Create Communication-Effective Graphs in SAS® V9.3 Without SAS/GRAPH®

LeRoy Bessler PhD, Bessler Consulting and Research, Le_Roy_Bessler@wi.rr.com

ABSTRACT

You no longer need to license SAS/GRAPH software to create graphs. This introductory tutorial shows how to create widely usable, designed-for-communication-effectiveness examples with ODS GRAPHICS and the SG procedures now included with in Version 9.3 of Base SAS. In a few cases, it also demonstrates what you can do better with SAS/GRAPH as an alternative. This tutorial shows you easy ways to get beyond taking the defaults, to get better results with minimum time and effort. It covers pie charts and bar charts written to disk to later be inserted into Microsoft PowerPoint, Word, or Excel, or simply printed, but also emphasizes and demonstrates the benefits of web-enabled time series plots, and shows how to create them. For pie charts and bar charts it is possible to supply the associated precise numbers, but for time series with plot points on multiple lines and/or with point-dense single lines, it is not. For time series, the benefits of ALT text (a.k.a. “data tips”) and optionally a companion spreadsheet are emphasized and implemented.

INTRODUCTION

My focus in this paper is on graphs used for management reporting in business, government, or other organizational settings, not statistical or heavy-duty analytical graphs. Although the SG (Statistical Graphics) procedures were originally developed for statistical graphics, they can be used for management reporting. By graphs for management reporting I mean graphs or web graphs used to answer several common questions. How are things going? Are they better, worse, or about the same? How do measurements for different entities compare? Questions like these are commonly answered visually with time plots (or time series graphs), bar charts, and pie charts. **For dramatic and unequivocal proof of the importance of graphing your data, see Reference 6.**

For macros that can enable you to more easily create graphs like some examples shown here, see Reference 1.

Graphs accelerate and facilitate inference and decision-making, but the actual numbers are required for reliable inference and decisions.

My scope does include static graphs, i.e., ones destined directly for print (probably rare these days) or for imbedding in PowerPoint for presentations, in PDF for Adobe Acrobat Reader, or in RTF for Microsoft Word. However, web-enabled graphs are my preferred choice now. Web enablement provides quick and easy navigation between graph and graph or between graph and table. It also supports what is officially called ALT text, but also is instead referred to as pop-up text (at least by me), flyover text, mouseover text, hover text, floatover text, tool tips, or data tips.

ALT text is an accessibility aid. For visually impaired web users, the HTML source code to support ALT text can be converted into audio by so-called screen reader software. But ALT text can be useful for any web user. A problem with many graphs is that you can not necessarily hard-annotate everything that you would like to communicate about a plot point, bar, pie slice, geographic unit area, or whatever. In any case, it is preferable to guessing the values for a plot point by comparing its location with axis tick marks. ALT text can be used for anything you want, as much as you want, formatted with line breaks wherever you want to enhance readability. You can even format it as a little table:

Your Label for Yvar = y value formatted however you like
Your Label for Xvar = x value formatted however you like
Any comments that you might have

Among the options that web enabled graphs offer is the ability to connect them, forwards and backwards, with SAS-created spreadsheets. Besides allowing detail look-up, the benefit of a spreadsheet rather than just a SAS table is that the user then has the option of post-processing the graph-supporting data in any way desired, using a tool that almost any user already has and knows how to use. That work was done with SAS/GRAPH (see Reference 2), but it is extended here to ODS GRAPHICS and PROC SGPLOT. However, you must add NOGTITLE to the ODS HTML statement in order to get the link to the spreadsheet from the web-deployed graph to work.

For scatter plots and line charts, it is usually difficult, when not impossible, to provide hard annotation for the data points. For them, web enablement with ALT text and/or a companion table are/is essential. Bar charts and pie charts can be created with all detail in the image area. The code here routes the charts to a disk location, from which they can be inserted in a slide or a Microsoft Word document, but they could be web-enabled.

Not demonstrated here is use of the URL parameter for HBAR and SERIES statements to make bars and points drillable. URL= specifies a variable to contain the bar- or point-appropriate hyperlink.

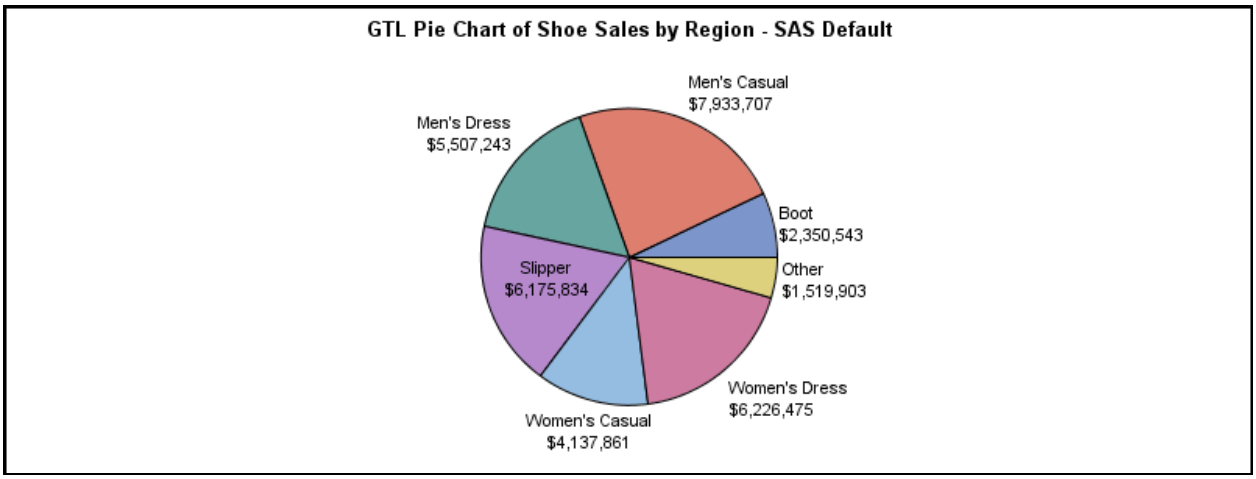
PIE CHARTS: A Work Horse of Graphic Communication

I understand that some statisticians, as well as some critics of graphic style, object to pie charts.

Well, it is a fact that pie charts are one of the commonest graphic delivery instruments, regardless of ideological abstention in some venues and despite misconceptions about what is possible. **If a pie chart is designed and executed well, there is no better way to visually compare the relative sizes of shares of the whole.**

Let's proceed to achieve this, but starting from defaults, which DO deliver sub-optimal results.

Unacceptable Default Pie Chart: NOT Communication-Effective for several reasons enumerated below.



1. Pie slices are ordered by label name, not by size.
2. "Other" withholds, rather than delivers, information.
3. The numeric values of percent share of the whole are not shown.
4. Some labels are outside, one label is inside.

Failing to be able to order by size makes it cumbersome to identify the relative significance of the shares of the whole. This is NOT an inherent pie chart defect. It was a choice of the SAS ODS GRAPHICS software developer.

The presence of an "Other" slice is an all too common pie chart practice. Any graph or other report should answer questions, not prompt them (like "What is IN 'Other'?")

Graphs accelerate inference. Numbers are necessary for reliable inference. Pie chart slices should be accompanied by the numeric percents of the whole.

Putting labels inside the pie slices causes two problems:

(a) Black text on white background is maximally readable. Text on a color background is more difficult to read.

(b) Slices are always more constricted than the while space outside the pie. This increases the difficulty of also supplying the numeric percent of the whole.

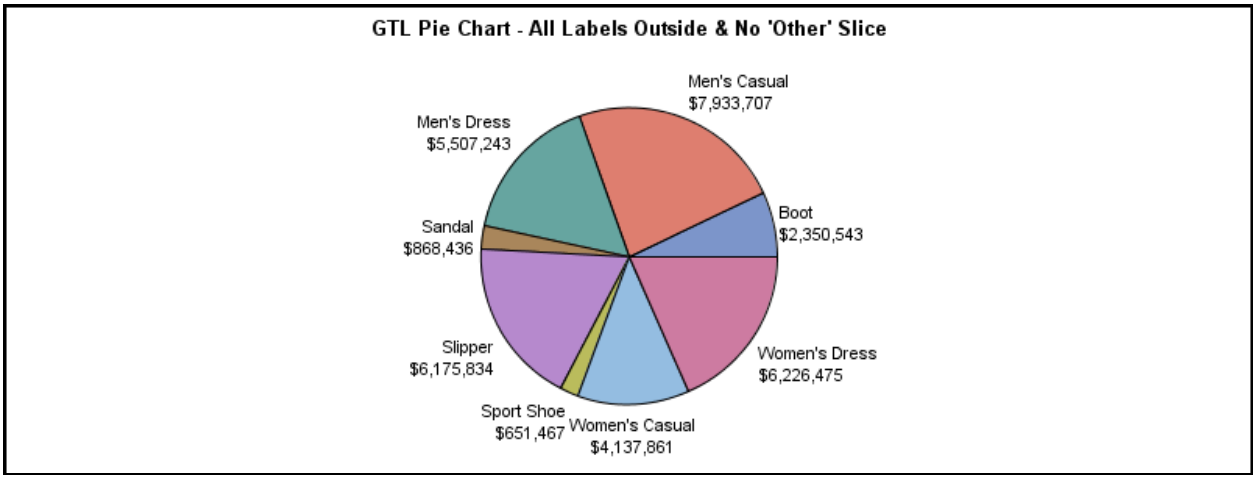
Software Problem: Instead of providing an SG procedure, or an SG procedure feature, to DIRECTLY create a pie chart, the software developers require users to get involved with GTL (Graphic Template Language), which is a less user-friendly tool.

Here is the code used to create the pie chart above:

```
proc template;
define statgraph SASDefaultPieChart;
  begingraph;
    entrytitle "GTL Pie Chart of Shoe Sales by Region - SAS Default";
    layout region;
      piechart category=Product response=Sales;
    endlayout;
  endgraph;
end; run;

ods listing gpath="D:\@WIILSU Jun2012\Results";
ods graphics on / reset=all border=on height=300px width=800px
                  imagename='SGRENDER_GTL_PieChart_SASDefault';
proc sgrender data=sashelp.shoes template=SASDefaultPieChart;
run;
```

Better, But Not Best, Pie Chart: Pie slices are still ordered by label name, not size, and percent of whole is missing.



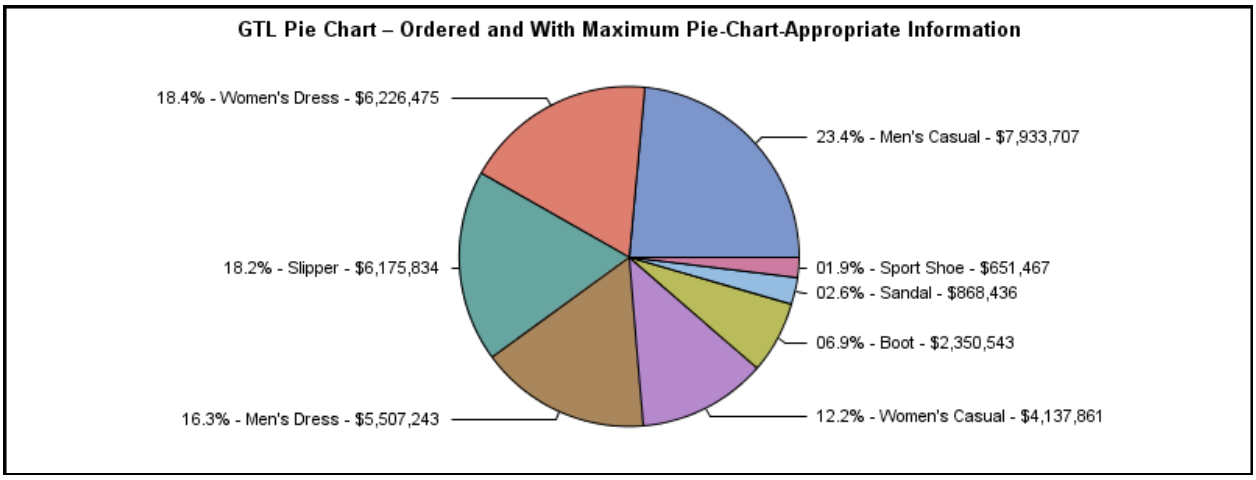
Better: All slice labels are now outside the pie, and all shoe styles are now explicitly identified—No “Other” permitted.

Here is the code, now including two simple added parameters, used to create the pie chart above:

```
proc template;
define statgraph BesslerNearDefaultPieChart;
  begingraph;
    entrytitle "GTL Pie Chart - All Labels Outside & No 'Other' Slice";
    layout region;
      piechart category=Product response=Sales /
               datalabellocation=outside
               otherslice=FALSE
               ;
    endlayout;
  endgraph;
end;
run;

ods listing gpath="D:\@WIILSU Jun2012\Results";
ods graphics on / reset=all border=on height=300px width=800px
                  imagename='SGRENDER_GTL_PieChart_BesslerNearDefault';
proc sgrender data=sashelp.shoes template=BesslerNearDefaultPieChart;
run;
```

Best Pie Chart: Pie slices are ordered by size, and numeric percent of whole is included in the slice labels.



