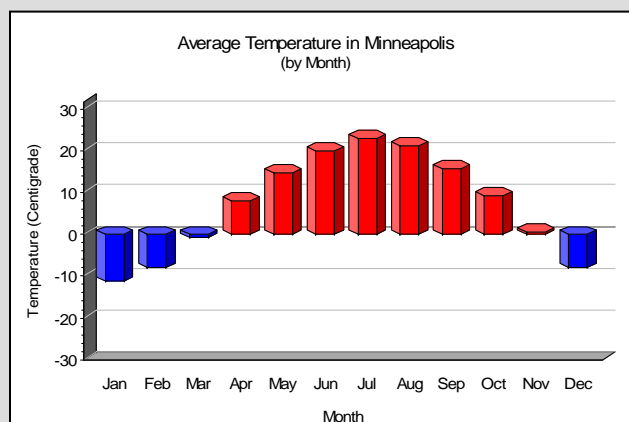
Generating 3-D Bar Charts in GCHART: Not Recommended but Supported by SAS

The revised 3-D version of Temperatures in Minneapolis shown in Figure 13 does nothing to enhance the underlying data that are being reported. January is still the coldest month, and July is the hottest month. If anything, the hexagons that replace the short bars for March and November distort the data, because they are over-sized.

Figure 13. Generating a 3-D chart in PROC GCHART only requires a change from VBAR to VBAR3D and the addition of the SHAPE option. Choices for SHAPES include: BLOCK (default), PRISM, STAR and HEXAGON.

```
pattern1 color=Blue;
pattern2 color=Red;
axis1 label=(a=90 'Temperature (Centigrade)')
  minor=(n=4) order=(-30 to 30 by 10);
axis2 label=('Month');
title1 'Average Temperature in Minneapolis';
title2 '(by Month)';

proc gchart data=n07.MplsWeather;
  vbar3D month / discrete noframe
  shape=hexagon
  width=5 type=sum sumvar=TC
  subgroup=AboveZeroYvN nolegend
  raxis=axis1 maxis=axis2
  autoref clipref cref=graybb
  coutline=black woutline=1;
  format month monthfmt.;
run; quit;
```



The Histogram: For Continuous Data

Definition:

From Wikipedia:

In statistics, a **histogram** is a graphical display of tabulated frequencies. A histogram is the graphical version of a table which shows what proportion of cases fall into each of several or many specified categories. The histogram differs from a bar chart in that it is the *area* of the bar that denotes the value, not the height, a crucial distinction when the categories are not of uniform width (Lancaster, 1974). The categories are usually specified as non-overlapping intervals of some variable. The categories (bars) must be adjacent.

The word *histogram* is derived from Greek: *histos* 'anything set upright' (as the masts of a ship, the bar of a loom, or the vertical bars of a histogram); *gramma* 'drawing, record, writing' [8].

For a histogram then, both height and width of the bar convey information about the category that is being measured. For width to have significance, the data need to be continuous.

Creating Histograms in PROC GCHART does NOT work

The Discrete Option Revisited

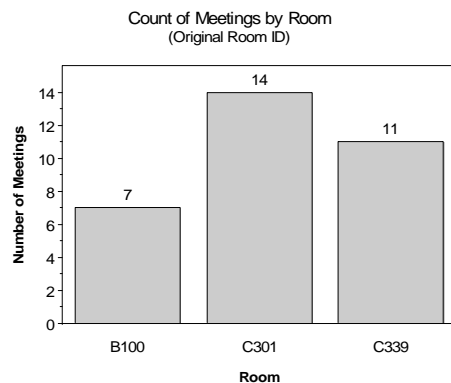
The DISCRETE option was touched upon briefly in the discussion about months reformatted as numbers in the temperature data associated with Figures 12 and 13. In Figure 14, the impact of DISCRETE on ROOM NUMBERS is assessed, whereas in Figure 15 DISCRETE is removed to accommodate continuous data. For Figure 14, a new variable ROOMNUMBER is calculated as:

```
roomNumber=input(substr(room,2),3.);
```

