

Move=(0,+5): Making SAS/GRAPH® Work For You

Deb Cassidy

I've often been asked "Can SAS/GRAPH® do...?" SAS/GRAPH can do almost anything - if you are willing to take the time. This presentation will walk you through some graphs and show you step-by-step how to enhance your graphics. In some cases, there are multiple ways to get the same effect. I'll also include a few things that do not work. This presentation only uses options available in statements or PROCs. You can also use annotate datasets, the Graphics Editor and Data Step Graphics Interface to enhance your graphs.

The following created a basic graph in BASE SAS®.

```
PROC PLOT DATA=MWSUG;  
  PLOT ATTEND*CONF_NO;  
  
TITLE1 'SAS USERS ATTENDING MWSUG';  
FOOTNOTE 'NOT ACTUAL DATA '  
  'MWSUG--SEPT 93';
```

Would this graph convince you the MidWest conference was worth attending? How about the graph enhanced using SAS/GRAPH?

GOPTIONS

SAS/GRAPH lets you change many options. Before you begin a graph, make sure you know how options are set for your device. You will need to be concerned with both the viewing device and printing device. The one option everyone must specify is DEVICE=. SAS prompts you if you don't tell it the device. You may not need any other options or you may need several. Some sites have created macros to supply all the options for a device so individuals do not need to enter them. You can override options with another GOPTIONS statement. You can also override some within the statements for your graph. To know what options your device has as defaults, run

```
GOPTIONS DEVICE= your device;  
PROC GOPTIONS;
```

If you want to know what statements are in effect at a specific point in your program, you should specify the statement(s) you want to see. For example:

```
PROC GOPTIONS SYMBOL AXIS NOLIST;
```

will give you the SYMBOL and AXIS statements while suppressing the list of graphics options.

Options stay in effect until you reset them or end your SAS session (for batch and non-interactive, each job would be a session). Any options listed before a RUN or a PROC will affect the graph. RUN or PROC tells SAS to go ahead and create the graph. For example, if you submit the following as one program you will get the color I indicated in the title.

```
GOPTIONS CTEXT=RED;  
PROC GSLIDE;  
  TITLE1 'THIS IS RED';  
RUN;
```

```
GOPTIONS CTEXT=GREEN;  
PROC GSLIDE;  
  TITLE1 'NOT GREEN BUT BLUE';  
GOPTIONS CTEXT=BLUE;  
RUN;
```

```
GOPTIONS CTEXT=GREEN;  
PROC GSLIDE;  
  TITLE1 'THIS IS GREEN';  
PROC GSLIDE;  
  TITLE1 'THIS IS BLUE';  
GOPTIONS CTEXT=BLUE;  
RUN;
```

I suggest always putting your options before a graph and using RUN after each graph to make sure you get the correct options. However, too many RUN statements can cause problems. If you are getting multiple copies of a graph, look for duplicate RUN statements. This is most likely to happen if you have a RUN within a macro and a RUN after the macro. If you rely on PROC statements to signal the end of a graph, you may need a RUN statement after the last graph.

Table 1 lists several options which will affect the appearance of the graphs. COLORS=(color list) is the one of the most likely to cause problems when you use multiple devices. Let's say you decided to use all system defaults for your bar chart with 4 bars. Your color monitor has 4 colors as a default (blue,red,green,cyan) and you get 4 bars with a cross-hatch (X1). You print it on your black-and-white printer expecting the same cross-hatch with black and shades of gray. Instead you get everything in black with different patterns. When you don't specify color for patterns and symbols, SAS picks the first pattern and cycles through all the colors before it moves on to the next pattern. On the color monitor you get blue X1, red X1, green X1, and cyan X1. A fifth bar would have resulted in blue with the X2 pattern. The printer only has 1 color -black. (This assumes you submitted the job again and routed it to the printer. If you do a screen print, you will get different results.) The two easiest solutions are to always specify a color on symbol and pattern statements or only use black on your monitor.

If you specify any color that the device doesn't use, it will automatically be changed for you.