Below is the more complicated code needed to create the pie chart above. A companion paper (see Reference 1) provides a macro solution to make this easier and less vulnerable to possible error.

```
proc summary data=sashelp.shoes nway;
class Product;
var Sales;
output out=ToPrep sum=TotalByClass;
run;

proc sql noprint;
select sum(TotalByClass) into :GrandTotal from ToPrep;
quit;

data ToChart;
length SliceNameWithPercentAndValue $ 256;
set ToPrep;
SliceNameWithPercentAndValue =
  trim(left(
  put(((TotalByClass / &GrandTotal) * 100),z4.1)
  )) ||
  '% - ' || trim(left(Product)) ||
  ' - ' ||
  trim(left(put(TotalByClass,dollar10.)));
run;
proc sort data=ToChart;
by Descending TotalByClass;
run;

proc template;
  define statgraph BesslerBestPieChart31May2012;
   begingraph;
     entrytitle
       "GTL Pie Chart – Ordered and With Maximum Pie-Chart-Appropriate Information";
     layout region;
       piechart category=SliceNameWithPercentAndValue
                response=TotalByClass /
                datalabelcontent=(category)
                datalabellocation=callout
                otherslice=FALSE;
     endlayout;
   endgraph;
  end;
run;
```

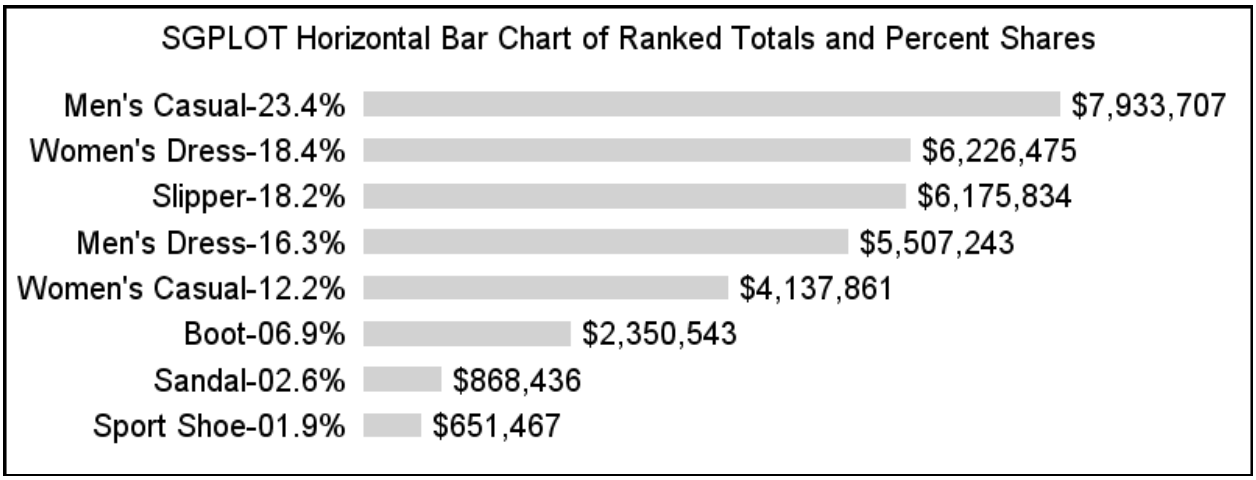
```
ods listing gpath="D:\@WIILSU Jun2012\Results";

ods graphics on / reset=all
                  border=on
                  height=300px
                  width=800px
                  imagename="BesslerBestPieChartCreatedNotUsingMacro";

proc sgrender data=ToChart template=BesslerBestPieChart31May2012;
run;
```

BAR CHART OF SUMS WITH PERCENT SHARES: ALTERNATIVE FOR WHEN THE PIE CHART IS INFEASIBLE OR UNACCEPTABLE

I have long been an advocate for horizontal bar charts rather than vertical bar charts. Vertical bar charts work well only when the bar labels are short. Tilted, or worse, vertical labels for vertical bars are somewhere between inelegant and outright anti-communicative. With V9.2, the length on labels was extended to 256, which is always adequate, and, for horizontal bar charts, always useful for longer labels that are, in fact, often needed. 256 would be impractical (no space left for the bar—unless you expand the image width, which IS possible), but it's a welcome, friendly limit.



Below is the code used to create the chart above. A companion paper (see Reference 1) provides a macro solution to make this easier and less prone to possible error.

```
proc summary data=sashelp.shoes nway;
class Product;
var Sales;
output out=ToPrep sum=TotalByClass; run;

proc sql noprint;
select sum(TotalByClass) into :GrandTotal from ToPrep; quit;

data ToChart;
length BarNameWithPercent $ 256;
set ToPrep;
BarNameWithPercent = trim(left(Product)) ||
  '-' || trim(left(put(((TotalByClass / &GrandTotal) * 100),z4.1))) || '%'; run;

proc template;
define style styles.MinimalWithNoFrame; /* remove a useless box around the bars */
  parent=styles.Minimal;
  class graphwalls / frameborder=off;
end; run;

ods listing gpath="D:\@WIILSU Jun2012\Results" style=styles.MinimalWithNoFrame
ods graphics on / reset=all border=on height=300px width=800px
                  imagename="SGPLOThorizontalBarChartOfRankedTotalsAndPercentShares";

title height=16pt "SGPLOT Horizontal Bar Chart of Ranked Totals and Percent Shares";
proc sgplot data=ToChart;
hbar BarNameWithPercent / response=TotalByClass categoryorder=respdesc
                          datalabel datalabelattrs=(size=16pt) barwidth=0.5 nooutline;
yaxis display=(nolabel noline noticks) valueattrs=(size=16pt);
xaxis display=none;
run;
```

BAR CHART OF SUMS WITH TWO LEVELS OF CATEGORIZATION:

The SASHELP.SHOES data set classifies sales not only by Product, but also by Region.

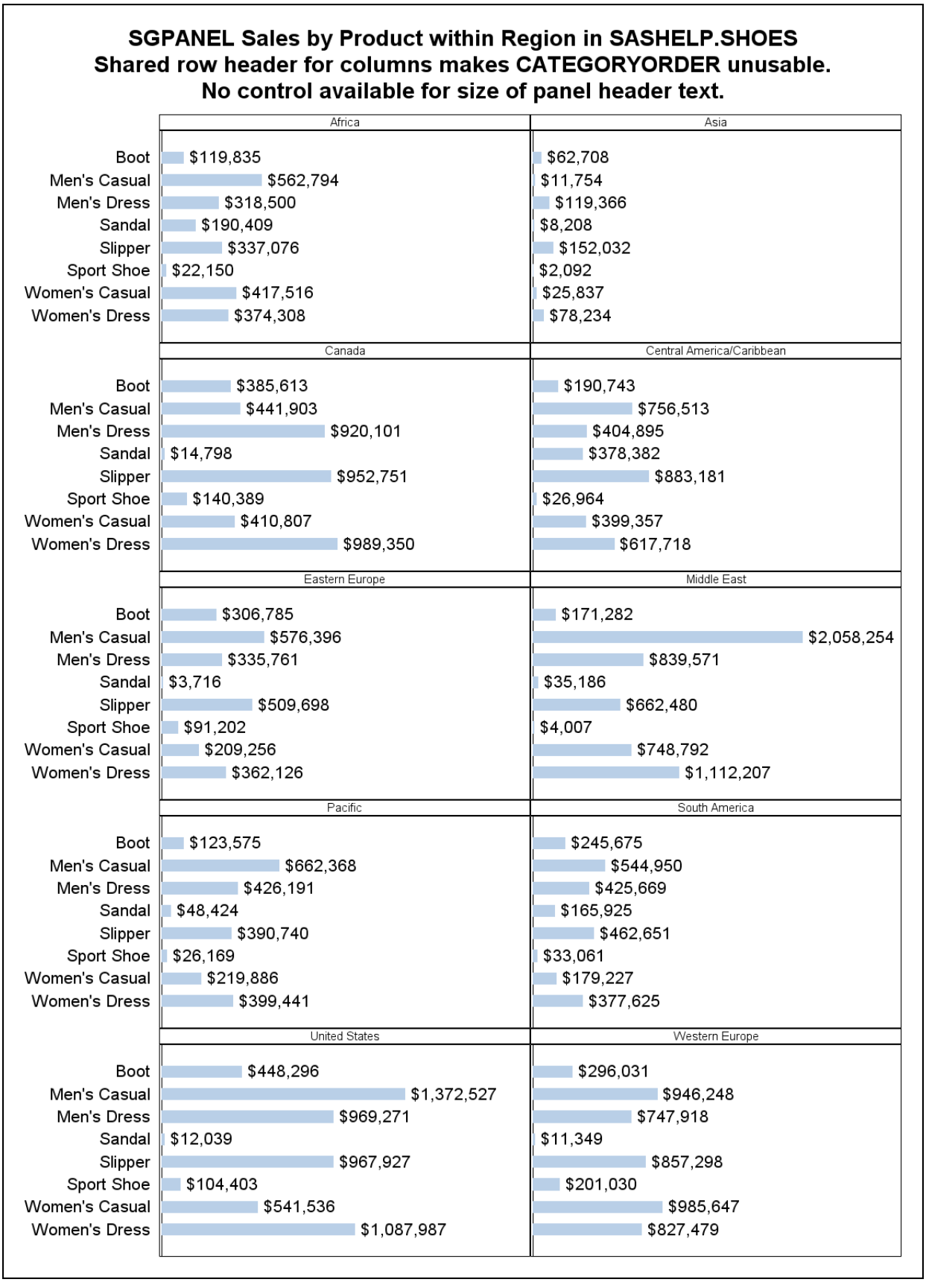
Here the code used to handle that situation, and to create the chart on the following page:

```
ods listing gpath="D:\@WIILSU Jun2012\Results";

ods graphics on / reset=all border=on height=1600px width=1155px
                  imagename='SGPANELhbar';

title1 height=16pt "SGPANEL Sales by Product within Region in SASHELP.SHOES";
title2 height=16pt "Shared row header for columns makes CATEGORYORDER unusable.";
title3 height=16pt "No control available for size of panel header text.";
proc sgpanel data=sashelp.shoes;
panelby region / novarname rows=5 columns=2;
rowaxis display=(nolabel noline noticks) valueattrs=(size=12pt);
colaxis display=none;
hbar Product / response=Sales barwidth=0.5 nooutline
               datalabel datalabelattrs=(size=12pt);
run; quit; title;
```

The main disadvantage of this graphic solution is that it is unable to order the bars within each region based upon the Sales By Product.

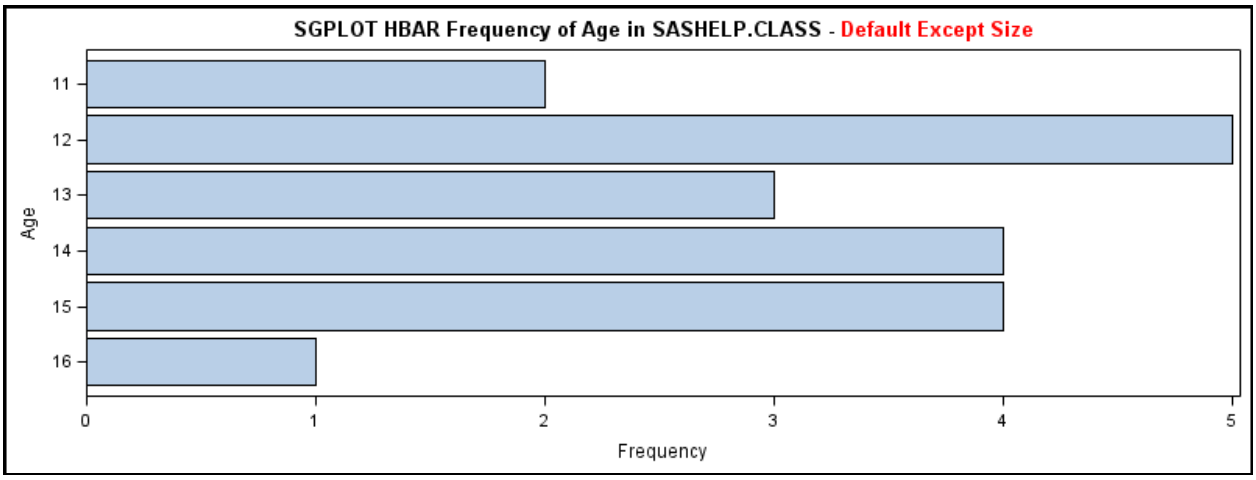

SGPANEL Sales by Product within Region in SASHELP.SHOES
Shared row header for columns makes CATEGORYORDER unusable.
No control available for size of panel header text.
Africa
Boot $119,835
Men's Casual $562,794
Men's Dress $318,500
Sandal $190,409
Slipper $337,076
Sport Shoe $22,150
Women's Casual $417,516
Women's Dress $374,308
Asia
$62,708
$11,754
$119,366
$8,208
$152,032
$2,092
$25,837
$78,234
Canada
Boot $385,613
Men's Casual $441,903
Men's Dress $920,101
Sandal $14,798
Slipper $952,751
Sport Shoe $140,389
Women's Casual $410,807
Women's Dress $989,350
Central America/Caribbean
$190,743
$756,513
$404,895
$378,382
$883,181
$26,964
$399,357
$617,718
Eastern Europe
Boot $306,785
Men's Casual $576,396
Men's Dress $335,761
Sandal $3,716
Slipper $509,698
Sport Shoe $91,202
Women's Casual $209,256
Women's Dress $362,126
Middle East
$171,282
$2,058,254
$839,571
$35,186
$662,480
$4,007
$748,792
$1,112,207
Pacific
Boot $123,575
Men's Casual $662,368
Men's Dress $426,191
Sandal $48,424
Slipper $390,740
Sport Shoe $26,169
Women's Casual $219,886
Women's Dress $399,441
South America
$245,675
$544,950
$425,669
$165,925
$462,651
$33,061
$179,227
$377,625
United States
Boot $448,296
Men's Casual $1,372,527
Men's Dress $969,271
Sandal $12,039
Slipper $967,927
Sport Shoe $104,403
Women's Casual $541,536
Women's Dress $1,087,987
Western Europe
$296,031
$946,248
$747,918
$11,349
$857,298
$201,030
$985,647
$827,479