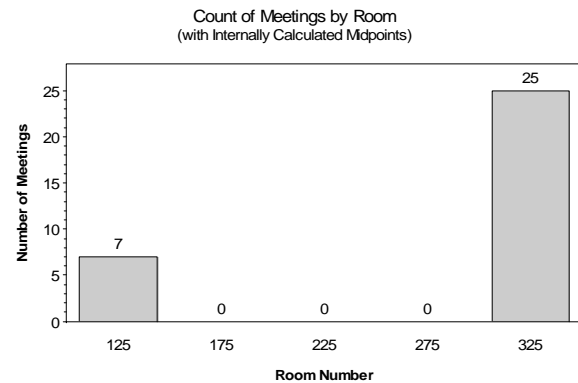
SAS uses an internal algorithm to calculate midpoints when the DISCRETE option is omitted from the code. This has a disastrous effect on panels 2 and 3 in Figure 14. The room numbers don't even exist!

Figure 14. How MIDPOINT Value Types and the DISCRETE Option Interact in PROC GCHART.

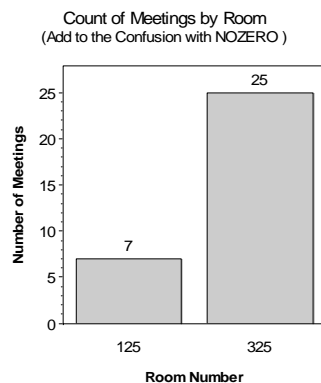
1) Value Type: Character, DISCRETE is irrelevant



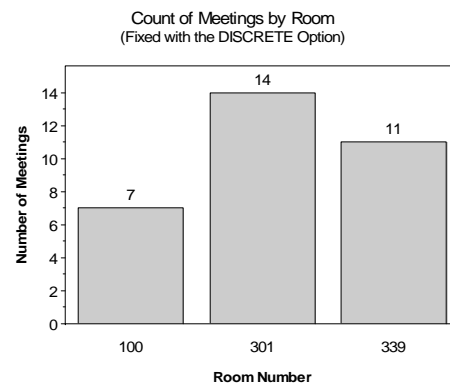
2) Value Type: Numeric, No DISCRETE (SAS takes over)



3) Value Type: Numeric, No DISCRETE NOZERO Confuses



4) Value Type: Numeric, with DISCRETE Duplicates #1



Code for Panel 4 that uses DISCRETE Correctly

```
pattern1 color=grayCC;
axis1 label=(a=90 f="Arial/Bold" "Number of Meetings")
  order=(0 to 14 by 2) minor=(n=1);
axis2 label=(f="Arial/Bold" "Room Number");
title1 "Count of Meetings by Room";
title2 '(Fixed with the DISCRETE Option)';
proc gchart data=N07.meetings;
  vbar roomNumber / discrete
    width=15 outside=freq raxis=axis1 maxis=axis2
    coutline=black woutline=1;
run;
```

While *room numbers* are discrete, *hours* are continuous. Therefore, the source code in Figure 15 below is highlighted to show how options can be used to display continuous data. In the second panel, SPACE is set to zero so that the bars will coincide with each other, and the LEVELS option in the fourth panel generates a four-bar bar chart with midpoints determined internally by SAS. The RANGE option for continuous data is also used.

Despite these manipulations valid histograms cannot be produced in PROC GCHART. If, for example, continuous data are being graphed, the lowest value for HOURS in panel #2 should be -0.4. However, negative time doesn't exist! Worse yet are the unequal ranges in the fourth panel that appear under equal-width bars. By definition, bar *area* not height alone defines a histogram. Use PROC UNIVARIATE instead to generate histograms.

Figure 15 Working with Continuous Data in PROC GCHART.

1) Source Code for panels #2-#4.