SAS/GRAPH works from cells like graph paper. You use HPOS and VPOS to specify the number of cells. The higher the number, the more cells although each is smaller. HPOS is the number of columns and VPOS is the number of rows. For my device, I want to have 80 x 32 cells. Since I will be printing on a black-and-white printer and I don't care about seeing the colors on the monitor, I'll restrict my color list. I also want all my text to use the SWISS font except as noted.

```
GOPTIONS HPOS=80 VPOS=32
COLORS=(BLACK) FTEXT=SWISS;
```

ENHANCED LINE GRAPH

The following code creates the final enhanced graph shown. The first step should be to run the graph without any enhancements to make sure you have the basic graph correct. Then you should work through each enhancement. As you become more familiar with the options, you can put more in each run. However, sometimes you will not be able to use some

options with others. For example, if you titles and footnotes are too tall, you will not get you graph. You may need to change all the heights and widths I've used if you have a different HPOS and VPOS.

```
PROC Gplot DATA=MWSUG;
  PLOT ATTEND*CONF_NO /
    VAXIS=AXIS1 HAXIS=AXIS2
    VMINOR=1 HMINOR=0;

  AXIS1 ORDER=(200 TO 550 BY 50)
    LABEL=(A=90 'ATTENDANCE')
    WIDTH=2
    OFFSET=(2 PCT, 2.2 PCT);
  AXIS2 VALUE=
    (TICK=1 J=C 'MINNEAPOLIS'
      J=C '1990'
    TICK=2 J=C 'CHICAGO'
      J=C '1991'
    TICK=3 J=C 'KANSAS CITY'
      J=C '1992'
    TICK=4 H=1.5
      J=C 'INDIANAPOLIS'
      J=C '1993')
    LABEL=(H=1.5 'CONFERENCE')
    WIDTH=2 OFFSET=(3 PCT);
  SYMBOL1 H=3 FONT=, VALUE=DIAMOND
    I=JOIN WIDTH=2;
  TITLE1 'SAS' MOVE=(+0, +1 PCT)
    F=SPECIAL 'R'
    MOVE=(+0,-1 PCT) F=SWISS
    'USERS ATTENDING'
    F=CENTBI 'MWSUG';
  TITLE2 H=3.3 A=-90 ' ';
  NOTE MOVE=(71 PCT, 50 PCT)
    'FUTURE CONFERENCES'
    DRAW=(+0 PCT, -1 PCT,
      71 PCT, +0 PCT)
    MOVE=(71 PCT, 46 PCT)
    '1994 - OMAHA'
    MOVE=(71 PCT, 43 PCT)
    '1995 - SOME CITY';
  NOTE MOVE=(84, 78.5)PCT
    BOX=2 H=.5 F=DUPLICATE
    'EXPECTED';
  FOOTNOTE H=.5 J=L 'FICTITIOUS DATA'
    J=R 'MWSUG - - SEPT 93';
RUN;
```

The plot shows the attendance on the Y-axis (vertical) and the conference number on the X-axis (horizontal).

VAXIS=AXIS1 says to create the vertical axis using the instructions in the AXIS1 statement. HAXIS=AXIS2 says to create the horizontal axis using the instructions in the AXIS2 statement. You do not have to use a specific axis number for vertical or horizontal. Some graphs have 2 horizontal or vertical axes.

AXIS numbers can be 1 to 99 although most people start with 1.

AXIS1 says to start at 200 and go to 550 by 50 unit increments. In the example, one value is at 505. If you try to use ORDER=(200 to 500 by 50), you will lose that point. You could also use ORDER=(200 TO 500 BY 50, 505) which says to put a tick mark at 505 also. However, the distance between 500 and 505 will be the same as between 450 and 500.

LABEL=(A=90 'ATTENDANCE') says to use ATTENDANCE instead of the variable name. A=90 angles the entire label from the horizontal line. You may need to use A=90 and R=0 together. R= says to ROTATE each letter. In this case, you are over-riding any default rotation. I've usually had this problem if there is already a label for the variable in my dataset.

WIDTH=2 makes the axis itself wider. If you have a data point in the corner of the graph, you may want to use OFFSET= to give it more room. The first value moves the first major tick mark away from the intersection of the axes. The second value moves the last major tick mark away from the end of the axis. You can use one without the other. OFFSET only affects the axis specified.

The AXIS2 statement uses some of the same options as AXIS1. I've used the default order although I could have specified (1 to 4) or (1 to 4 by 1). I've also labeled the major tick marks. You can use a format to label the tick marks but you have fewer options. I wanted to create a 2-line label. I also wanted to increase the height for the current site. This method lets me customize each tick mark. To create multiple lines, you must center each section of the label. This also works for X-axis labels. SAS Institute is aware that it doesn't always work for Y-axis labels.

