

# SAS/GRAPH<sup>®</sup> Odds and Ends;

## Little Things That Make a Big Difference

Arthur L. Carpenter

### ABSTRACT

SAS/GRAPH is very flexible and at times it can be very frustrating. Often it seems that those little things that ought to be the easiest take the longest to 'get right'. This paper will deal with an eclectic set of topics that sometimes cause confusion and difficulties. It is not unusual that the little things, the topics that should be easy, take the longest to code, and that a little hint or a trick is what is needed.

Topics will include:

- options for text handling in titles, footnotes, and labels
- AXIS statement options to dress up a scatter plot
- SYMBOL statements options and how they are accessed
- GREPLAY and TEMPLATES to display banner titles

This entry level presentation is designed for those users of SAS/GRAPH that are able to generate a graph, but have difficulty getting it to look the way that they want it (or the way the boss wants it).

### TEXT OPTIONS IN TITLES

#### 1. Creating Subscripts and Superscripts

You can create subscripts and superscripts by changing text size and by using the MOVE= option in the TITLE, FOOTNOTE, and NOTE statements. The MOVE= option displaces the specified text from the last position that text was written (relative displacement). The units of displacement can be specified as part of the MOVE= option (valid choices are CELLS, CM, IN, or PCT), on the graphic option GUNIT=, or left as the default, which is CELLS.

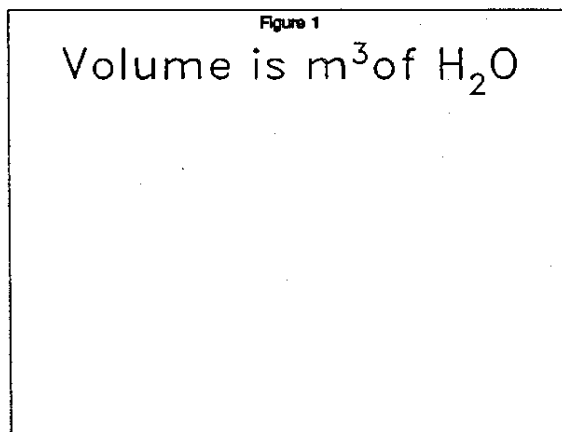
The form of the option is MOVE=(x,y), where the x is the number of horizontal units to move (displace) and y is the number of vertical units. A plus (+) or minus (-) sign precedes the x or y.

The process for creating a subscript is to change the character height, move to a new position, write the subscript, move back to position, and reset the height. In this example MOVE= is used to rewrite "Volume is m<sup>3</sup> of H<sub>2</sub>O" using both a subscript and a

superscript.

```
title1 'Figure 1';
title2 f=duplex
      h=6 'Volume is m'
      h=4 move=(+0,+2) '3'
      h=6 move=(+0,-2) 'of H'
      h=4 move=(+0,-2) '2'
      h=6 move=(+0,+2) 'O';
```

The plus (+) and minus (-) signs are used here in order to make the position changes relative to the text string just written.

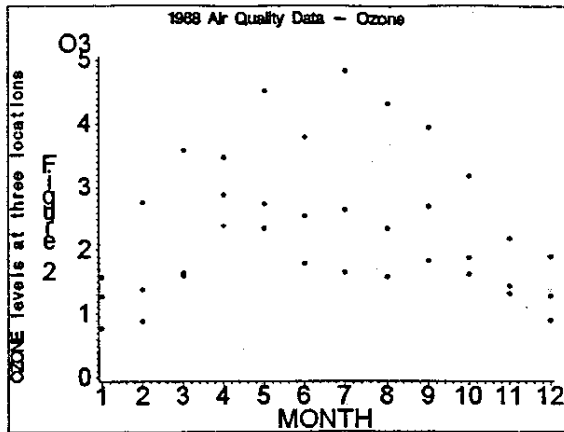


#### 2. Orientating Text in a Title

The use of the ANGLE= and ROTATE= title and footnote options in the following code produce text lines along the vertical borders of the plot.

```
goptions htext=2;
proc gplot data=voll.ca88air;
plot o3 * month;
title1 '1988 Air Quality Data - Ozone';
title2 angle=90 'OZONE levels at three
locations';
title3 angle=90 ' ';
footnote1 angle=-90 rotate=90 'Figure
2.';
run;
```

In this code TITLE and FOOTNOTE statements which utilize ANGLE= and ROTATE= options are used with a PROC GLOT. Notice that TITLE2 reads from the bottom to the top while FOOTNOTE1 reads top to bottom. TITLE3 is used to create a space between TITLE2 and FOOTNOTE1.