2) NO DISCRETE (Internal algorithm assigns midpoints)

```
proc gchart data=N07.meetings;
  vbar hours / space=0
  width=10 outside=freq raxis=axis1
  maxis=axis2 coutline=black woutline=1;
run; quit;
```

3) Actual Hours with DISCRETE Option

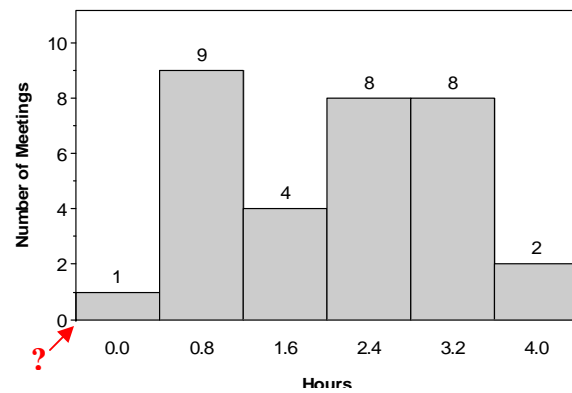
```
proc gchart data=N07.meetings;
  vbar hours / discrete space=1
  width=7 outside=freq raxis=axis1
  maxis=axis2 coutline=black woutline=1;
run; quit;
```

4) Bars with Ranges for a Histogram

```
proc gchart data=N07.meetings;
  vbar hours / levels=4 range space=0
  width=15 outside=freq raxis=axis1
  maxis=axis2 coutline=black woutline=1;
run; quit;
```

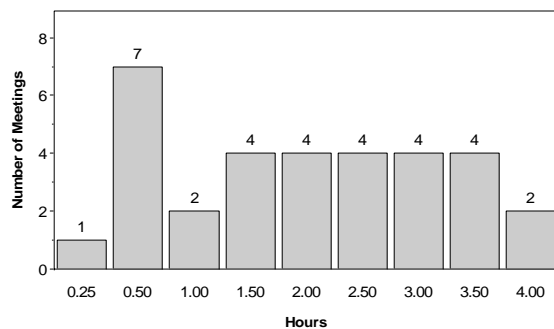
2) NO DISCRETE (Internal algorithm assigns midpoints)

Count of Meetings by their Length
(With Internally Calculated Midpoints)



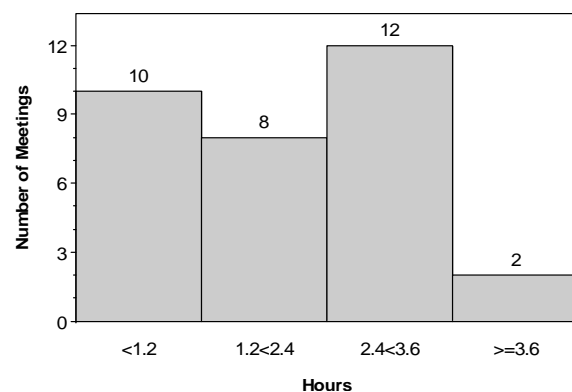
3) Actual Hours with DISCRETE Option

Count of Meetings by their Length
(Actual Hours with Discrete Option)

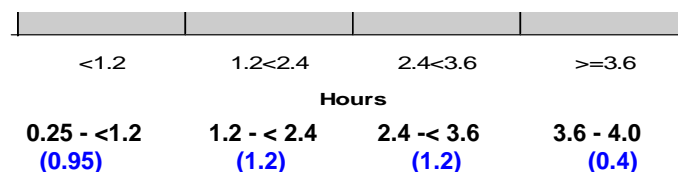


4) Bars with Ranges for Histograms

Count of Meetings by their Length
(Levels with Range)



5) Enlargement of the Ranges from Panel #4. Actual ranges in blue combine information from the third and fourth panels.



Charts from panels #2 and #3 in Figure 15 come from *The How-To Book for SAS/GRAPH® Software* [4], pp. 89-92.

Circular Charts: Pie, Donut and Star Charts

Description and Uses:

Pie charts are used to show classes or groups of data in proportion to the whole data set. The entire pie represents all the data, while each slice represents a different class or group within the whole [13].

Use pie charts to...

- convey approximate proportional relationships (relative amounts) at a point in time
- compare part of a whole at a given point in time
- Exploded: emphasize a small proportion of parts

Do not Use ...

For exact comparisons of values, because estimating angles is difficult for people.

For rank data: Use column/bar charts in this case; use multiple column/bar charts for grouped data [10].