The plot statement has two other options - VMINOR=1 and HMINOR=0. These statement indicate how many minor tick marks to include. On the vertical axis I want 1 smaller tick mark between each major one. In this example, that provides marks for every 25 units. Since my conference numbers are discrete, I want to suppress minor tick marks

on the horizontal axis. You could also have used MINOR=NONE in the AXIS2 statement to delete them. You also have the option of customizing minor tick marks.

SYMBOL1 describes how to put symbols on the points. By default, you get a symbol on each point and the points are not joined. I want the symbols to have HEIGHT=3. I want to INTERPOLATE the points between the actual points by simply JOINING the points. There are many other options for INTERPOLATE. The line joining the points will have a WIDTH of 2. I also want to specify the symbol. There are special symbols that are only used in a symbol statement. You can also use any character by specifying a font and placing the letter in quotes. For example:

FONT=SPECIAL VALUE='R'

would give the registered trademark symbol on the graph. If you don't specify the font, SAS looks for the specified value as one of the special symbols. If it isn't used as a special symbol, you get the actual value specified. By using VALUE=DIAMOND, I get a diamond as the symbol. If there is a font specified and I use VALUE=DIAMOND, I actually get the word DIAMOND. Symbols and patterns will keep any option you used in an earlier statement with the same number. If I had an earlier SYMBOL1 FONT=DUPLEX statement, I would get the word DIAMOND in duplex. The FONT=, included in this symbol statement makes sure I get the special symbol. If this sounds confusing, try a few examples or use PROC GOPTIONS SYMBOL NOLIST; to see what SAS "sees".

The first title statement shows how I named my talk. I want the registered trademark symbol to be superscript. MOVE= tell SAS to go to a different point before writing the next section of text. A + or - indicates this point is relative to where the last text was written. No sign indicates to go to that point regardless of the location of the current text. The first value indicates the horizontal direction and the second value the vertical direction. I want to slide the character up but keep it in the default horizontal position. The +0 indicates to keep the default. I want to move the text up 1 percent. After I finish with the character, I

want to move the next text back to the original line so I use -1 percent. Don't forget to include the blank spaces in one of the strings. I also used different fonts. If you want to move the first string and use a relative position, you must use TITLE ' ' MOVE= ...;

TITLE2 is a blank title so why did I include it? By using A=-90, I moved the title to the right-hand side of the graph. This makes the graph smaller and gives me a larger right margin. The NOTE statement will then appear in that margin without overwriting any of the graph. The HEIGHT= determines how much additional space I have.

NOTE statements may be one of the most useful and least used statements in SAS/GRAPH. In some cases, NOTES can let you avoid creating an ANNOTATE dataset. By default, NOTES are written in the procedure output area. They are left-justified and placed at the top of the area. They will overwrite your graph instead of reducing the size of the graph like TITLE and FOOTNOTE do. You can have any number of NOTES and they do not have a number. I rarely use NOTE without a MOVE= to position it exactly where I want it.

I have only 1 note statement but it prints 3 lines of text and draws a line. The MOVE= works like it did in the TITLE statement but it uses absolute values. The tricky part is drawing the line. The line is actually drawn from right to left. To draw a line, you need to specify the ending point. You know the left hand point is just below the starting point of your text - a point you specified. However, you don't know the ending point of your text. By starting relative to the end of the text, you only need to specify the X value for the left end of the line.

If you want to box the entire note, you can only use relative points in your MOVE. Finding the correct coordinates can be frustrating if you have more than a couple lines. I recommend using an ANNOTATE dataset for that case.

The second note does use a BOX= option. The value specifies the width of the lines for the box. This note actually appears on the graph.

I've also reduced the HEIGHT of the text and changed the font.

The footnote statement left justifies part of the text and right justifies the rest. The default is centered text.

BASIC BAR CHART

The next graph requires data manipulation before creating the graph. In base SAS you would use the following code.

```
PROC FORMAT;
  VALUE DRIVE 0='NO'
              1='YES';
  VALUE CITY 1='MINNEAPOLIS'
             2='CHICAGO'
             3='KANSAS CITY'
             4='INDIANAPOLIS';

PROC CHART;
  VBAR DRIVE / GROUP=CONF_NO
  DISCRETE TYPE=PERCENT G100 ;
  FORMAT DRIVE DRIVE. CONF_NO CITY.;
  TITLE
  '%DRIVING/NOT DRIVING TO MWSUG ';
```

The formats are used instead of the actual value. GROUP= is used to get separate bars for each group but still show them on the same graph. If you used a BY CONF_NO statement, you would get 4 graphs. The default is to use the midpoints for CONF_NO but DISCRETE is used to override the default and use the actual values. TYPE= says to compute percentages. G100 indicates each group should total 100%. The default is for the total of all bars to be 100%.