BAR CHARTS OF FREQUENCY DISTRIBUTIONS
PROC SGPLOT (from ODS GRAPHICS) and PROC GCHART (from SAS/GRAPH)

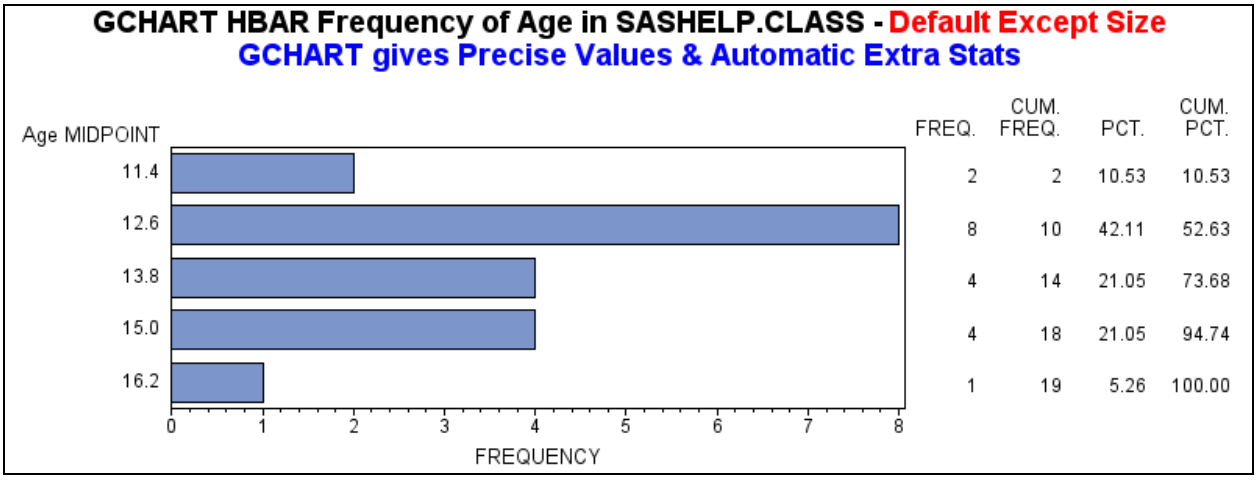
Though this paper promised to be about doing graphs without SAS/GRAPH, this section will show you that SAS/GRAPH offers some useful functionality that is NOT available with ODS GRAPHICS and PROC SGPLOT.

BAR CHARTS OF FREQUENCY DISTRIBUTIONS
Defaults – SGPLOT vs. GCHART

I have long been an advocate for horizontal bar charts rather than vertical bar charts. Vertical bar charts work well only when the bar labels are short. Tilted, or worse, vertical labels for vertical bars are somewhere between inelegant and outright anti-communicative. With V9.2, the length on labels was extended to 256, which is always adequate, and, for horizontal bar charts, always useful for longer labels that are, in fact, often needed. 256 would be impractical (no space left for the bar—unless you expand the image width, which IS possible), but it's a welcome, friendly limit.



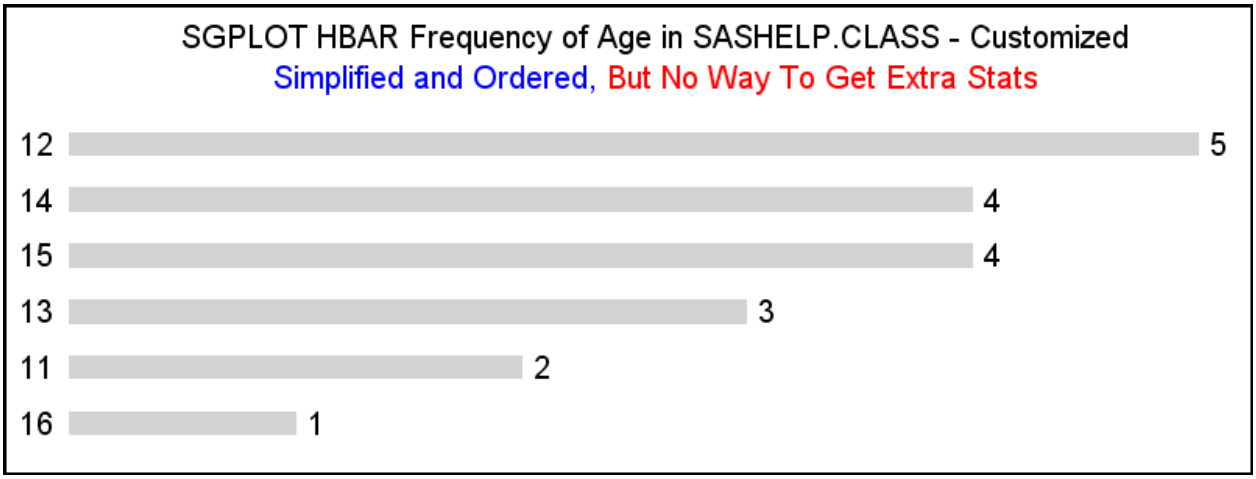

```
ods listing gpath="D:\@WIILSU JUN2012\RESULTS";
ods graphics on / reset=all height=300px width=800px
                  imagename='SGPLOThbarFreqChart_SAS_Default';
title "SGPLOT HBAR Frequency of Age in SASHELP.CLASS - "
      color=red "Default Except Size";
proc sgplot data=sashelp.class;
hbar Age / stat=freq;
run;
```




```
ods graphics off;
goptions reset=all gsfname=anyname gsfmode=replace device=png ypixels=300;
filename anyname "D:\@WIILSU JUN2012\Results\GCHARThbarFreqChart_SAS_Default.png";
title "GCHART HBAR Frequency of Age in SASHELP.CLASS - " color=red "Default Except
Size" justify=center color=blue "GCHART gives Precise Values & Automatic Extra Stats";
proc gchart data=sashelp.class;
hbar Age; run; quit;
```

BAR CHARTS OF FREQUENCY DISTRIBUTIONS
Customization (includes Ordering) – SGPLOT vs. GCHART

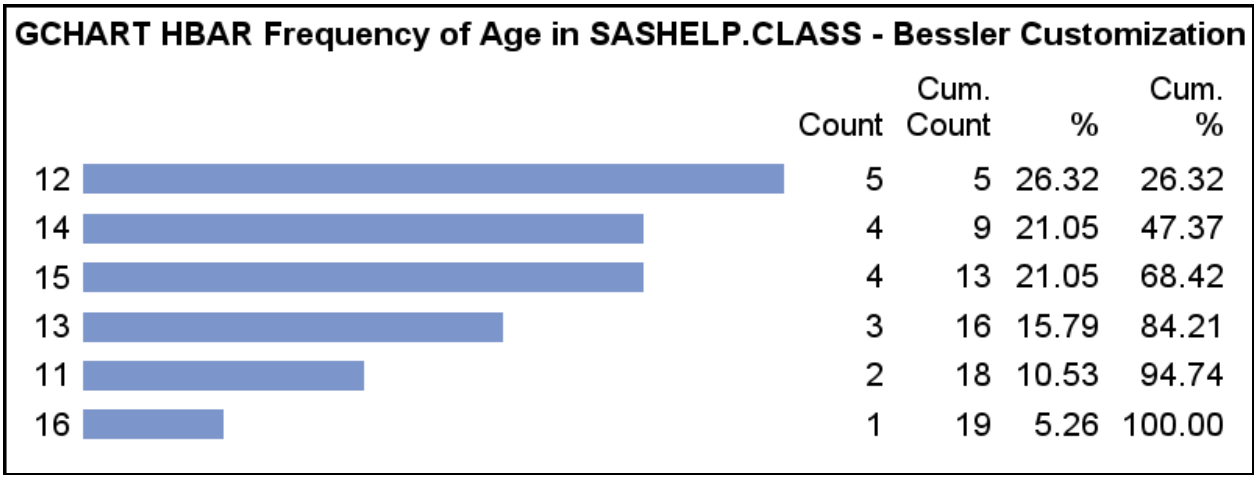
NOTE: One of my longstanding graphic design guidelines is: Show Them What's Important. Order the information.




```
proc template;
define style styles.MinimalWithNoFrame; /* remove a useless box around the bars */
  parent=styles.Minimal;
  class graphwalls / frameborder=off;
end; run;

ods listing gpath="D:\@WIILSU JUN2012\Results" style=styles.MinimalWithNoFrame;
ods graphics on / reset=all border=on height=300px width=800px
                  imagename='SGPLOThbarFreqChart_Bessler_Custom';
title1 height=16pt "SGPLOT HBAR Frequency of Age in SASHELP.CLASS - Customized";
title2 height=16pt color=blue "Simplified and Ordered, " color=red "But No Way To Get
Extra Stats";
proc sgplot data=sashelp.class;
yaxis display=(nolabel noline noticks) valueattrs=(size=16pt); xaxis display=none;
hbar Age / stat=freq categoryorder=respdesc datalabel datalabelATTRS=(size=16pt)
     barwidth=0.4 nooutline; run;
```

With SGPLOT, it is advantageous to have the counts automatically appended to the bar end using ANNOTATE. This is more important when there are many bars, especially short ones, which would be distant from the table of information provided in PROC GCHART as in the exhibit below.



Here is the code used to create the chart above:

```
ods listing style=styles.listing; /* See NOTE below this code block */
ods graphics off;
goptions reset=all gsfname=anyname gsfmode=replace device=png ypixels=300;
filename anyname "D:\@WIILSU JUN2012\Results\GCHARThbarFreqChart_Bessler_Custom.png";
goptions htext=16pt border;
title height=1 ' ' justify=center height=16pt
      "GCHART HBAR Frequency of Age in SASHELP.CLASS - Bessler Customization";
proc gchart data=sashelp.class;
axis1 label=none style=0;
axis2 label=none style=0 value=none major=none minor=none;
hbar Age / name='GCHARThbarFreqChart_Bessler_Custom' maxis=axis1 raxis=axis2 width=3
  space=2.25 discrete descending noframe coutline=same freqlabel='Count'
  percentlabel='%' cfreqlabel='Cum. Count' cpercentlabel='Cum. %';
run; quit;
```

NOTE: IF you run the above code in a fresh SAS session, you CAN drop the first statement. It sets the style for the ODS LISTING destination to what IS the default for a fresh SAS session. However, if you run this code back-to-back with the ODS GRAPHICS PROC SGPLOT code for the earlier example, and you OMIT the first statement in this PROC GCHART code, then **styles.MinimalWithNoFrame** will still be in effect and your result will look different with respect to the font used for the text. **What is puzzling is that the font in that case is NOT the same font as shown in the SGPLOT example and not the font shown in this GCHART example.** However, I am content to leave THAT puzzle unexplained, and to just get the result that I expected. For PROC GCHART, the frame is removed with the **noframe** parameter, and no custom style is required.

TIME PLOTS, TIME SERIES GRAPHS, TREND LINES, LINE CHARTS, etc.

In SAS/GRAPH, the GPLOT procedure can visually present the evolution of data values over time. In ODS GRAPHICS, the SG procedures provide many more ways. In References 3 & 4, in collaboration with Alexandra Riley, I compared a variety of ways to visually present time series data using the old and new technology. In Reference 5, I provided more information about such plots and tools.

One of the tools omitted in that article is the SERIES plot, which is the focus here. I use SERIES plots both with PROC SGPLOT and PROC SGPANEL.

These examples use web-deployed graphs so that ALT text can be provided, rather than forcing the viewer to guess y and x values based on tick mark values at the axes. Another solution for time series, available in Reference 2, is to link the plot forwards and backwards with a spreadsheet.

NOTE 1: Most of the illustrations in this section are inserted PNG files, not screen prints of the web graphs from a web browser window. However, all code does create the web-enabling HTML for the graphs.
NOTE 2: These examples require that the code in Appendix 1 be run first to prepare the input data.

In PROC GPLOT, DESCRIPTION=' ' can be used on the PLOT statement used in order to prevent the display of the default pop-up text that tries to describe the graph when you rest the mouse anywhere in the graph area on the web page. Or you can use it to provide a custom description to replace the default. For Statistical Graphics procedures, the default description can neither be nullified nor customized. These pop-up descriptions can be a nuisance when you want to instead display the ALT text for the plot points.

The ALT text for PROC GPLOT is provided via the PLOT statement's HTML parameter, which identifies a variable on the plot input data set. The variable can be customized with anything the designer/programmer desires, including line breaks, any text, and the plot point values in any format to any precision, e.g. with day of the week name, as well as month, day, year. For Statistical Graphics procedures, ALT text is triggered by IMAGEMAP=ON in an ODS GRAPHICS statement. The text displayed is simply a list of the form Variable Name (or Label) = Value, and cannot be customized.

For these graphs, the data used will be SASHELP.CITIDAY, which is shipped with SAS/ETS. It contains daily history for financial market information. If you don't have SAS/ETS, you can ask SAS Technical Support for how to download the data set. The data used here is limited to year 1990.

Since the best way to determine the precise value of the Dow in these plots is from pop-up ALT text,
here I display along the y axis only the minimum and maximum values for the Dow.