Procedure Syntax:

The option list below is abbreviated, since circular charts are only getting a cursory review in this paper. Knowledge about VBAR should make it much easier to learn how to construct PIE, DONUT and STAR charts, since all are derived from the same graphics procedure. For a complete listing of the syntax, see the SAS/GRAPH Version 8 and 9 reference manuals [14], pp. 559-582 and [15], pp. 818-842. Copyright 1999, SAS Institute Inc., Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute Inc., Cary, NC

PROC GCHART<DATA=*input-data-set*>

PIE | DONUT | STAR *chart-variable(s)* </option(s)>

Where options following the forward slash (/) can include the following:

■ **appearance options**

COUTLINE=*bar-outline-color* | SAME

FILL=SOLID | X

LEGEND=LEGEND<1...99>

NO CONNECT (STAR only)

NOHEADING

WOUTLINE=*bar-outline-width* (in pixels)

■ **statistic options**

SUMVAR=*summary-variable*

■ **grouping and subgrouping options**

SUBGROUP=*subgroup--variable*

■ **slice-labeling options**

LEGEND | LEGEND=LEGEND<1...99>

PERCENT=ARROW | INSIDE | NONE | OUTSIDE>

SLICE=ARROW | INSIDE | NONE | OUTSIDE>

VALUE=ARROW | INSIDE | NONE | OUTSIDE>

■ **donut-labeling options (DONUT only):**

DONUTPCT=*percent*

LABEL=(*text argument(s)*)

■ **catalog entry description options (for GREPLAY)**

NAME=*'entry-name'*

Pie and Donut Charts:

A donut chart is an enhanced pie chart that can handle the display of subgroup affiliations. The two types of charts are shown in Figures 16 and 17.

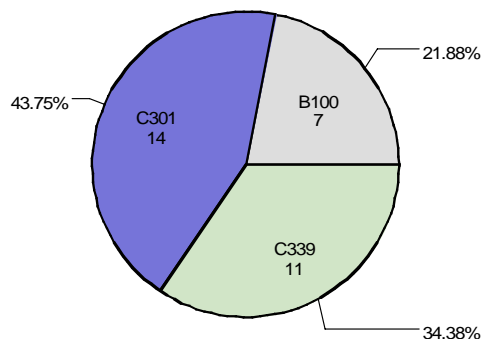
Figure16. The pie chart provides an alternative to VBAR for displaying frequencies.

```
pattern1 color=graydd;  
pattern2 color=VLIB;  
pattern3 color=PALG;  
title1 "Count of Meetings by Room";
```

```
proc gchart data=N07.meetings;  
  pie room /  
  noheading percent=arrow  
  slice=inside value=inside  
  coutline=black woutline=2;  
run; quit;
```

Pie Chart

Count of Meetings by Room



- **noheading** suppresses the automatic heading "FREQUENCY of room" that would be inserted at the top of the chart. The heading is replaced with a title.
- **percent=arrow** Prints the percent (outside) the slice with a connecting line (arrow). By default PERCENT = NONE.
- **slice=inside** Says where to place the midpoint value.
- **value=inside** Prints the value or chart statistic inside the chart. In this example, the chart statistic is the default or frequency.

Figure 17. The donut plot shows that ACCOUNTS meets most frequently in B100, MARKET in C301 and SHIP in C399.

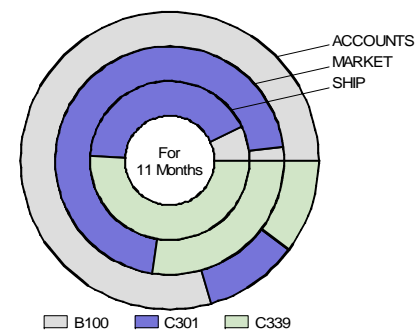
```

legend1 label=none shape=bar(4,3)
  position=(bottom center) across=3 mode=share;
pattern1 color=graydd;
pattern2 color=VLIB;
pattern3 color=PALG;
title1 "Lengths of Meetings by Room and Department";
proc gchart data=N07.meetings;
  donut room /
    noheading sumvar=hours subgroup=dept
    donutpct=30 label=('For' j=c '11 Months')
    legend=legend1 coutline=black woutline=2;
run;