ENHANCED BAR CHART

This chart shows the percentage for both "YES" and "NO". What if you really just want the percentage that replied "YES"? Since PROC CHART and PROC GCHART do the percentage calculations, you can't just delete the "NO" observations. To get around this problem and still show percentages instead of frequencies or sum, you must modify the data first. The variable PERCENT is renamed to avoid confusion later.

```
PROC FREQ DATA=MWSUG2;
  BY CONF_NO;
  TABLES DRIVE/NOPRINT OUT=DRIVEPCT
  (RENAME=(PERCENT=DRV_PCT));
```

Use this dataset for your PROC GCHART as shown.

```
PROC GCHART DATA=DRIVEPCT
  (WHERE=(DRIVE=1));
  VBAR DRIVE / GROUP=CONF_NO
    SUMVAR=DRV_PCT
    SUM DISCRETE
    MAXIS=AXIS1
    GAXIS=AXIS2
    RAXIS=AXIS3
    VMINOR=1 SPACE=7;

  AXIS1 LABEL=NONE VALUE=NONE;
  AXIS2 MINOR=NONE LABEL=NONE;
  AXIS3 ORDER=(0 TO 100 BY 10)
    LABEL=(A=90 J=C 'PERCENTAGE'
      J=C 'BY SITE');
  FORMAT CONF_NO CITY. DRV_PCT 3.;
  TITLE 'PERCENTAGE DRIVING TO MWSUG';
  FOOTNOTE H=.75 J=L 'FICTITIOUS DATA'
    J=R 'MWSUG -- SEPT 93';
```

The WHERE= on the DATA= restricts the graph to "YES" observations. The SUMVAR=PERCENT will "sum" the percentages. Since there is only one observation for each conference, you will get the value in the percentage dataset. The SUM option prints the value for the sum over the bar. The FORMAT statement prints the value without decimal places. If you want to have a percent sign with the number above the bar, you have two options. You can use a percent format but this will also format the values on the axis. You would then need to specify the values to appear on the axis to suppress the % signs. The other alternative is to label the bars using an ANNOTATE dataset. You can create this dataset directly from the actual data. The PERCENT option will not work here because SAS will calculate the percentage so all bars add to 100%.

DISCRETE is used to get the actual values rather than midpoints. VMINOR=1 specifies 1 minor tick mark between the major tick marks on the Y-axis.

MAXIS refers to the midpoint point which is the variable listed in your VBAR statement. GAXIS refers to the variable in the GROUP= option. RAXIS is the response axis or the Y-axis. The midpoint axis has both labels and values suppressed. Otherwise, you would have "1" appearing under each bar. The

group axis suppresses the label for CONF_NO and eliminates the minor tick marks. If you used the option NOAXES, you would eliminate the lines for axes as well as the values and labels. The response axis is forced to 0 to 100 regardless of the actual percentages.

The response axis also attempts to create a 2-line label which doesn't quite work. One alternative is to suppress the axis label and use angled titles or footnotes to write the label.

This would be similar to the null title which added more space at the right. You just need to angle the text to the left.

I opted to use formatted values for the conference sites instead of the specifying tick marks. Unfortunately, my formatted values are too long for the axis. Sometimes SAS can automatically angle/rotate the values to fit. In this case, it couldn't. One option is to angle the values yourself which can be done by specifying the tick marks. Example:

```
TICK=1 A=75 'MINNEAPOLIS'
```

The other option is to use the SPACE= option on your VBAR statement. This option provides more space between bars. (GSPACE will provide more space between groups of bars.) The value is given in character cells. You can also use SPACE=0 to eliminate the space.

OTHER EXAMPLES

Many times the secret to getting the graph is manipulating the data first. For example, let's say I wanted a star instead of a diamond for the Indianapolis point on the first line graph but you still want to join the points. To do this, create a new dataset with a flag variable. Assign a '1' to the Indianapolis observation and '2' to the other observations. Duplicate all the observations and assign a '3'. Use the following symbol statements.

```
SYMBOL1 VALUE=STAR I=NONE';
SYMBOL2 VALUE=DIAMOND I=NONE;
SYMBOL3 VALUE=NONE I=JOIN;
```

The first symbol only prints a star at the point and the second only print a diamond. The third only joins the points and does not print a

symbol on the points. Your plot statement is now

```
PLOT ATTEND*CONF_NO = FLAG;
```

Each value of FLAG gets a different symbol. The smallest value of flag will get the first symbol, the second smallest gets the second symbol, etc.