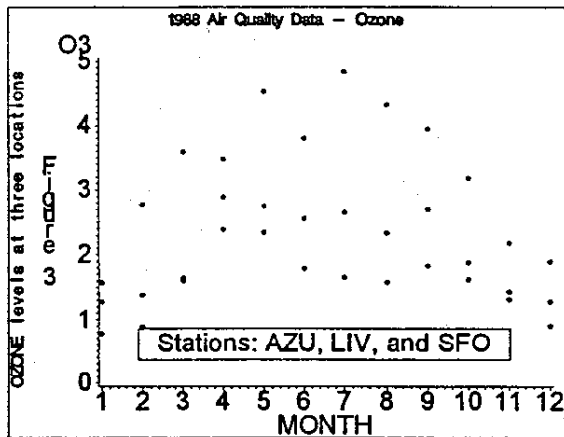
### 3. Placing Text in a Box

Legends and text information can be added to a plot by use of the BOX option in the TITLE statement. When used in conjunction with the MOVE= option the text can be placed anywhere on the graph. The following example adds a text box to the plot shown in the previous figure.

```

title1 '1988 Air Quality Data - Ozone';
title2 move=(75,20)pct box=1 blank=yes
'Stations: AZU, LIV, and SFO';
title3 angle=90 'OZONE levels at three
locations';
title4 angle=90 ' ';
footnotel angle=-90 rotate=90 'Figure
3';

```



The BLANK=YES option creates a box that blocks any symbols that may be plotted behind it.

## CHANGING TEXT IN AXES

### 4. Rotate Text in an AXIS Statement

Axis labels can also be produced using the AXIS

statement, and the ANGLE= and ROTATE= options behave in a similar fashion in this statement as they do in the TITLE and FOOTNOTE statements. The following code will produce axis labels very similar to those shown in Figures 2 and 3.

```

axis1
label = (f=simplex angle=90 ' '
j=c angle=90 h=1.1
'Ozone levels at three locations'
j=c angle=-90 rotate=90
'Figure 4' );

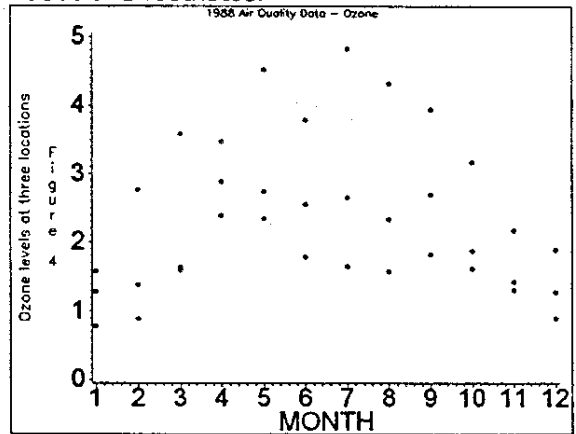
```

```

proc gplot data=voll.ca88air;
plot o3 * month / vaxis=axis1;
symbol v=dot c=black;
title h=1.5 f=simplex '1988 Air Quality
Data - Ozone';
footnotel ;
run;

```

The LABEL= option is used to define label text within the AXIS statement. Within the parentheses the options and text strings are arranged as they are for titles and footnotes.



### 5. Tick Mark Text

It is at times useful to replace the tick mark numbers on one or both of the axes with textual information. By using the AXIS statement you can define labels for each major tick mark as well as for the axis itself. The VALUE= option in the AXIS statement allows specific text strings to be associated with specific values and increased control is achieved by making use of font and size options.

Primary control in the VALUE= axis option is exercised by specifically identifying the tick mark number. Usually this implies that the user has specified the number of major tick marks e.g. (major=(n=12) or order=(1 to 12)), or knows how many tick marks there will be. The tick mark is designated with the T= option. Attributes for a specific tick mark will immediately follow the appropriate T= option. The user can independently control the font, size, and color of the text for each

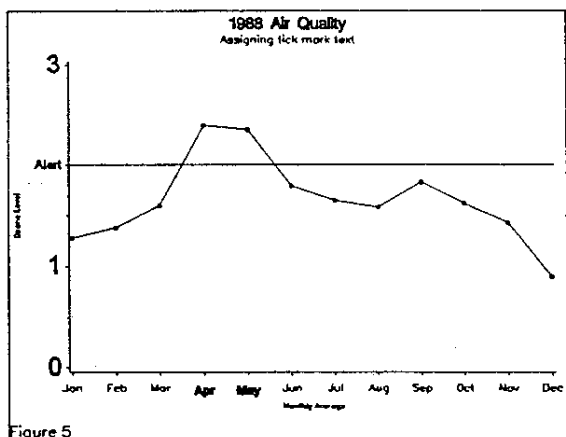
tick mark.