```

Donut Chart

Lengths of Meetings by Room and Department

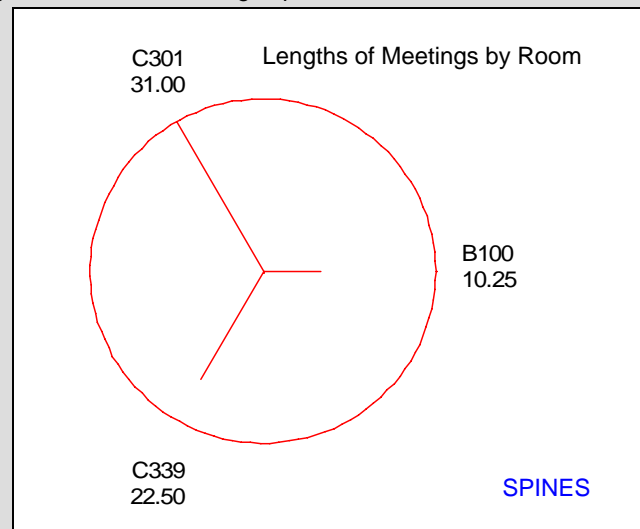
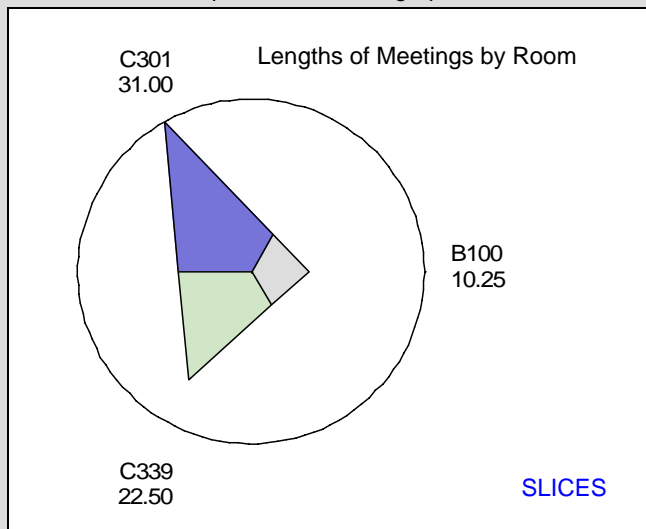


- **donutpct=30** specifies the size of the donut hole in percent. Default value is 25.
- **label=('For' j=c '11 Months')** defines the label that fits into the donut hole. *j=c* (justified=center) cause a line break to be inserted after *For*.

Enhancing a Star Chart by Overlaying with GREPLAY

The length of a spine in a star chart represents the value of the chart statistic for the midpoint value. A circle is drawn with a radius equal to the largest spine, and the origin represents the minimum value in the data (typically zero) [15], p. 573. Spines become "slices" when spine endpoints are connected. However, it is difficult to see where the spines fit in to the first panel of slices in Figure 18,

Figure 18. The star chart arranges the spines (or slices) counterclockwise around the circle midpoint value order starting at the three o'clock position. For this graph, B100 is followed by C301, and C339 brings up the rear.



To solve the problem, a composite graph places spines over slices. The placement is made with an application of the GREPLAY procedure, and the output is displayed in Figure 19.

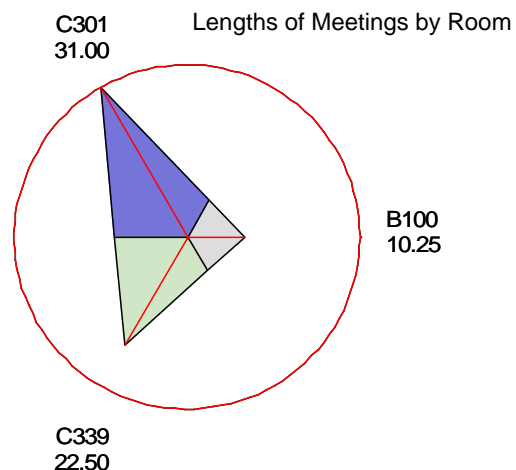
Figure 19. Spines and slices are combined to form one chart with PROC GREPLAY.

```

goptions nodisplay htext=13pt htitle=15pt;
/* SLICE STAR CHART */
pattern1 color=graydd; pattern2 color=VLIB;
pattern3 color=PALG;
title1 "Lengths of Meetings by Room";
proc gchart data=N07.meetings;
  star room / noheading name="starA"
  sumvar=hours fill=s coutline=black woutline=1;
run; quit;
/* SPINE STAR CHART */
proc gchart data=N07.meetings;
  star room / noheading name="starB"
  sumvar=hours fill=none noconnect
  coutline=red woutline=1;
run; quit;

filename chrt34 'c:\N07\ChartBasics\Paper\EMF\chrt34.emf';
goptions display gsfname=chrt34 gsfname=replace;
proc greplay nofs igout=work.gseg tc=tempCat;
  tdef T1X1
  1/llx=0 ulx=0 urx=100 lrx=100
  lly=0 uly=100 ury=100 lry=0;
  template=T1X1;
  treplay 1:StarA 1:StarB;
run; quit;