I wrote the following code as an easy way to see some other options and provide more information on defaults. Compare the results of the 3 GSLIDEs. The manual explains the options in detail.

```
PROC GSLIDE;
TITLE1 'TITLE 1 AT DEFAULT';
TITLE2 'TITLE 2 AT DEFAULT';
TITLE3 'TITLE 3 AT DEFAULT';
NOTE 'NOTE AT DEFAULT';
FOOTNOTE1 'FOOTNOTE 1 AT DEFAULT';
FOOTNOTE2 'FOOTNOTE 2 AT DEFAULT';
FOOTNOTE3 'FOOTNOTE 3 AT DEFAULT';
RUN;

PROC GSLIDE;
TITLE1 'TITLE 1 AT DEFAULT';
TITLE2 LSPACE=10
      'TITLE 2 WITH LSPACE=10';
TITLE3 'TITLE 3 AT DEFAULT';
NOTE 'NOTE AT DEFAULT';
FOOTNOTE1 'FOOTNOTE 1 AT DEFAULT';
FOOTNOTE2 LSPACE=5
      'FOOTNOTE 2 WITH LSPACE=5';
FOOTNOTE3 'FOOTNOTE 3 AT DEFAULT';
RUN;

PROC GSLIDE;
TITLE 1 'TITLE 1 AT DEFAULT';
TITLE 3
      'TITLE 3 AT DEF. WITHOUT TITLE 2';
NOTE 'NOTE AT DEFAULT';
FOOTNOTE1 'FOOTNOTE 1 AT DEFAULT';
FOOTNOTE2 'FOOTNOTE 2 AT DEFAULT';
FOOTNOTE3 A=90 'ARE YOU AWAKE?';
MOVE=(50 PCT, 20 PCT) 'FOOTNOTE 3';
FOOTNOTE4 U=1 'UNDERLINING'
      LANGLE=90 'FOOTNOTE 4';
FOOTNOTE5 J=R U=1 'UNDERLINING'
      ANGLE=90 'FOOTNOTE 5';
RUN;
```

A HELPFUL GRID

You think MOVE= will be helpful but how do know what values to use without a lot of guesswork? The following code will generate a grid which can be placed over you graph to help determine coordinates. This grid is based

on percentage of the graphics output area. You might also want a grid which shows your cells based on HPOS and VPOS. You will need to change the DO loop accordingly. For each line you want to draw, your annotate dataset needs 1 observation that moves to one end of the line. A second observation is needed that will draw to the other end of the line.

The grid is displayed using PROC GANNO. I've saved the grid in a catalog and specified a name.

```
DATA ANNO;
/* X AND Y VALUES BASED ON
  PERCENTAGE OF THE GRAPHICS
  OUTPUT AREA */
RETAIN XSYS '3' YSYS '3';
DO I=0 TO 100 BY 10;
/* DRAW VERTICAL LINES */
X=I; Y=0; FUNCTION='MOVE';
OUTPUT;
Y=100; FUNCTION='DRAW';
OUTPUT;
/* DRAW HORIZONTAL LINES */
Y=I; X=0; FUNCTION='MOVE';
OUTPUT;
X=100; FUNCTION='DRAW';
OUTPUT;
END;

PROC GANNO ANNO=ANNO
GOUT=MWSUG.PLOTS NAME=GRID1;
RUN;
```

CONVERTING LISTINGS TO GRAPHS

The plot and chart from base SAS were converted to graphs to replayed. You can convert output or log to a graph although you may have to experiment with HPOS and VPOS to get it legible. It still may not look as nice. You must first save the section of output you want as a file. How to do this varies by operating system.

You must use a FILENAME statement to tell SAS where the file is. The following example is from CMS. You may want to use a GOUT= and NAME= option to save your graph to a specific catalog and to supply a name. The default name will be GPRINT with any subsequent graph in the same catalog being GPRINT1, GPRINT2, etc.

```
FILENAME MYLIST 'MWSUG1 LISTING A';
```

```
PROC GPRINT FILEREF=MYLIST;
```

You can add TITLE, NOTE, and FOOTNOTE statements or use an ANNOTATE dataset.

You can also specify the color of the text.

OTHER COMMENTS

The values needed for height and width will vary based on HPOS and VPOS. PERCENT is the only value that will be consistent across devices and options because it is proportional. Even inches and centimeters may not work if you go to a smaller device or change options.