Introduction to the Time Series Examples: Discussion of Statements and Parameters

Statements Common To All of the Time Series Charts

```
ods graphics on /
    reset=all     /* Suppress any leftovers from prior code runs */
    border=off    /* Not usually desired on a web page
                     but turn on if wanting to see how much extra space
                     might be available in the browser window
                     for a wider image */
    antialiasmax=2500 /* antialiasing is on by default,
                         but antialiasmax is defaulted to 600.
                         If that is insufficient,
                         then antialiasing will be incomplete */
    tipmax=2500 /* This is the maximum number of distinct mouse-over areas allowed
                   before data tips are disabled. Default is 400. */
    imagemap=on /* This turns on the data tips.
                   SAS/GRAPH allows customization of the data tips,
                   but ODS GRAPHICS does not. */
    imagename='DesiredFileNameForThePNGimageFile';

yaxis display=(nolabel) /* the yaxis variable is identified in the plot title */
      values=(&Ymin_1990 &Ymax_1990); /* show only the yaxis minimum and maximum,
             available as symbolic (or macro) variables from setup processing. */

xaxis display=(nolabel) /* the xaxis variable is identified in the plot title */
      grid  /* provide reference lines for each xaxis tick mark value         */
      . . . /* here specify a list of values or (see below) Interval=         */
```

Example Assignments for Parameters Added To ODS GRAPHICS Statement for Some Series Charts

```
width=800px   /* width  of image in pixels, default is 640 */
height=600px  /* height of image in pixels, default is 480 */
```

If your target is HTML (web page), your choice of dimensions should rarely, if ever, exceed the dimensions of the available live space in the browser and monitor likely to be used. Live Space is the viewable display area WITHOUT scrolling. The usability of a graph is diminished if it needs to be scrolled.

Typical (but not All) Parameters Used on SERIES Statement in This Paper

```
series y=YourYAXISvar x=YourXAXISvar /
       markers               /* turns on plot-point markers
                                default is nomarkers */
       markerattrs=(size=7
                    symbol=circlefilled
                    color=red /* But DO NOT specify a color
                                 if using GROUP= on the SERIES statement
                                 to produce a multi-line overlay */
                   )
       lineattrs=(thickness=N /* where, in this paper, N is 2 or 3.
                                 3 is used when chart has multiple lines
                                 (as in the twelve months of 1990 overlaid)
                                 so that line colors are more easily distinguished */
                  pattern=solid /* If you do not specify your pattern preference,
                                   the software might make a decision for you,
                                   which you might not like. */
                  color=blue  /* But DO NOT specify a color
                                 if using GROUP= on the SERIES statement
                                 to produce a multi-line overlay */
                   );
```

Parameters of Special Interest on the XAXIS Statement

For Date, Time, or DateTime Variables:

`Interval= can have numerous different values.`
`Used here will be MONTH, SEMIMONTH, and WEEK.`

In some graphs here, the XAXIS variable is Day Of Month, which is not a Date.
There Interval= does not apply.

For Any Variable:

`FitPolicy=` defaults to **`THIN`**, which means that ODS GRAPHICS SG procedures will OMIT any tick mark values that it cannot fit (based on its estimates, which CAN BE wrong). Also, it does this omission with no message in the SAS log, even if you have explicitly specified a list of tick mark values to be displayed with the **`VALUES=`** parameter or have certain expectations based on what you might have specified with the **`INTERVAL=`** parameter.

When the displayed tick mark values are fewer than what you would like, there are five options:

increase the width of the image if the xaxis has the problem
specify **`FitPolicy=Stagger`**
specify **`FitPolicy=StaggerRotate`** (first try Stagger, if no help, use Rotate)
specify **`FitPolicy=RotateStagger`** (first try Rotate, if no help, use Stagger)
specify **`FitPolicy=Rotate`** (least desirable)

NOTE: When a yaxis set of values is thinned, if that causes a problem, your only option is to increase the height of the image. **If the yaxis variable has discrete character values, thinning is absolutely unacceptable.**

Time Series Examples: All Using SERIES Plots

All examples use Dow Index data for one year of trading days in 1990

One of these uses PROC SGPANEL which creates an array of subplots for segments of the time period. In the case presented here, it is an array of the twelve months of year 1990.

The remaining examples all use PROC SGPLOT.

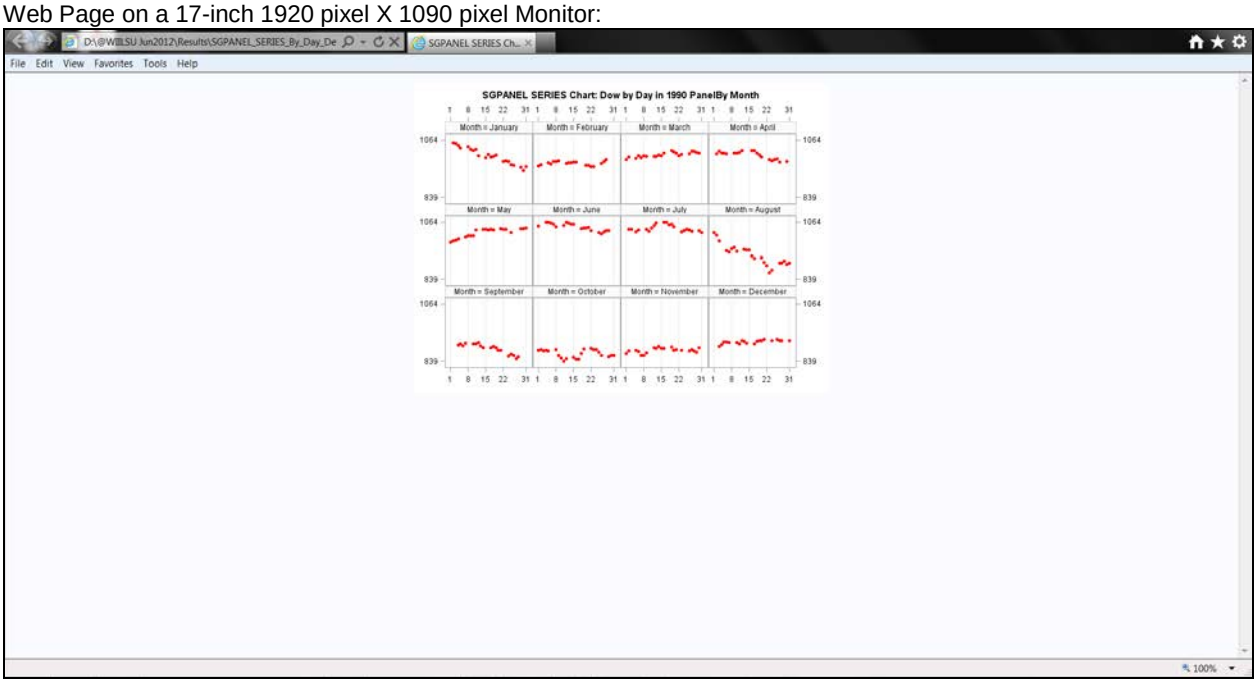
If GROUP=MONTH is added to the SERIES statement, the twelve months are plotted as separate lines in an overlay.

Without GROUP=MONTH, the data is plotted as one line.

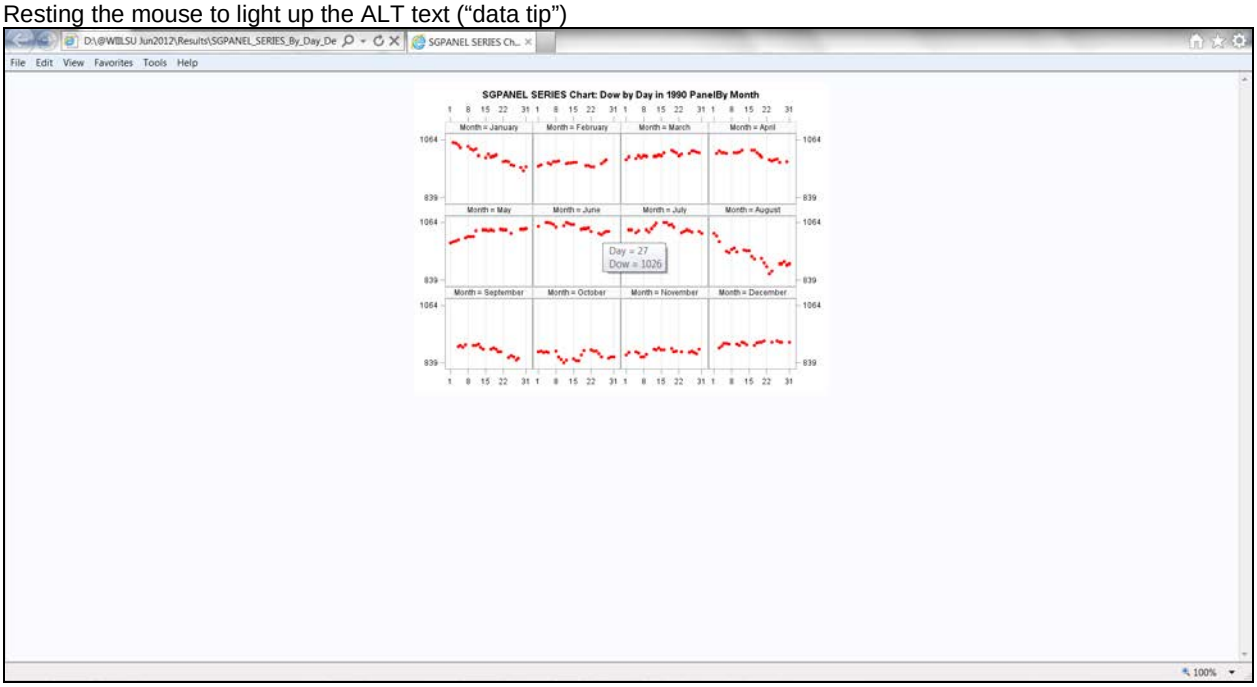
The last example demonstrates a plot and spreadsheet hyperlinked to each other forwards and backwards. ALT text (aka “data tips”) on the web-deployed plot DOES make precise numbers available for temporary look-up, but the spreadsheet makes them available both for casual inspection as a whole, and for reuse with Excel however the user of the deliverable sees fit to explore or further manipulate the data.

SGPANEL SERIES Plot By Day (default size) Paneled By Month

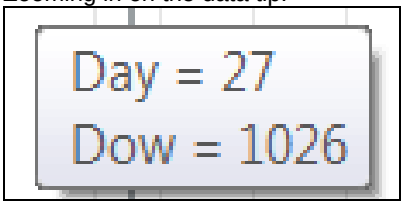
Web Page on a 17-inch 1920 pixel X 1090 pixel Monitor:



Resting the mouse to light up the ALT text (“data tip”)

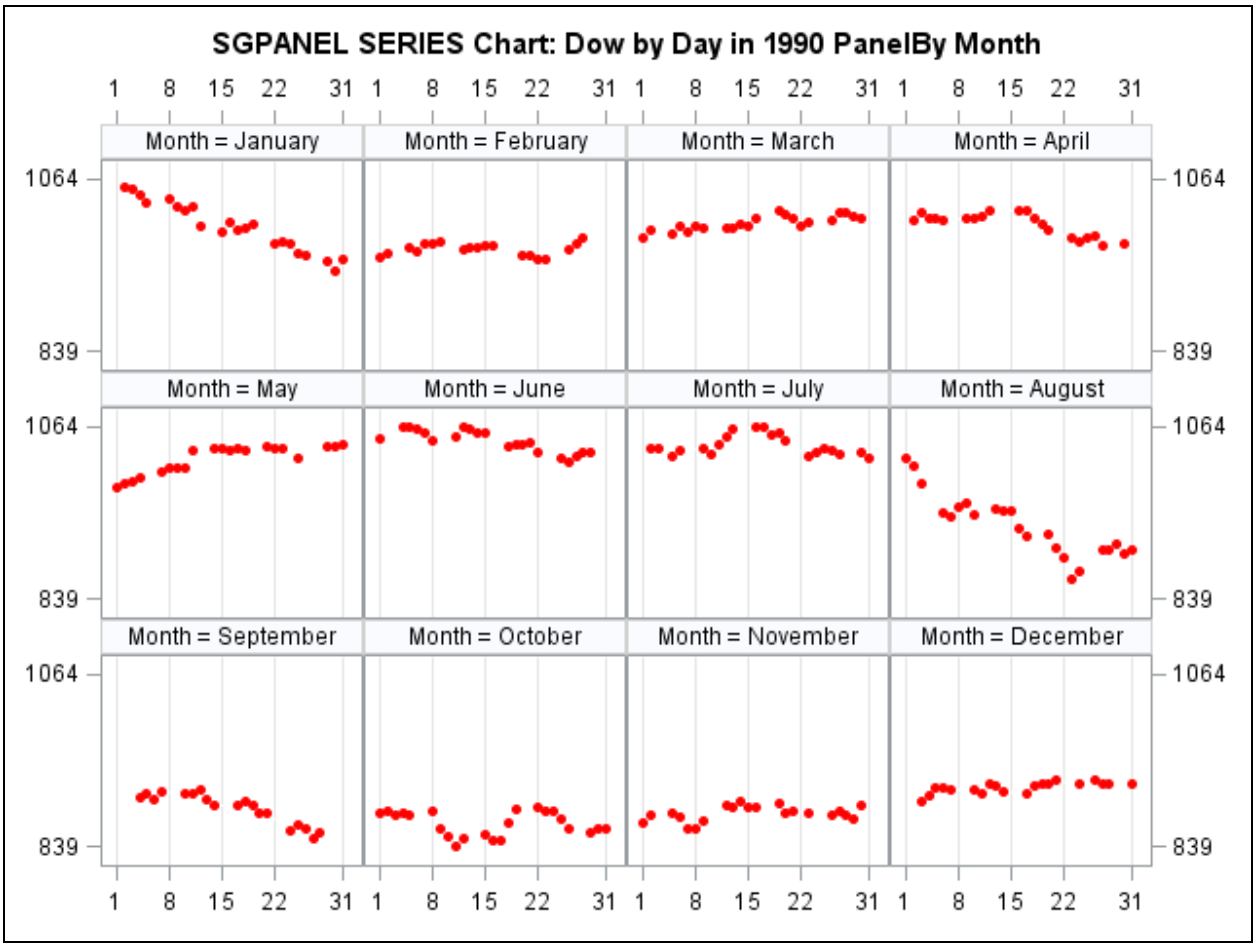


Zooming in on the data tip:



The Image File At Actual Size

In this context, it is easy to reliably associate the Month=*MonthName* labels with the appropriate subplot. However, in a very complex array, a viewer unfamiliar with this format might be uncertain as to whether a label goes with the subplot above or the subplot below. A better frame structure would clearly associate each label with its subplot.