```

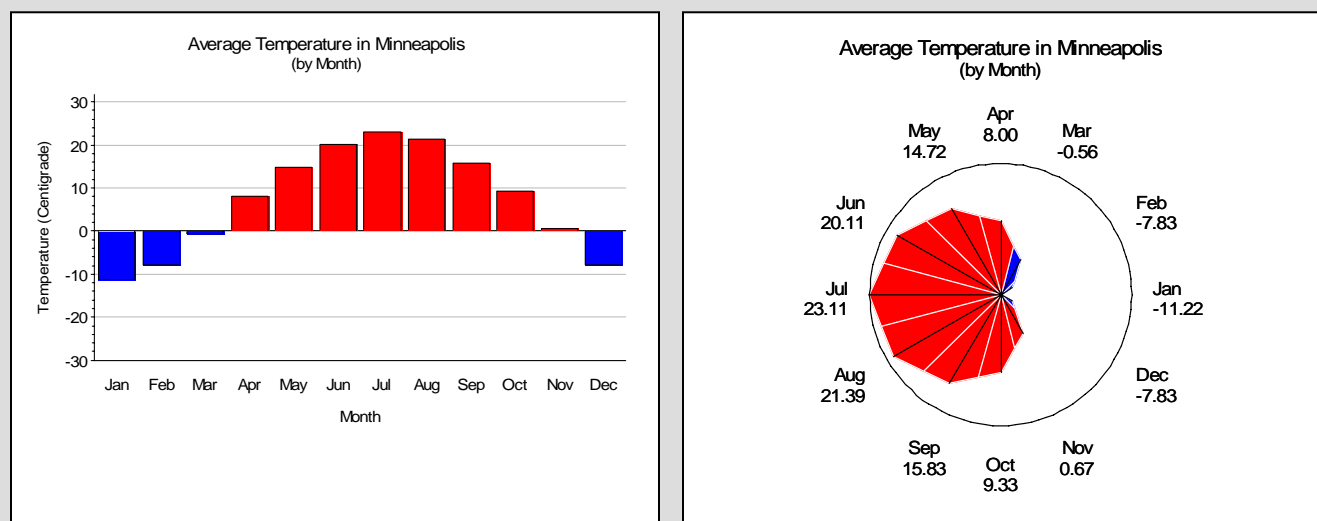


SAS/GRAPH generated titles have been cropped in WORD so that the star charts could be enlarged.

- **goptions nodisplay** hides the graphics output which not needed until GREPLAY is invoked.
- **name="starA"** ... names the catalog entry stored in a WORK graphics catalog, TEMPCAT. The slice style is set with **fill=s**(olid).
- **name="starB"** names the catalog entry stored in a WORK graphics catalog, TEMPCAT. The spine style is set with **noconnect** and to accommodate the overlay, **fill=none**.
- **proc greplay** "replays" graphs stored in graphs through templates defined in the procedure. The template, T1X1 has 1 panel that covers the entire graphics output area. Both graphs are played in order through the same panel with **treplay** (template-replay).

The star charts in Figures 18 and 19 don't look much like stars, because they have only three spines. Therefore, an additional star chart is presented without commentary in Figure 20 for the temperature data in Minneapolis where the minimum value is less than zero.