Since it is impossible to cover all the options, you should use the manuals for additional detail. The programs for examples in the manual are included in a library provided with the system. Contact your site representative for details on how to access this library. The manual indicates the program name. These programs can help you learn how your device(s) work (and differ) as well as how to write the program.

At times, using SAS/GRAPH is trial and error. But remember, what didn't work this time may be the solution to your next graph. You may want to keep a notebook for both your successes and failures. Include the code, log and the graph.

SAS and SAS/GRAPH are registered trademarks of SAS Institute Inc. in the USA and other countries.

Contact:

Deb Cassidy

3809 Secretariat Court

Columbus, OH 43221

(614) 771-7861

(leave message on answering machine)

DATA FOR THE GRAPHS

```
DATA MWSUG;
```

```
  INPUT CONF_NO ATTEND;
```

```
CARDS;
```

```
1 200
```

```
2 500
```

```
3 450
```

```
4 501
```

```
;
```

```
DATA MWSUG2;
```

```
  INPUT CONF_NO DRIVE @@;
```

```
CARDS;
```

```
1 0 1 0 1 1 1 1
```

```
2 0 2 0 2 1 2 1 2 1
```

```
3 0 3 0 3 0 3 1
```

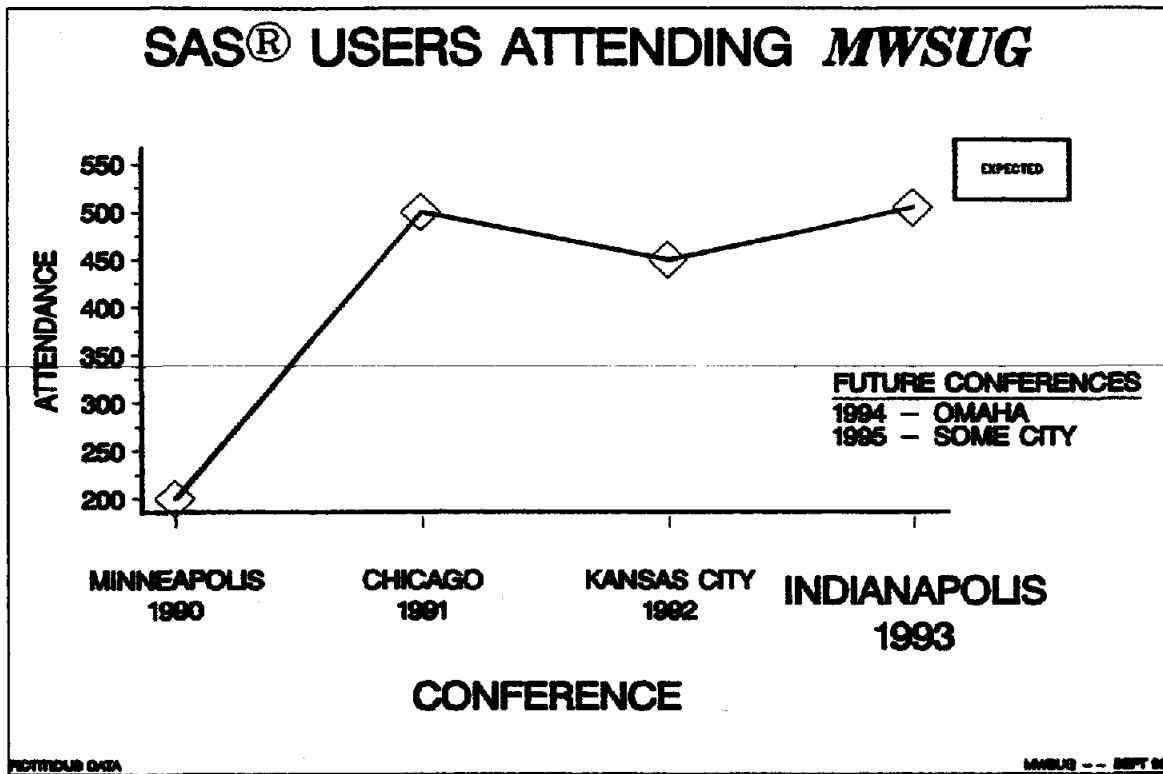
```
4 0 4 1 4 1 4 1
```

```
;
```

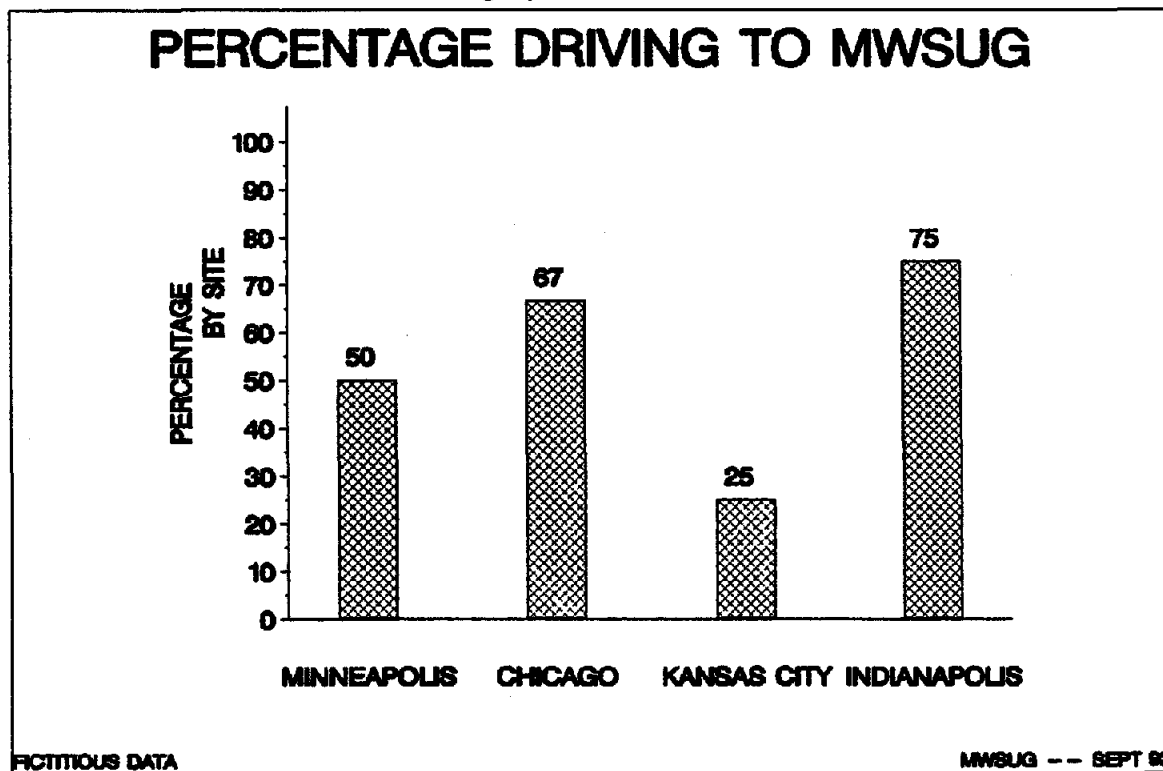
TABLE 1
PARTIAL LIST OF SAS/GRAPH OPTIONS

<u>OPTION</u>	<u>ACTION</u>	<u>DEFAULT/COMMENTS</u>
COLORS=(color list)	Sets default list of colors for text, symbols, etc	Default varies by device. You can override the colors in your statements or with the color options listed below.
CTEXT=color	Specifies the color for all text including titles, labels, etc.	Default is first color in COLORS= list.