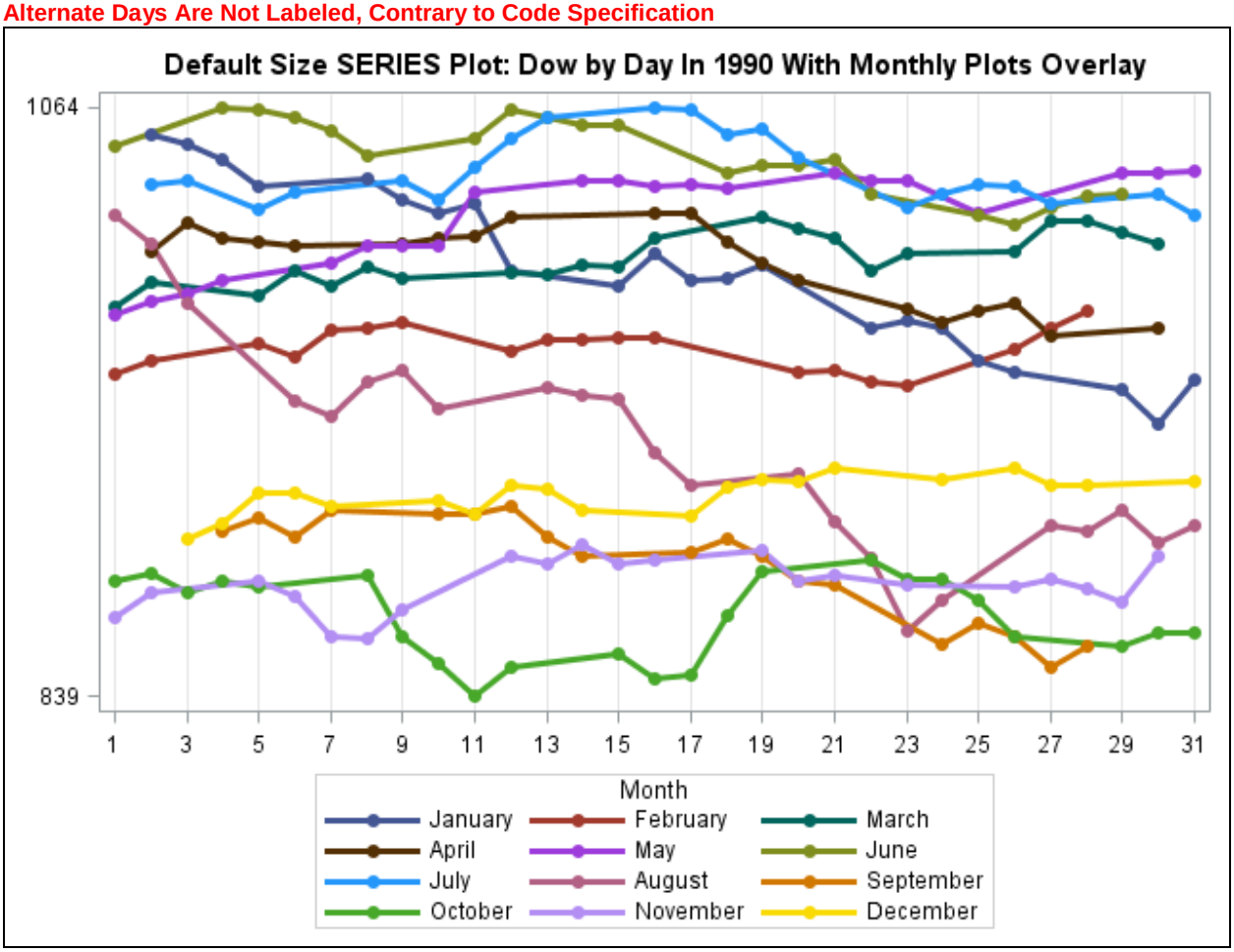


Below is the code to create this web-deployed graph. See predecessor processing in Appendix 1.

```
ods noresults; ods listing close;
ods graphics on / reset=all border=off antialiasmax=2500 tipmax=2500
    imagemap=on imagename='SGPANEL_SERIES_By_Day_DefaultSize_PanelBy_Month';
ods html path="D:\@WIILSU Jun2012\Results"
    (url=none) /* make the combination of HTML file and PNG image file portable */
    body='SGPANEL_SERIES_By_Day_DefaultSize_PanelBy_Month.html'
    (title='SGPANEL SERIES Chart Dow by Day in 1990 PanelBy Month');
title 'SGPANEL SERIES Chart: Dow by Day in 1990 PanelBy Month';
proc sgpanel data=work.DowByDayIn1990;
  panelby Month / columns=4 rows=3;
  series y=Dow x=Day / markers markerattrs=(size=7 symbol=circlefilled color=red)
         lineattrs=(color=white); /* hide the line which adds no value */
  rowaxis display=(nolabel) refticks=(values) values=(&Ymin_1990 &Ymax_1990);
  colaxis display=(nolabel) refticks=(values) values=(1 8 15 22 31) grid;
format Month MonthNm.;
format Dow 5. Day 2.;
run;
ods html close; ods listing;
```

Image File (Actual Size: default 640px X 480px) for SGPANEL SERIES Plot By Day With Monthly Lines Overlaid

Alternate Days Are Not Labeled, Contrary to Code Specification

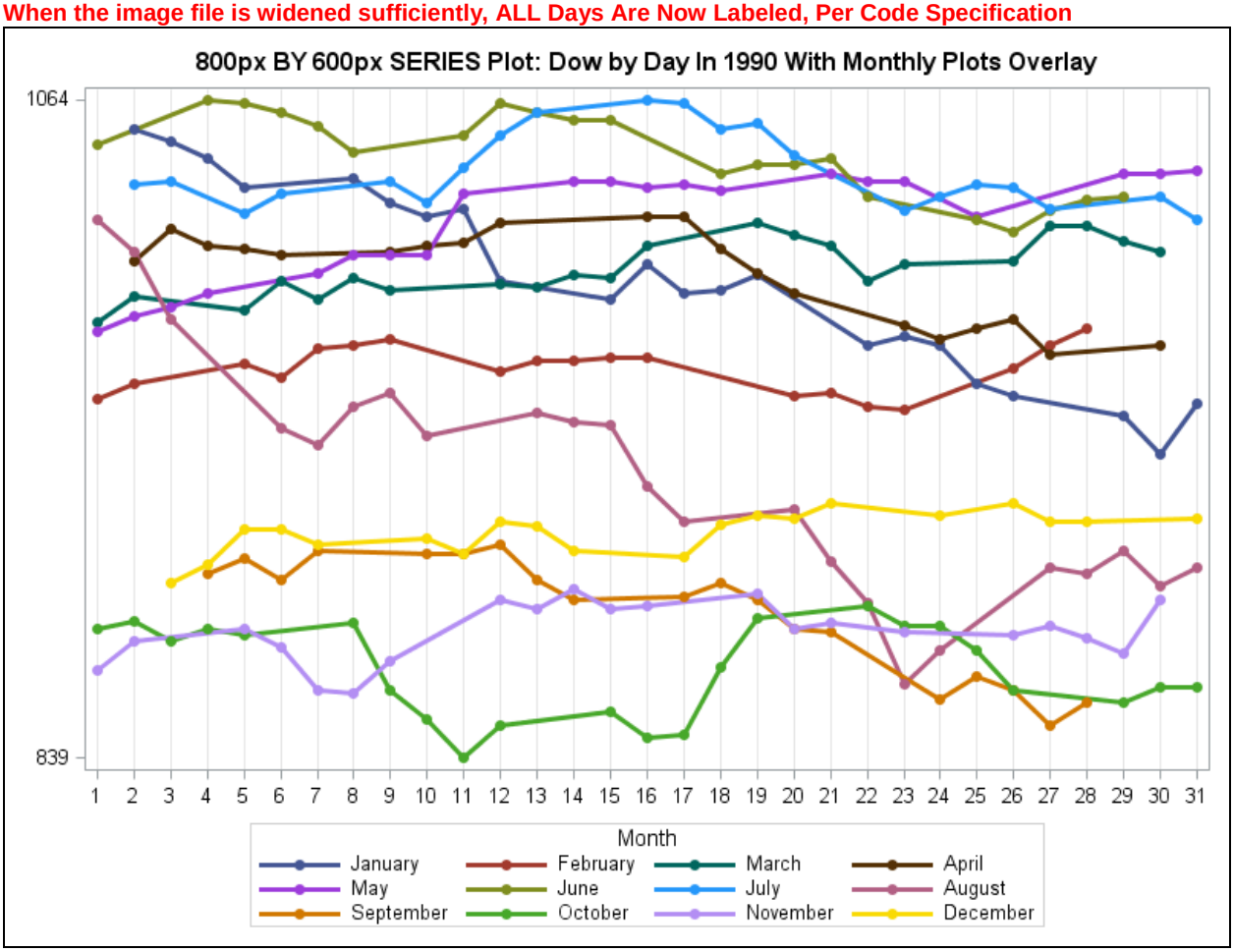


Below is the code to create this web-deployed graph.

```
ods noresults; ods listing close;
ods graphics on / reset=all border=off antialiasmax=2500 tipmax=2500
    imagemap=on imagename='SGPLOT_SERIES_By_Day_With_Months_Overlaid_DefaultSize';
ods html path="D:\@WIILSU Jun2012\Results" (url=none)
    body='SGPLOT_SERIES_By_Day_With_Months_Overlaid_DefaultSize.html'
    (title='Default Size SERIES Plot - Dow by Day In 1990 With Monthly Plots
Overlay');
title 'Default Size SERIES Plot: Dow by Day In 1990 With Monthly Plots Overlay';
proc sgplot data=work.DowByDayIn1990;
  series y=Dow x=Day /
         group=Month /* creates the overlay of monthly lines */
         markers markerattrs=(size=7 symbol=circlefilled)
         lineattrs=(thickness=3 /* thicken the lines for color distinguishability */
                    pattern=solid);
  yaxis display=(nolabel) values=(&Ymin_1990 &Ymax_1990);
  xaxis display=(nolabel)
        values=(1 to 31 by 1) /* NOT DELIVERED IN THE RESULT */
        grid;
format Month MonthNm.;
format Dow 5. Day 2.;
run;
ods html close; ods listing;
```

Image File (Reduced Here) for SGPANEL SERIES Plot By Day (custom 800px X 600px) With Monthly Lines Overlaid