In the following code the first axis statement (AXIS1) uses t=3 to identify the tick mark number which will have the string 'Alert' instead of its value (which would have been 2.0). AXIS2 defines a default font and height for the axis (h=1.5 f=simplex) and also changes these two options for the fourth and fifth tick marks (Apr and May).

```
* define the vertical axis ;
axis1 label = (f=duplex h=1 a=90
'Ozone Level')
order = (0 to 3 by 1)
minor = (n=1)
value = (t=3 h=1 'Alert');

* define the horizontal axis ;
axis2 label = (f=duplex h=1 'Monthly
Average')
minor = none
value = (h=1.5 f=simplex
t=1 'Jan' t=2 'Feb' t=3 'Mar'
t=4 f=swissb 'Apr'
t=5 f=swissb 'May'
t=6 'Jun' t=7 'Jul' t=8 'Aug'
t=9 'Sep' t=10 'Oct' t=11 'Nov'
t=12 'Dec');
```

In the resulting graph we have placed the word ALERT on the vertical axis (replacing the 2) and have changed the font and size for the two months of particular interest.



The control of the characteristics of tick marks in the VALUE= statement is positionally dependent. In this example the AXIS2 statement has a fairly complicated VALUE= option. The default height (h=1.5) and font (f=simplex) is established by making the specification before any of the T= options. As soon as the first T= option is specified, other options such as H= and F= will apply to the tick mark designated by the preceding T= option. Thus, the font can be changed for T=4 and T=5 without changing the default font for the other tick

marks.

## CONTROLLING THE SCALE OF THE AXES

### 6. AXIS Dates and LEGENDS

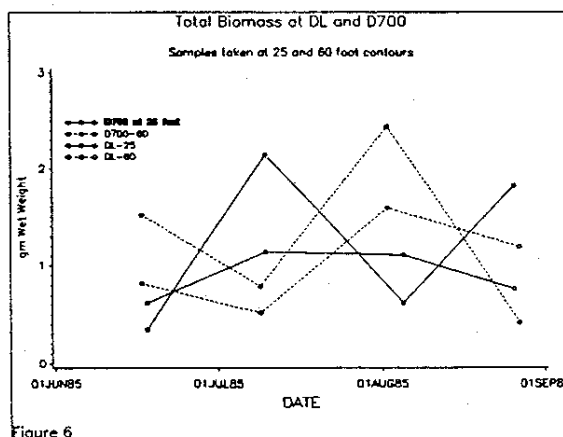
In the legend the default text associated with each plot symbol is the value of the plotted variable. You can easily change this text by using the VALUE= and TICK= options in the same way as they are used in the AXIS statement.

This example modifies the text of one of the four values of the variable STATION.

```
axis1 order=('01jun85'd to '01sep85'd by
month)
value=(f=simplex h=1.5);

axis2 label= (h=1.5 a=90 'gm Wet
Weight')
value=(f=simplex h=1.5);

legend origin=(10 pct, 65 pct)
mode=share
across=1
label=none
value=(t=1 h=1.2 f=swissb
'D700 at 25 feet')
;
```



### 7. Variable Scale Axes (logscale)

The AXIS statement options LOGSTYLE and LOGBASE can be used to create logscale graph paper which can be used as background to a plot of actual data.

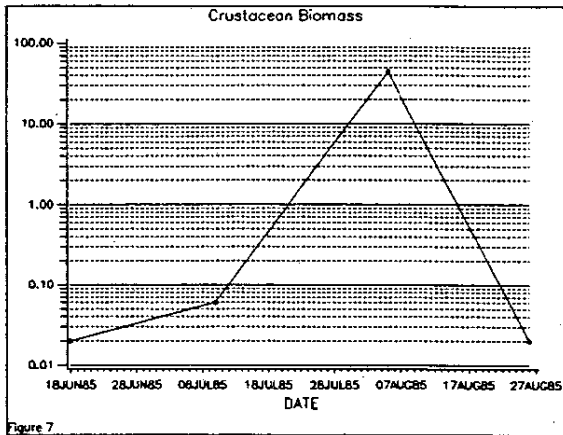
```
* vertical axis statement;
* note n=8 for minor ticks (not 9);
axis1 logstyle=expand logbase=&base
minor = (n=8 h=1)
major = (h=1.5)
label = none;
```

```
proc gplot data=pltdata ;
plot bmcruz * date = linetype / skipmiss
```

```

nolegend
vaxis=axis1;
symbol1 v=none I=join l=1 c=black;
symbol2 v=none I=join l=2 c=black;
symbol3 v=dot I=join l=1 c=black;
title "Crustacean Biomass";
footnotel j=1 h=1.5 f=simplex 'Figure
7';
run;