CTITLE=	Specifies the color for all titles, footnotes, notes, and border.	Default is CTEXT color .
CBY=	Specifies the color for BY lines.	Default is CTEXT color.
CPATTERN=	Specifies the color for patterns.	Each pattern cycles through the COLORS=list before going to the next pattern.
CSYMBOL=	Specifies the color for symbols.	Each symbol cycles through the COLORS=list before going to the next symbol.
FTEXT=	Specifies the font for all text.	Default is hardware font.
FTITLE=	Specifies the font for the first title line.	Default is FTEXT=. If not specified, default is complex.
FBY=	Specifies the font for the by lines.	Default is FTEXT=, hardware font, or simulate.
GUNIT=	Specifies the unit for height.	Default is CELLS. You can specify IN, CM, PCT.
HTEXT=	Specifies the height for all text.	Default is 1 unit.
HTITLE=	Specifies the height for the first title line.	Default is 2 units.
HPOS=	Specifies the number of horizontal positions.	Default depends on device.
VPOS=	Specifies the number of vertical positions.	Default depends on device.
HORIGIN=	Horizontally offsets the lower left corner of the display area.	Default is 0.0 IN (no offset). Specify either IN or CM.
VORIGIN=	Vertically offsets the lower left corner of the display area.	Default is 0.0 IN (no offset). Specify either IN or CM.
HSIZE=	Specifies the horizontal size of the graphics area.	Default is 0.0 IN (entire display area). Specify either IN or CM.
VSIZE=	Specifies the vertical size of the graphics area.	Default is 0.0 IN (entire display area). Specify either IN or CM.
ROTATE ROTATE=	Turns the graph	ROTATE turns it from the default for the device. ROTATE=LANDSCAPE or ROTATE=PORTRAIT specifies the direction.