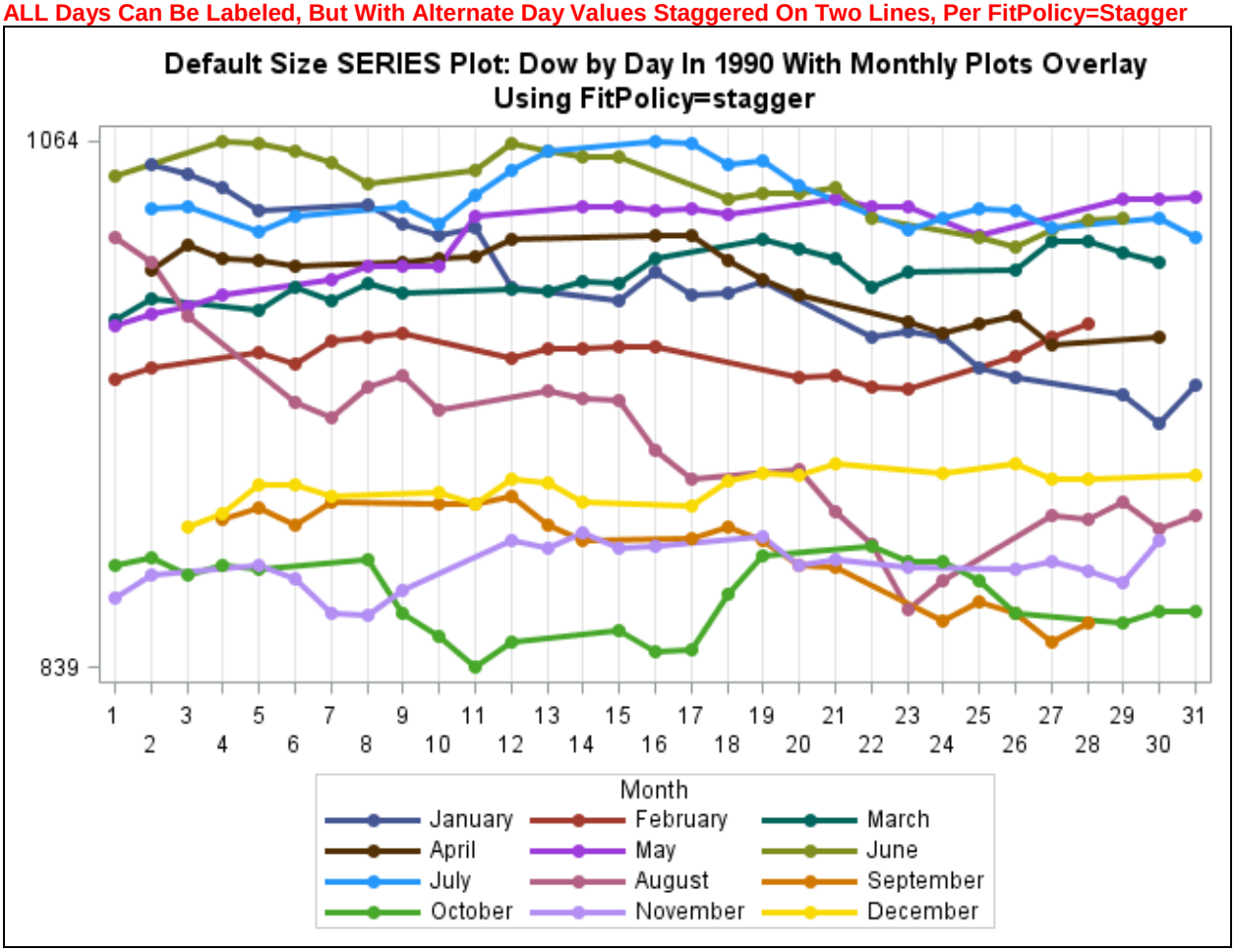
When the image file is widened sufficiently, ALL Days Are Now Labeled, Per Code Specification




```
ods noresults; ods listing close;
ods graphics on / reset=all border=off antialiasmax=2500 tipmax=2500
    width=800px height=600px /* override default size 640px X 480px */
    imagemap=on imagename='SGPLOT_SERIES_By_Day_With_Months_Overlaid_800pxBY600px';
ods html path="D:\@WIILSU Jun2012\Results" (url=none)
    body='SGPLOT_SERIES_By_Day_With_Months_Overlaid_800pxBY600px.html'
    (title='800px BY 600px SERIES Plot - Dow by Day In 1990 With Monthly Plots
Overlay');
title '800px BY 600px SERIES Plot: Dow by Day In 1990 With Monthly Plots Overlay';
proc sgplot data=work.DowByDayIn1990;
  series y=Dow x=Day / group=Month markers markerattrs=(size=7 symbol=circlefilled)
         lineattrs=(thickness=3 pattern=solid);
  yaxis display=(nolabel) values=(&Ymin_1990 &Ymax_1990);
  xaxis display=(nolabel) values=(1 to 31 by 1) grid;
format Month MonthNm.;
format Dow 5. Day 2.;
run;
ods html close; ods listing;
```

Image File (Actual Size: default 640px X 480px) for SGPANEL SERIES Plot By Day With Monthly Lines Overlaid

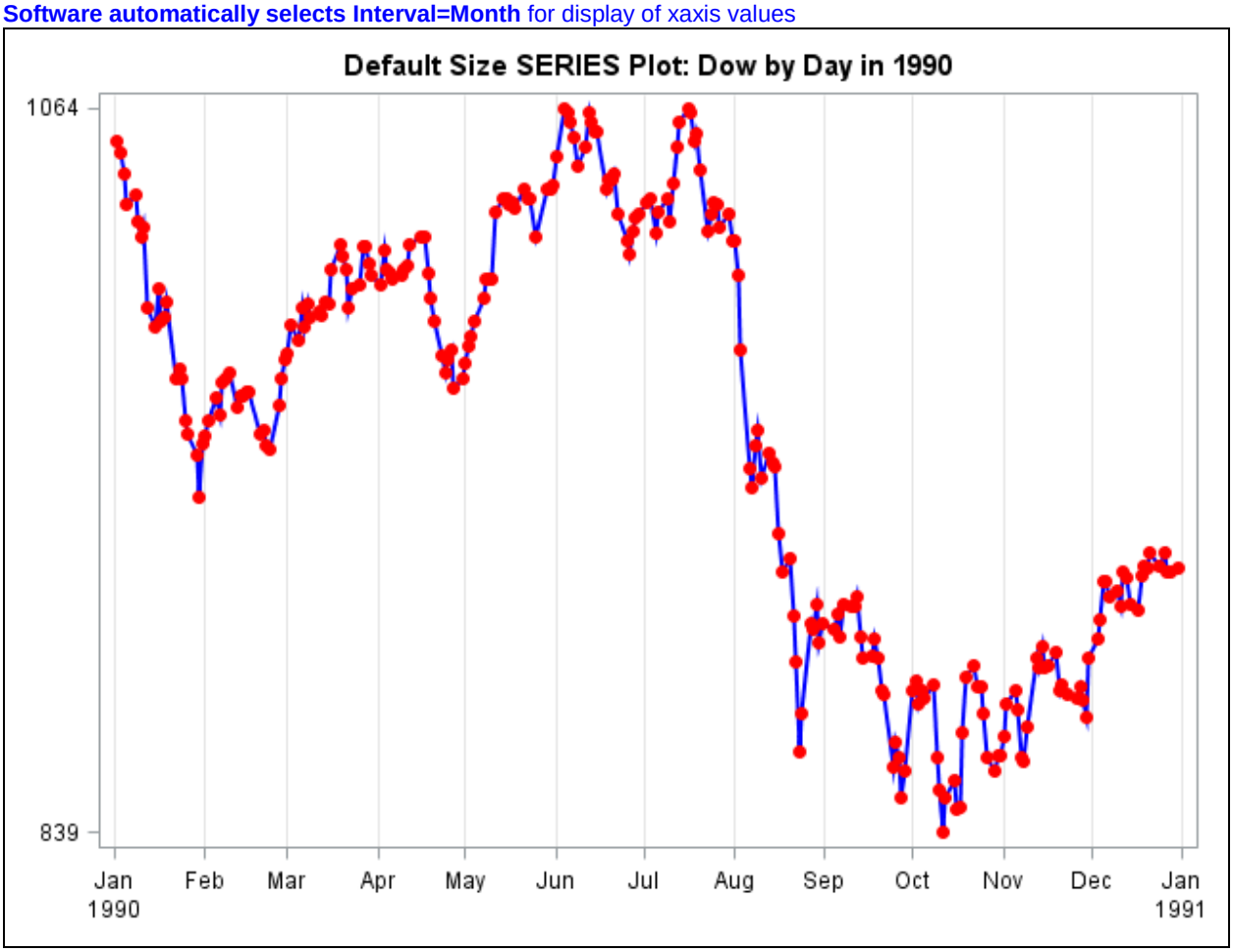
Even for the default width image file,
ALL Days Can Be Labeled, But With Alternate Day Values Staggered On Two Lines, Per FitPolicy=Stagger




```
ods noresults; ods listing close;
ods graphics on / reset=all border=off antialiasmax=2500 tipmax=2500
    imagemap=on
imagename='SGPLOT_SERIES_By_Day_With_Months_Overlaid_DefaultSize_FitPolicyEqStagger';
ods html path="D:\@WIILSU Jun2012\Results" (url=none)
  body='SGPLOT_SERIES_By_Day_With_Months_Overlaid_DefaultSize_FitPolicyEqStagger.html'
  (title='Default Size SERIES Plot FitPolicy=stagger - Dow by Day In 1990 With Monthly
Plots Overlay Using FitPolicy=stagger');
title  height=11pt
       'Default Size SERIES Plot: Dow by Day In 1990 With Monthly Plots Overlay';
title2 height=11pt 'Using FitPolicy=stagger';
proc sgplot data=work.DowByDayIn1990;
  series y=Dow x=Day / group=Month
         markers markerattrs=(size=7 symbol=circlefilled)
         lineattrs=(thickness=3 pattern=solid);
  yaxis display=(nolabel) values=(&Ymin_1990 &Ymax_1990);
  xaxis display=(nolabel) values=(1 to 31 by 1) grid
        FitPolicy=stagger; /* stagger alternate xaxis values on two lines*/
format Month MonthNm.;
format Dow 5. Day 2.;
run;
ods html close; ods listing;
```

Image File (Actual Size: default 640px X 480px) for SGPLOT SERIES Plot By Day

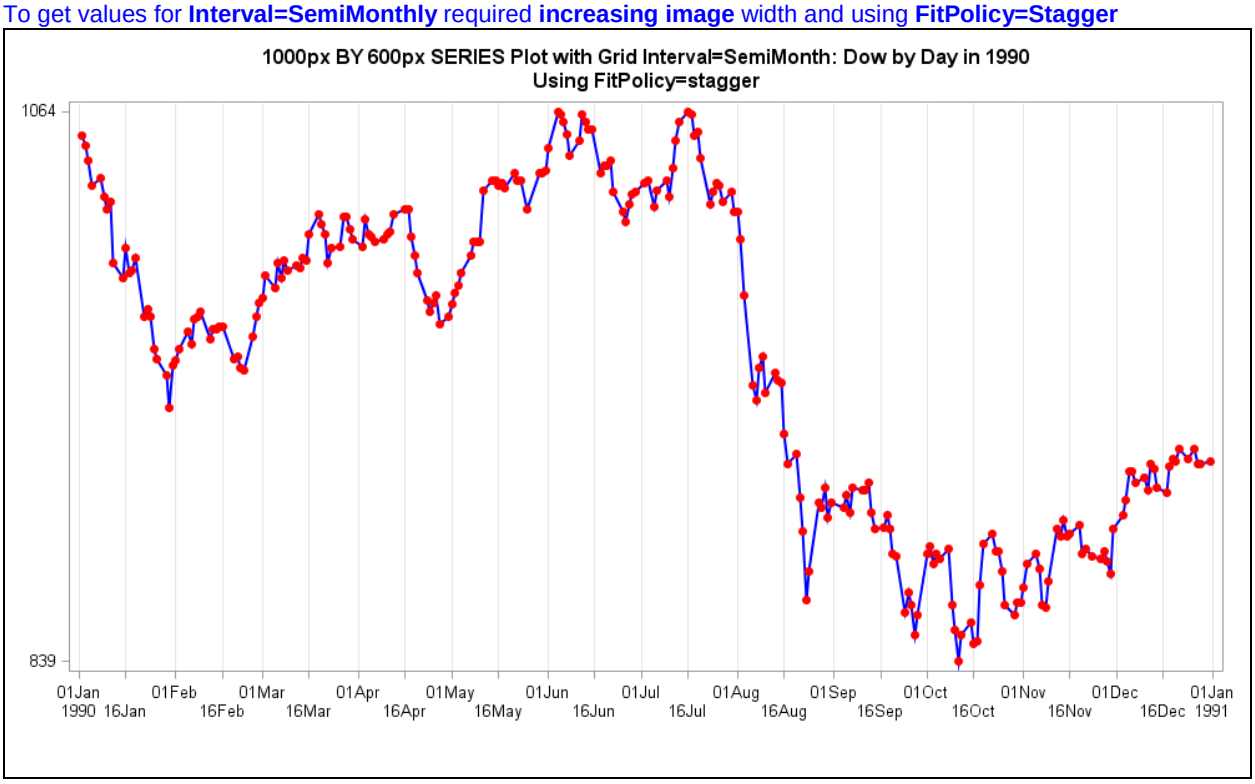
Software automatically selects Interval=Month for display of xaxis values




```
ods noresults; ods listing close;
ods graphics on / reset=all border=off antialiasmax=2500 tipmax=2500
    imagemap=on imagename='SGPLOT_SERIES_By_Day_DefaultSize';
ods html path="D:\@WIILSU Jun2012\Results" (url=none)
    body='SGPLOT_SERIES_By_Day_DefaultSize.html'
    (title='Default Size SERIES Plot - Dow by Day in 1990');
title 'Default Size SERIES Plot: Dow by Day in 1990';
proc sgplot data=work.DowByDayIn1990;
  series y=Dow x=Date /
         markers markerattrs=(size=7 symbol=circlefilled color=red)
         lineattrs=(thickness=2 pattern=solid color=blue);
  yaxis display=(nolabel) values=(&Ymin_1990 &Ymax_1990);
  xaxis display=(nolabel) grid;
format Dow 5.;
run;
ods html close; ods listing;
```

Image File (Reduced Here) for SGPLOT SERIES Plot By Day (Custom Size 1000px X 600px)

To get values for **Interval=SemiMonthly** required **increasing image** width and using **FitPolicy=Stagger**