```



```

axis2 order= -4 to 4 by 1
label=none;
* symbol1 is used with the data;
symbol1 v=dot c=black;
* symbol2 is used to control the normal
line;
symbol2 I=join v=none l=1 c=black;

```

The probability plot is generated using CUMULPR as the vertical axis variable.

```

* Plot the probabilities with the second
axis
* formatted to normal quantiles;
proc gplot data=cumulpr;
plot cumulpr*length=1 / vaxis=axis1;
plot2 y*length=2 / vaxis=axis1;
format y probit.;
title 'Cumulative Normal Distribution';
title2 'Vertical Scale Based on
Probabilities';
title3 a=90 'Cumulative Probability';
title4 a=-90 'Normal Quantiles';
footnotel h=2 f=simplex j=1 'Figure 8';
run;

```

### 8. Creating Probability Scales

The distribution of lengths of fish is compared to the normal distribution in the following graph. A standard normal curve is first generated and then a format is created matching the normal quantiles with their associated cumulative probabilities. These are then plotted against the fish length data.

The data are standardized to a normal (0,1) distribution by using PROC STANDARD. This is necessary since the quantiles and the formats are based on this distribution. The quantiles from any other distribution could have been used, it is only necessary that the data are comparable.

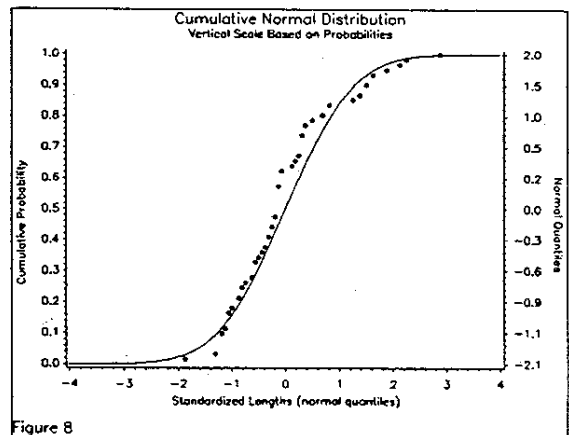
Once normalized the cumulative probabilities are calculated. The percentages can be calculated by PROC FREQ and then summed in a DATA step. The data set containing the percentages and the normal curve data are concatenated for plotting. The data used with the PLOT2 statement produces the normal curves. The cumulative probabilities for the data are calculated using the PROBIT function that converts probabilities to normal quantiles.

The same AXIS statement is used for both the left and right axis, and the values used in the ORDER= option are for the unformatted left axis.

```

* Axis and symbol definitions;
* Axis1 used to plot probabilities;
axis1 order= 0 to 1 by .1
label=none;
* Axis2 used to plot normal quantiles;

```



## USING SYMBOLS TO OVERLAY PLOTS

### 9. Cluster Analysis Scatter Plot

In this example PROC Gplot is used to plot the results of a cluster analysis. The two selected plot variables are SALINITY and DO (dissolved oxygen), and the four separate clusters are overlayed on the same plot using different symbols.

The Gplot is straight forward and uses the SYMBOL statements to assign unique symbols to each cluster.

```

* graph the clusters using Gplot;
proc gplot data=both;
plot &yvar * &xvar = cluster;

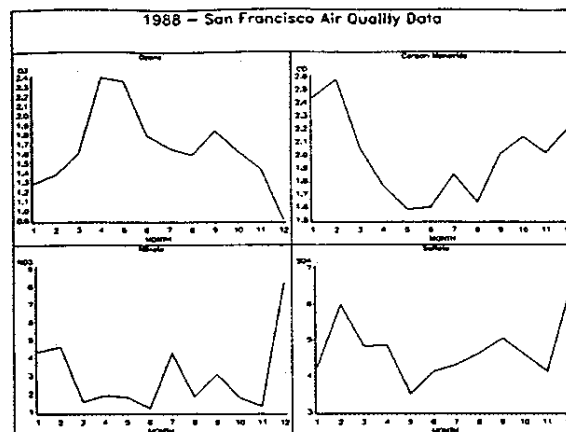
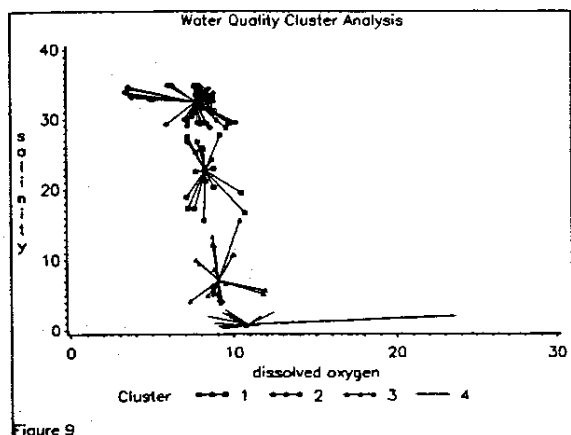
symbol1 line=1 I=join c=black v=square;