Figure 12. Bar and Star Charts for the Minneapolis monthly temperature data.



Summary and Conclusions

A tutorial for the GCHART procedure has been presented in this paper. While vertical bar charts have been covered in depth, horizontal bar charts, pie charts, donut charts and star charts have also been described to underscore the breadth of the procedure. Every attempt has been made to minimize complexity without compromising versatility. Thus, midpoint, subgroup and group charts have been presented that require no more than 15 lines of code for their construction. In addition, efforts have been made to show how text can be manipulated to improve a chart's appearance. Instructions have also been provided for generating different types of bar charts and for reordering bars in a graph.

Perhaps GCHART should be reigned in by hard-wiring midpoints as DISCRETE. Continuous data are best displayed as histograms, and histogram construction is the purview of PROC UNIVARIATE.

Copyright Statement

The paper, *Charting the Basics with PROC GCHART*, along with all associated files on the NESUG CD is protected by copyright law. This means if you would like to use part or all of the original ideas or text from these documents in a publication where no monetary profit is to be gained, you are welcome to do so. All you need to do is to cite the paper in your reference section. For ALL uses that result in corporate or individual profit, written permission must be obtained from the author. Conditions for usage have been modified from <http://www.whatiscopyright.org>.

References

- [1] Cassidy, Deb. *An Introduction to SAS/Graph®*. Proceedings of the SAS Global Forum, Cary, NC: SAS Institute Inc., 2007, paper #112.
- [2] Cleveland, William S. *The Elements of Graphing Data: Revised Edition*. Summit, NJ: Hobart Press, 1994.
- [3] Cochran, Ben. *A Gentle Introduction to SAS/GRAPH® Software*. Proceedings of the Twenty-Eighth SAS User Group International Conference, Cary, NC: SAS Institute Inc., 2003, paper #200.
- [4] Miron, Thomas. *The How-To Book for SAS/GRAPH Software*. Cary, NC: SAS Institute Inc., 1995. Copyright 1995, SAS Institute Inc., Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute Inc., Cary, NC
- [5] Tufte, Edward R. *The Visual Display of Quantitative Information: Second Edition*. Cheshire, CT: Graphics Press, 2001.
- [6] Watts, Perry. *Building a Better Bar Chart with SAS/GRAPH® Software*. Proceedings of the 20th Annual North-east SAS Users Group Conference. Baltimore, MD, 2007, paper #NP16.

Web Citations:

- [7] http://en.wikipedia.org/wiki/Bar_chart. *Bar chart: From Wikipedia, the free encyclopedia*. A definition and history of the bar chart is provided.
- [8] <http://en.wikipedia.org/wiki/Histogram>. *Histogram: From Wikipedia, the free encyclopedia*. The histogram is defined and compared to a bar chart.
- [9] http://en.wikipedia.org/wiki/Radar_chart. *Radar Chart: From Wikipedia, the free encyclopedia*. A radar chart also known as a star chart or spider chart is defined by example.
- [10] http://www.sapdesignguild.org/resources/diagram_guidelines/charts_bk.html#SaeulD. *Chart Types and their Uses*. Provides an overview and lists uses for bar charts, segmented bar charts, histograms and pie charts.
- [11] http://www.skymark.com/resources/tools/bar_charts.asp. *Bar Charts*. Properties and caveats associated with bar charts are described.
- [12] <http://www.skymark.com/resources/tools/histograms.asp>. *Histograms*. Properties and caveats associated with histograms are described.
- [13] http://www.skymark.com/resources/tools/pie_charts.asp. *Pie Charts*. Properties and caveats associated with pie charts are described.

SAS Institute References:

- [14] SAS Institute Inc. *SAS/GRAPH® Reference, Version 8*, Cary NC: SAS Institute Inc., 1999.
- [15] SAS Institute Inc. *SAS/GRAPH® Reference, Volumes 1, 2, and 3*, Cary NC: SAS Institute Inc., 2004.
- [16] TS-DOC:TS-674. *An Introduction to Exporting SAS/GRAPH Output to Microsoft Office SAS Release 8.2 and higher*. <http://support.sas.com/techsup/technote/ts674/ts674.html>.