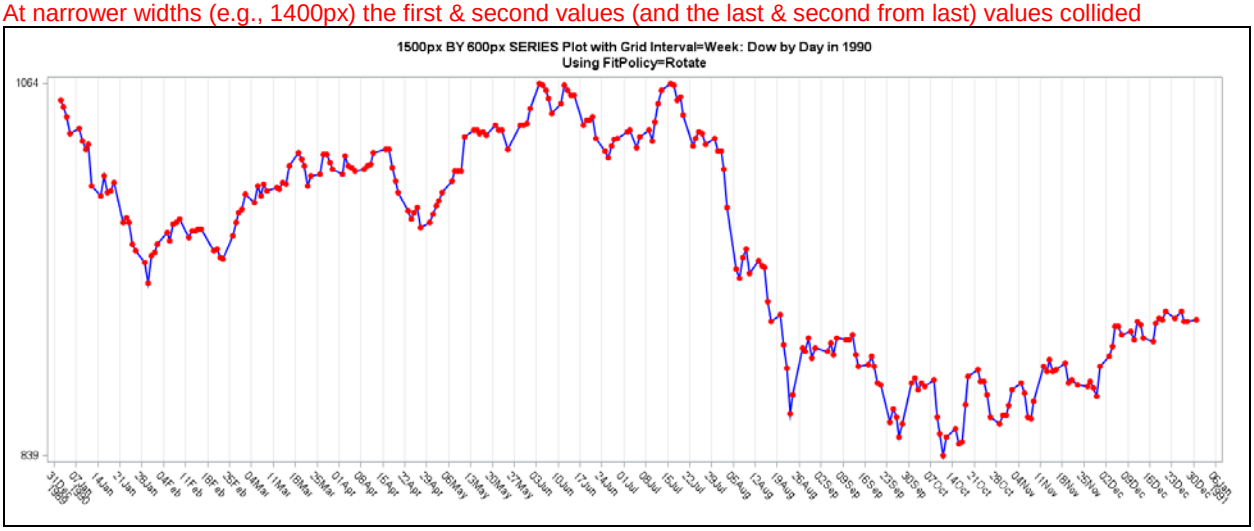



```
ods noresults; ods listing close;
ods graphics on / reset=all border=off antialiasmax=2500 tipmax=2500
    width=1000px height=600px
    imagemap=on
imagename='SGPLOT_SERIES_By_Day_1000pxBY600px_IntervalEqualSemiMonth_FitPolicyEqualSta
gger';
ods html path="D:\@WIILSU Jun2012\Results" (url=none)

body='SGPLOT_SERIES_By_Day_1000pxBY600px_IntervalEqualSemiMonth_FitPolicyEqualStagger.
html'
    (title='1000px BY 600px SERIES Plot with Grid Interval=SemiMonth - Dow by Day in
1990 Using FitPolicy=stagger');
title  height=11pt '1000px BY 600px SERIES Plot with Grid Interval=SemiMonth: Dow by
Day in 1990';
title2 height=11pt 'Using FitPolicy=stagger';
proc sgplot data=work.DowByDayIn1990;
  series y=Dow x=Date /
         markers markerattrs=(size=7 symbol=circlefilled color=red)
         lineattrs=(thickness=2 pattern=solid color=blue);
  yaxis display=(nolabel) values=(&Ymin_1990 &Ymax_1990);
  xaxis display=(nolabel) grid interval=semimonth FitPolicy=stagger;
format Dow 5.;
run;
ods html close; ods listing;
```

Image File (Reduced Here) for SGPLOT SERIES Plot By Day (Custom Size 1500px X 600px)

To get values for **Interval=Week** required **increasing image width** and using **FitPolicy=Rotate**
At narrower widths (e.g., 1400px) the first & second values (and the last & second from last) values collided

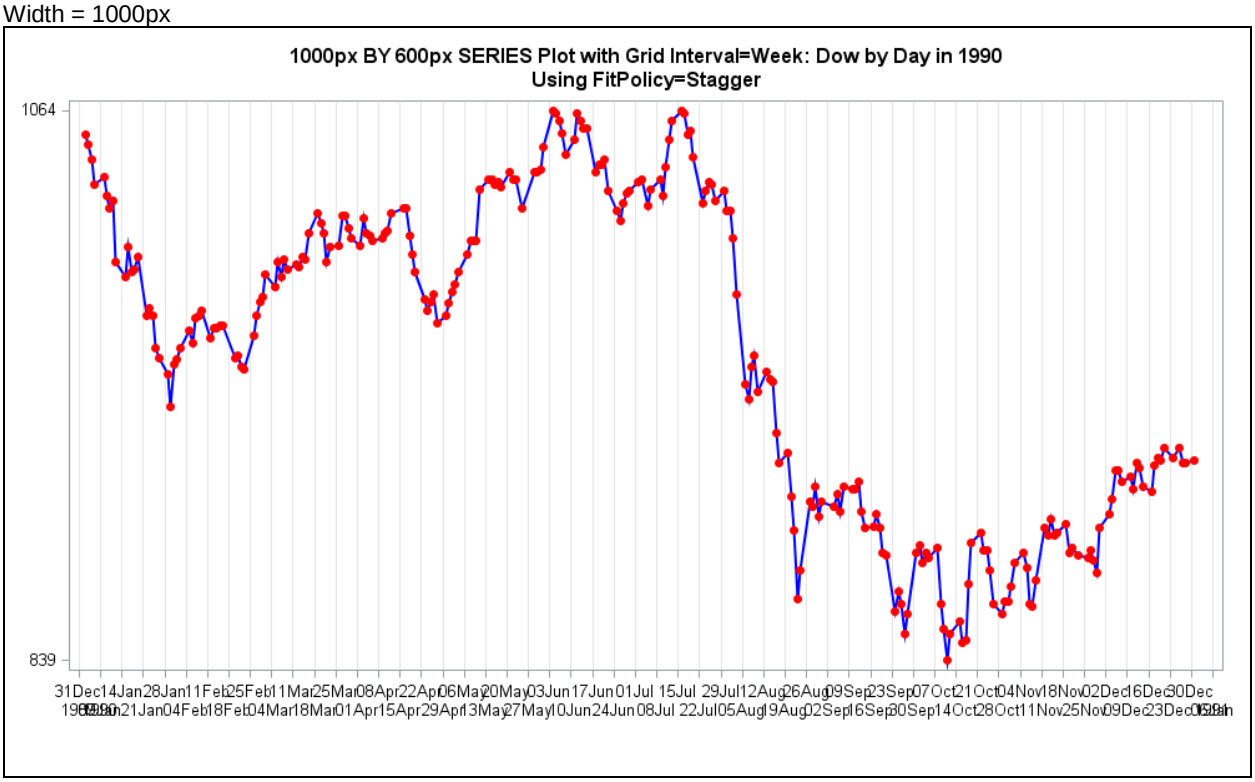



```
ods noresults; ods listing close;
ods graphics on / reset=all border=off antialiasmax=2500 tipmax=2500
    width=1500px height=600px
    imagemap=on imagename=
    'SGPLOT_SERIES_By_Day_1500pxBY600px_IntervalEqualWeek_FitPolicyEqualRotate';
ods html path="D:\@WIILSU Jun2012\Results" (url=none)
 body='SGPLOT_SERIES_By_Day_1500pxBY600px_IntervalEqualWeek_FitPolicyEqualRotate.html'
 (title='1500px BY 600px SERIES Plot with Grid Interval=Week - Dow by Day in 1990');
title  height=11pt
       '1500px BY 600px SERIES Plot with Grid Interval=Week: Dow by Day in 1990';
title2 height=11pt 'Using FitPolicy=Rotate';
proc sgplot data=work.DowByDayIn1990;
  series y=Dow x=Date /
         markers markerattrs=(size=7 symbol=circlefilled color=red)
         lineattrs=(thickness=2 pattern=solid color=blue);
  yaxis display=(nolabel) values=(&Ymin_1990 &Ymax_1990);
  xaxis display=(nolabel) grid interval=week FitPolicy=Rotate;
format Dow 5.;
run;
ods html close; ods listing;
```

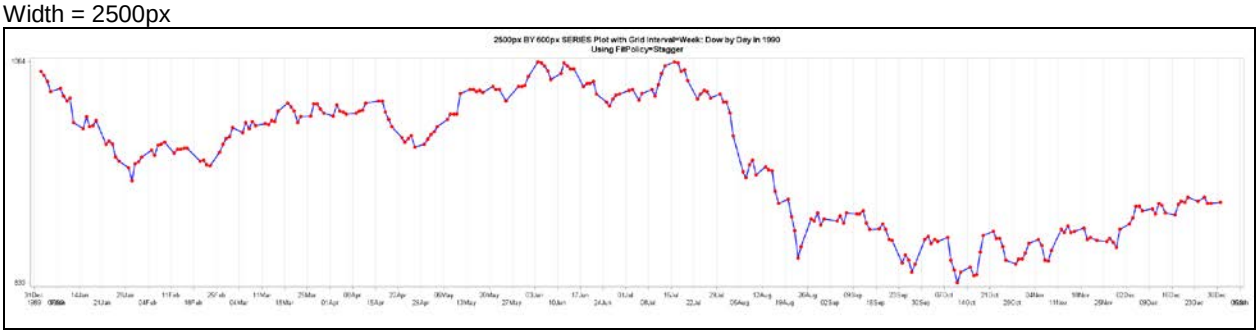
Image Files (Reduced Here) for SGPLOT SERIES Plot By Day With Interval=Week and FitPolicy=Stagger

With Interval=Week there is overlap between xaxis values when using FitPolicy=Stagger .
At greater widths (at least as wide as 2500px) overlaps still persist,
at least for the first & second values (and the last & second from last) values
Failure to recognize Stagger overlap for Interval=Week and to thin out the values is a known software defect.

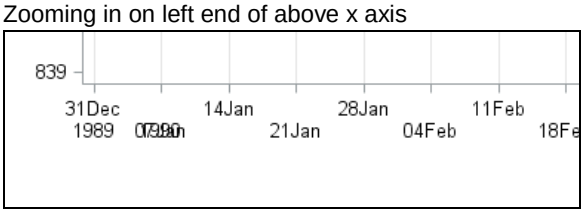
Width = 1000px



Width = 2500px



Zooming in on left end of above x axis



Zooming in on right end of above x axis

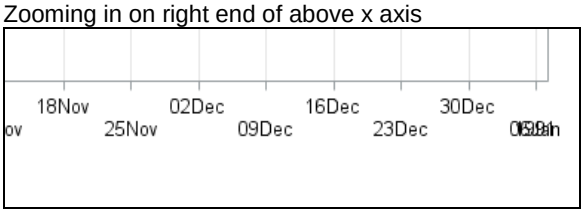
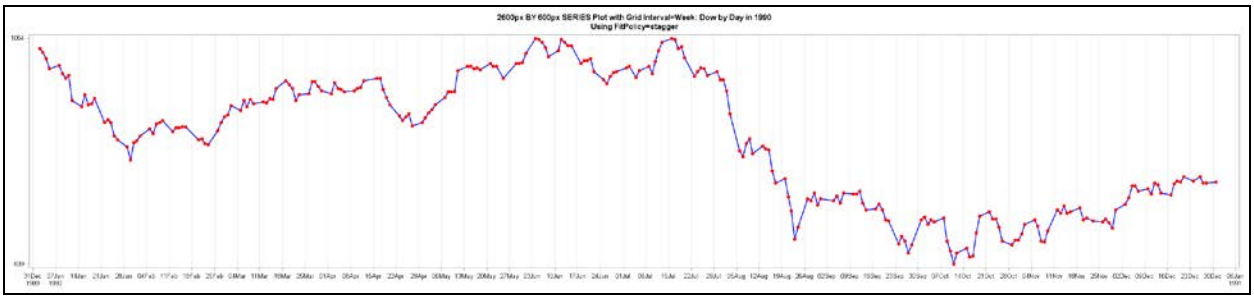


Image File (Reduced Here) for SGPLOT SERIES Plot By Day (Custom Size 2600px X 600px)

Non-overlapping values for Interval=Week and FitPolicy=Stagger requires increasing image width to 2600px, but at this width the software does not need to bother to stagger the values.




```
ods noresults; ods listing close;
ods graphics on / reset=all border=off antialiasmax=2500 tipmax=2500
    width=2600px height=600px
    imagemap=on imagename=
    'SGPLOT_SERIES_By_Day_2600pxBY600px_IntervalEqualWeek_FitPolicyEqualStagger';
ods html path="D:\@WIILSU Jun2012\Results" (url=none)
body='SGPLOT_SERIES_By_Day_2600pxBY600px_IntervalEqualWeek_FitPolicyEqualStagger.html'
 (title='2600px BY 600px SERIES Plot with Grid Interval=Week - Dow by Day in 1990');
title  height=11pt
       '2600px BY 600px SERIES Plot with Grid Interval=Week: Dow by Day in 1990';
title2 height=11pt 'Using FitPolicy=stagger';
proc sgplot data=work.DowByDayIn1990;
  series y=Dow x=Date /
        markers markerattrs=(size=7 symbol=circlefilled color=red)
        lineattrs=(thickness=2 pattern=solid color=blue);
  yaxis display=(nolabel) values=(&Ymin_1990 &Ymax_1990);
  xaxis display=(nolabel) grid interval=week FitPolicy=stagger;
format Dow 5.;
run;
ods html close; ods listing;
```

Simplifying the Appearance of the SERIES Plot

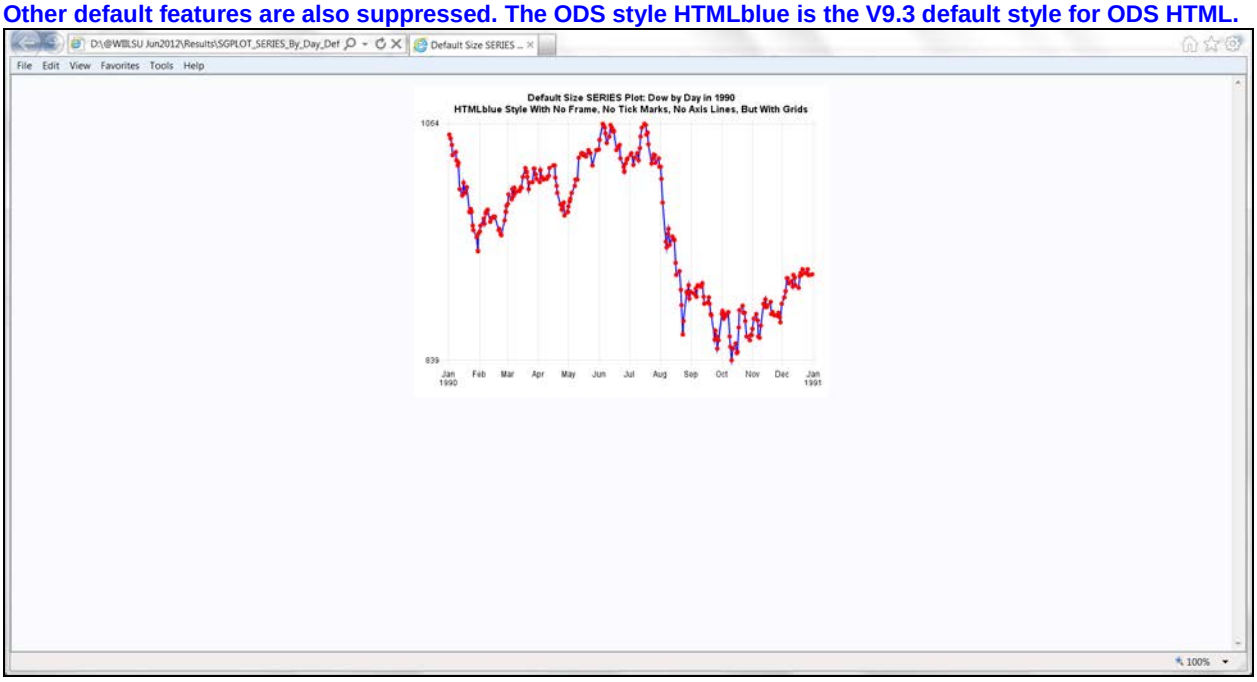
Though necessary tick mark values, and sometimes grid lines, provide communication value, the little tick marks themselves, axis lines, and any framing of the plot area add NO communication value.