```

```

symbol2 line=1 I=join c=black v=circle;
symbol3 line=1 I=join c=black
v=triangle;
symbol4 line=1 I=join c=black v=star;
symbol5 line=1 I=join c=black v=diamond;
title 'Water Quality Cluster Analysis';
footnote h=2 f=simplex j=1 'Figure 9';

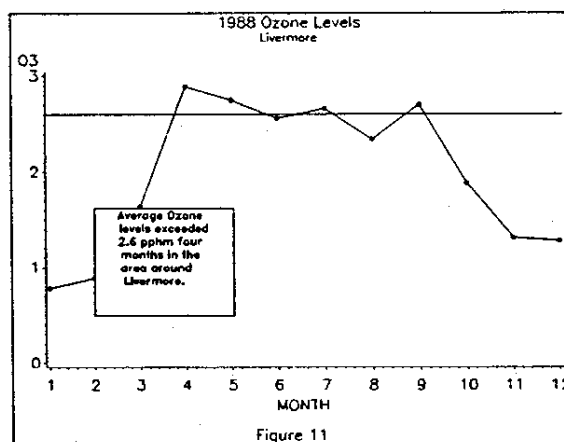
```



### 11. Text Box Using a Template

Panel coordinates need not be unique nor do they have to define distinct portions of the graphic page. Panels can be defined to overlap and overlay other panels. These overlapping panels can be used as legends, graphic inserts, or for artistic effect. Overlapping panels may or may not mask one another.

Ordinarily when two or more panels overlap the graphs from both will be displayed. Clipping prevents this by allowing the display of the contents of only one panel at a time. It is as if clipping makes the panel opaque to panels behind it (panels with higher panel numbers). Clipping is requested when the template is defined in the TEMPLATE DESIGN Window (or the TDEF statement).



## SUMMARY

SAS/GRAPH provides a number of easy to use tools. Taken together they can form an impressive array of tricks and enhancements that you can take advantage of when you next generate graphs using SAS/GRAPH.

Analysis of this figure indicates that while SALINITY does a good job of separating clusters, dissolved oxygen (DO) by itself does not.

## USING GREPLAY AND TEMPLATES

### 10. Displaying a Banner Title

It is not at all unusual to display several plots on a single page by using TEMPLATES and GREPLAY. Often these plots will have a common title which appears redundant if it is repeated on each panel. It is possible to create a banner title by using a separate PROC GSLIDE and then display it in its own template panel.

```

* Create the title panel;
proc gslide gout=voll.chapt6
name='e634full';
title h=2 f=duplex
'1988 - San Francisco Air Quality Data';
footnote1;
run;

```

## TRADEMARK INFORMATION

SAS/GRAPH is a registered trademark of SAS Institute, Inc. in the USA and other countries. • indicates USA registration.

## ABOUT THE AUTHOR

Arthur L. Carpenter has over seventeen years of experience as a statistician and data analyst and has served as a senior consultant with California Occidental Consultants, CALOXY, since 1983. His publications list includes a number of papers and posters presented at SUGI and has served as a steering committee member and president of the Southern California SAS User's Group, a Section Chair and Conference Co-chair at the Western Users of SAS Software regional conference, WUSS, and in various positions at SUGI. He has developed and presented several courses and seminars on statistics and SAS programming and has taught for Colorado School of Mines, University of Redlands, and University of California at San Diego.

CALOXY offers SAS contract programming and in-house SAS training nationwide, including a three day course on SAS/GRAPH. This presentation is based on examples taken from his book, *Quick Results Using SAS/GRAPH Software*, which was co-authored with Charlie Shipp.

## AUTHOR

Arthur L. Carpenter  
California Occidental Consultants  
4239 Serena Avenue  
Oceanside, CA 92056-5018

(619) 945-0613