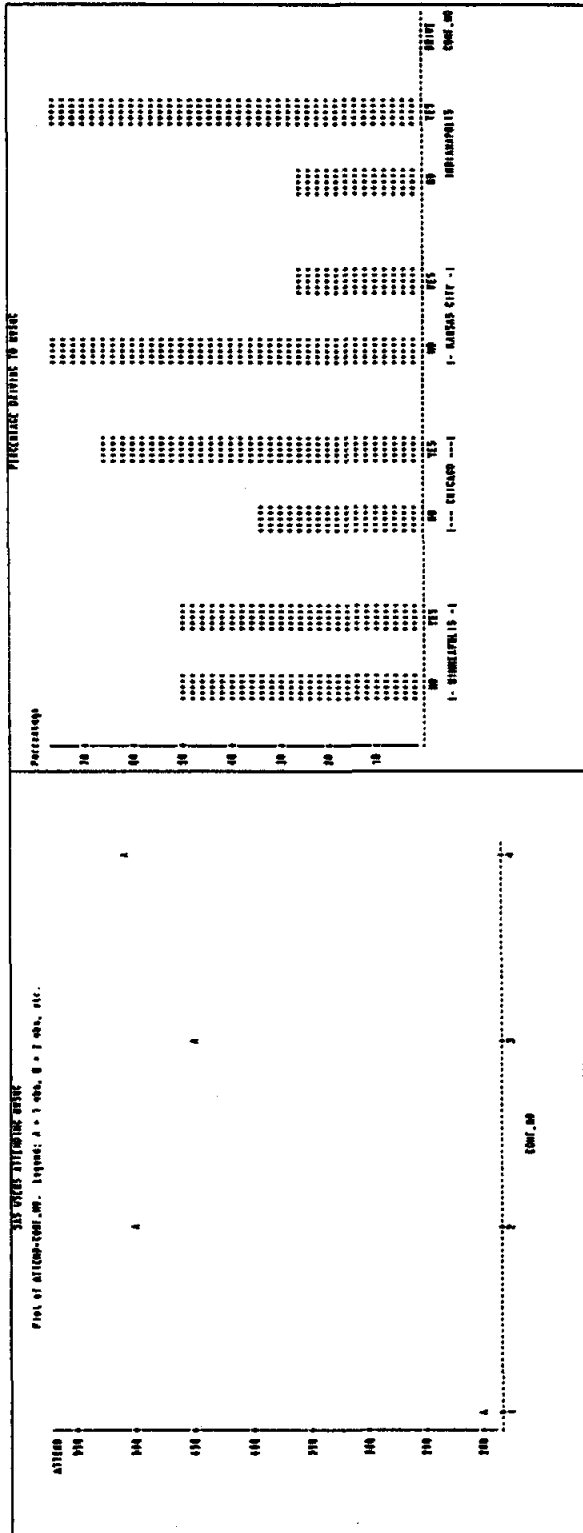
Fictitious data was used for this graph.



Fictitious data was used for this graph.



Fictitious data was used for these graphs.



NOTE AT DEFAULT TITLE 1 AT DEFAULT TITLE 2 AT DEFAULT TITLE 3 AT DEFAULT FOOTNOTE 1 AT DEFAULT FOOTNOTE 2 AT DEFAULT FOOTNOTE 3 AT DEFAULT	NOTE AT DEFAULT TITLE 1 AT DEFAULT TITLE 2 WITH LIPSPACE - 10 TITLE 3 AT DEFAULT FOOTNOTE 1 AT DEFAULT FOOTNOTE 2 WITH LIPSPACE - 10 FOOTNOTE 3 AT DEFAULT	NOTE AT DEFAULT TITLE 1 AT DEFAULT TITLE 2 AT DEFAULT WITHOUT TITLER FOOTNOTE 1 AT DEFAULT FOOTNOTE 2 AT DEFAULT FOOTNOTE 3 AT DEFAULT
---	---	--