Listings for the *Meetings* and *MplsWeather* Data Sets

| ----- Meetings Data Set ----- | | | | | | ----- MplsWeather Data Set ----- | | | |
|-------------------------------|----------|------|----------------|------------|-------|----------------------------------|------|--------|------------------|
| Obs | Dept | Room | Room Number | Date | Hours | Month | TF | TC | Above ZeroYvN |
| 1 | ACCOUNTS | C339 | 339 | 01/10/1995 | 1.00 | 1 | 11.8 | -11.22 | N |
| 2 | ACCOUNTS | B100 | 100 | 01/24/1995 | 0.50 | 2 | 17.9 | -7.83 | N |
| 3 | SHIP | C339 | 339 | 01/30/1995 | 2.00 | 3 | 31.0 | -0.56 | N |
| 4 | MARKET | C301 | 301 | 02/24/1995 | 3.50 | 4 | 46.4 | 8.00 | Y |
| 5 | SHIP | C339 | 339 | 02/28/1995 | 4.00 | 5 | 58.5 | 14.72 | Y |
| 6 | MARKET | C301 | 301 | 03/01/1995 | 4.00 | 6 | 68.2 | 20.11 | Y |
| 7 | ACCOUNTS | B100 | 100 | 03/03/1995 | 3.50 | 7 | 73.6 | 23.11 | Y |
| 8 | ACCOUNTS | B100 | 100 | 03/08/1995 | 0.50 | 8 | 70.5 | 21.39 | Y |
| 9 | ACCOUNTS | B100 | 100 | 03/21/1995 | 0.25 | 9 | 60.5 | 15.83 | Y |
| 10 | SHIP | C301 | 301 | 03/27/1995 | 1.50 | 10 | 48.8 | 9.33 | Y |
| 11 | ACCOUNTS | C301 | 301 | 03/29/1995 | 0.50 | 11 | 33.2 | 0.67 | Y |
| 12 | SHIP | C339 | 339 | 04/12/1995 | 0.50 | 12 | 17.9 | -7.83 | N |
| 13 | MARKET | C301 | 301 | 04/25/1995 | 1.50 | | | | |
| 14 | MARKET | B100 | 100 | 05/02/1995 | 0.50 | | | | |
| 15 | SHIP | C301 | 301 | 05/12/1995 | 2.50 | | | | |
| 16 | MARKET | C301 | 301 | 05/25/1995 | 3.50 | | | | |
| 17 | SHIP | C301 | 301 | 06/01/1995 | 2.00 | | | | |
| 18 | SHIP | C339 | 339 | 06/07/1995 | 3.00 | | | | |
| 19 | SHIP | C301 | 301 | 06/14/1995 | 2.00 | | | | |
| 20 | MARKET | C339 | 339 | 07/12/1995 | 0.50 | | | | |
| 21 | ACCOUNTS | B100 | 100 | 08/03/1995 | 3.00 | | | | |
| 22 | SHIP | C301 | 301 | 08/14/1995 | 2.50 | | | | |
| 23 | SHIP | C339 | 339 | 08/28/1995 | 1.50 | | | | |
| 24 | MARKET | C339 | 339 | 09/15/1995 | 3.00 | | | | |
| 25 | ACCOUNTS | C301 | 301 | 09/28/1995 | 0.50 | | | | |
| 26 | MARKET | C301 | 301 | 10/11/1995 | 3.00 | | | | |
| 27 | SHIP | C301 | 301 | 11/01/1995 | 1.50 | | | | |
| 28 | MARKET | C339 | 339 | 11/15/1995 | 1.00 | | | | |
| 29 | SHIP | B100 | 100 | 11/21/1995 | 2.00 | | | | |
| 30 | SHIP | C339 | 339 | 11/22/1995 | 3.50 | | | | |
| 31 | MARKET | C339 | 339 | 12/05/1995 | 2.50 | | | | |
| 32 | MARKET | C301 | 301 | 12/21/1995 | 2.50 | | | | |

Trademark Citation

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 registered trademarks or trademarks of their respective companies.

Contact Information

The author welcomes feedback via email at perryWatts@comcast.net