In PROC SGPLOT it is easy to turn off the tick marks, and OSTENSIBLY, the axis lines, it is cumbersome to remove the frame. With the frame still present, the suppressed axis lines are overlaid with the four-sided frame. in which case the absence of axis lines is not apparent.

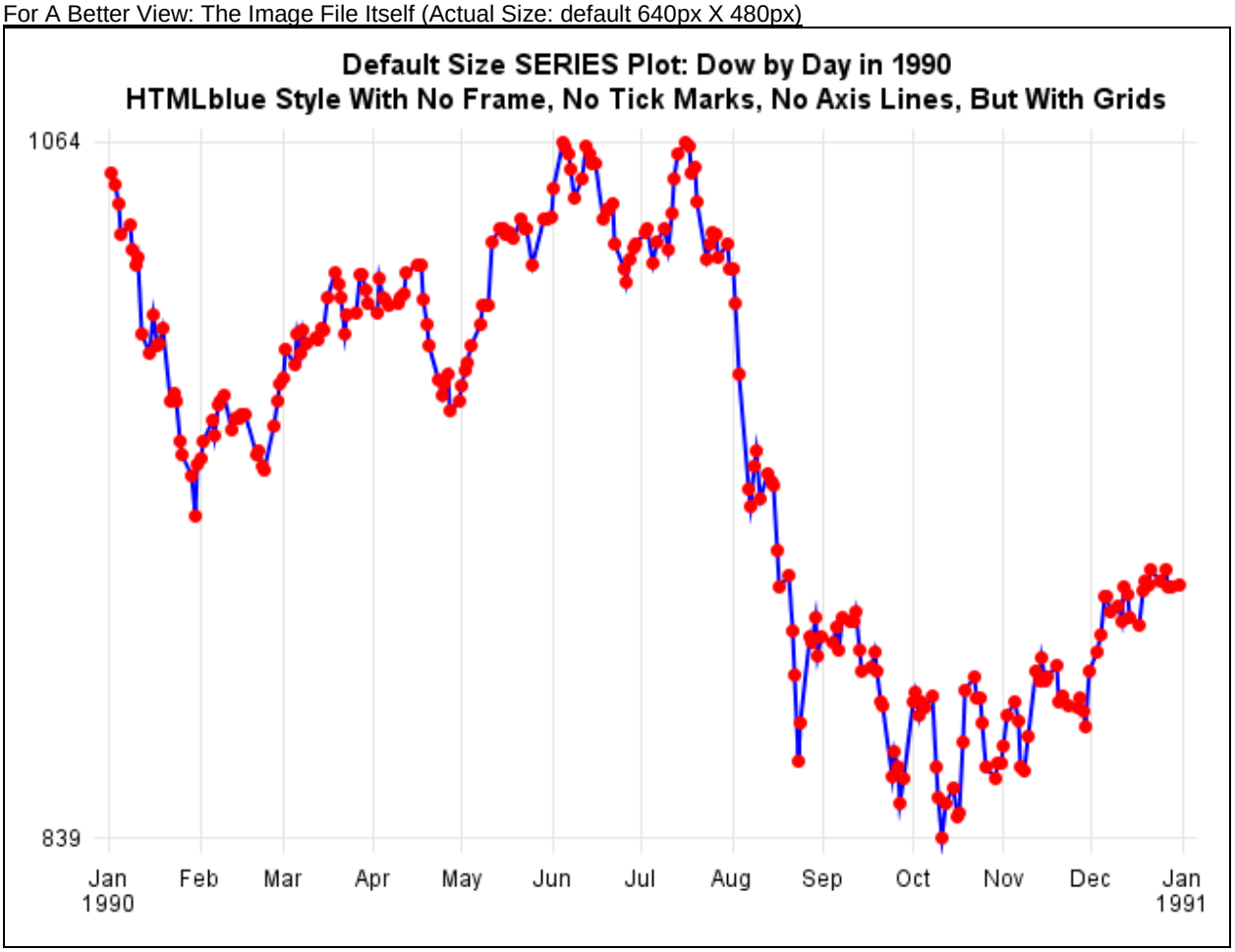
It is my understanding that in SAS Version 9.4 it will be possible to remove the frame around the graph display area with an option, so that it will no longer be necessary to create a customized style to accomplish that. That capability is not be confused with **ODS GRAPHICS ON / BORDER=OFF** which is used to turn off the border around the entire image, not around the display area that contains the plot.

Web Page for SGPLOT SERIES Plot By Day With Frame Around Plot Display Area Suppressed

Other default features are also suppressed. The ODS style HTMLblue is the V9.3 default style for ODS HTML.



For A Better View: The Image File Itself (Actual Size: default 640px X 480px)

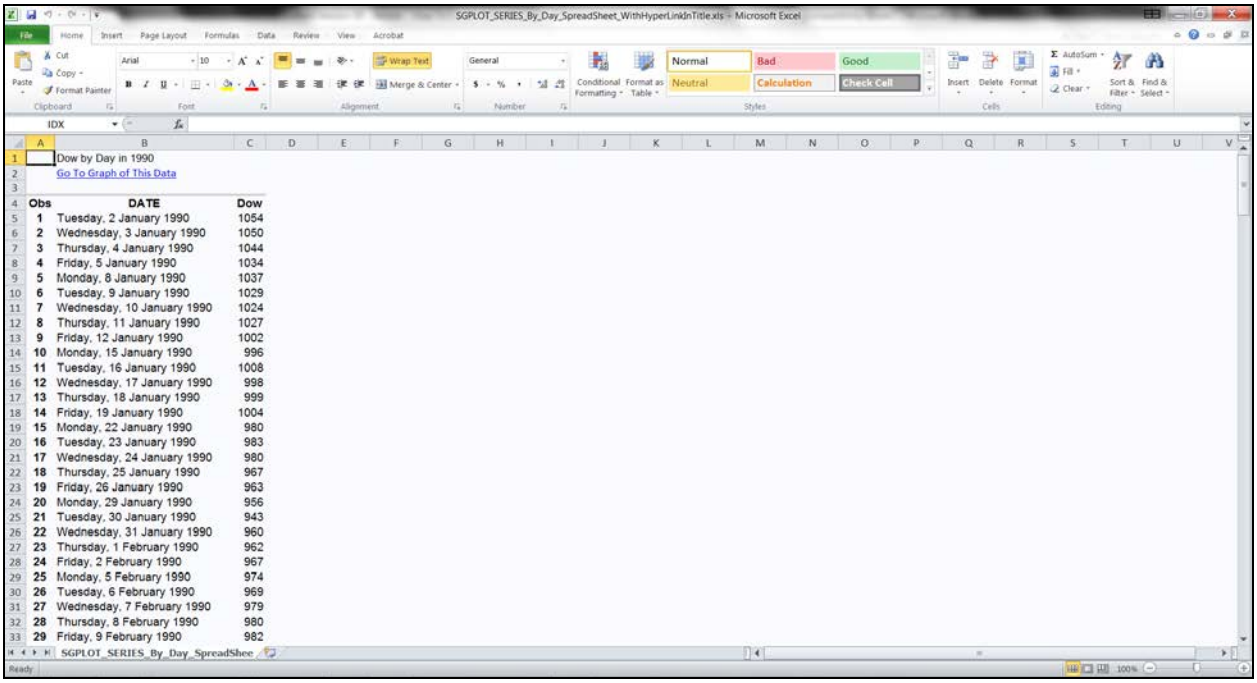


Here is the code used to create the web graph above:

```
proc template;
define style styles.HTMLblueWithNoFrame; /* remove a useless box around the bars */
  parent=styles.HTMLblue;
  class graphwalls / frameborder=off;
end; run;

ods noresults; ods listing close;
ods graphics on / reset=all border=off antialiasmax=2500 tipmax=2500
    imagemap=on
imagename='SGPLOT_SERIES_By_Day_DefaultSize_HTMLblueStyle_WithNoFrame';
ods html path="D:\@WIILSU Jun2012\Results" (url=none)
    style=styles.HTMLblueWithNoFrame
    body='SGPLOT_SERIES_By_Day_DefaultSize_HTMLblueStyle_WithNoFrame.html'
    (title='Default Size SERIES Plot - Dow by Day in 1990');
title1 height=11pt 'Default Size SERIES Plot: Dow by Day in 1990';
title2 height=11pt 'HTMLblue Style With No Frame, No Tick Marks, No Axis Lines, But
With Grids';
proc sgplot data=work.DowByDayIn1990;
  series y=Dow x=Date /
         markers markerattrs=(size=7 symbol=circlefilled color=red)
         lineattrs=(thickness=2 pattern=solid color=blue);
  yaxis display=(nolabel noticks noline) grid values=(&Ymin_1990 &Ymax_1990);
  xaxis display=(nolabel noticks noline) grid;
format Dow 5.;
run;
ods html close; ods listing;
```

The Most Communication-Effective and Most Usable Information Delivery: Web-Enabled Graph + Spreadsheet Linked Forwards and Backwards



Dow by Day in 1990

Go To Graph of This Data

Obs	DATE	Dow
1	Tuesday, 2 January 1990	1054
2	Wednesday, 3 January 1990	1050
3	Thursday, 4 January 1990	1044
4	Friday, 5 January 1990	1034
5	Monday, 8 January 1990	1037
6	Tuesday, 9 January 1990	1029
7	Wednesday, 10 January 1990	1024
8	Thursday, 11 January 1990	1027
9	Friday, 12 January 1990	1002
10	Monday, 15 January 1990	996
11	Tuesday, 16 January 1990	1008
12	Wednesday, 17 January 1990	998
13	Thursday, 18 January 1990	999
14	Friday, 19 January 1990	1004
15	Monday, 22 January 1990	980
16	Tuesday, 23 January 1990	983
17	Wednesday, 24 January 1990	980
18	Thursday, 25 January 1990	967
19	Friday, 26 January 1990	963
20	Monday, 29 January 1990	956
21	Tuesday, 30 January 1990	943
22	Wednesday, 31 January 1990	960
23	Thursday, 1 February 1990	962
24	Friday, 2 February 1990	967
25	Monday, 5 February 1990	974
26	Tuesday, 6 February 1990	969
27	Wednesday, 7 February 1990	979
28	Thursday, 8 February 1990	980
29	Friday, 9 February 1990	982

Here is the code used to create the interlinked plot and spreadsheet:

```
ods noresults; ods listing close;
ods graphics on / reset=all border=off antialiasmax=2500 tipmax=2500
                  width=800px height=600px
                  imagemap=on
imagename='SGPLOT_SERIES_By_Day_800pxBY600px_WithHyperLinkInTitle';
ods html path="D:\@WIILSU Jun2012\Results" (url=none) nogtitle
    body='SGPLOT_SERIES_By_Day_800pxBY600px_WithHyperLinkInTitle.html'
```

```
    (title='800px BY 600px SERIES Plot With Hyperlink In Title - Dow by Day in 1990');
title link='D:\@WIILSU
Jun2012\Results\SGPLOT_SERIES_By_Day_SpreadSheet_WithHyperLinkInTitle.xls'
      '800px BY 600px SERIES Plot With Hyperlink In Title: Dow by Day in 1990';
proc sgplot data=work.DowByDayIn1990;
  series y=Dow x=Date / markers markerattrs=(size=7 symbol=circlefilled color=red)
       lineattrs=(thickness=2 pattern=solid);
  yaxis display=(nolabel) values=(&Ymin_1990 &Ymax_1990);
  xaxis display=(nolabel) grid;
format Dow 5.;
run;
ods html close;
ods html path="D:\@WIILSU Jun2012\Results"
    body='SGPLOT_SERIES_By_Day_SpreadSheet_WithHyperLinkInTitle.xls';
title justify=left "<td COLSPAN=4>Dow by Day in 1990</td>"
justify=left
"<td COLSPAN=4>
<a href='SGPLOT_SERIES_By_Day_800pxBY600px_WithHyperLinkInTitle.html'>
Go To Graph of This Data</a></td>";
proc print data=work.DowByDayIn1990;
var Date Dow;
format Dow 5. Date weekdatx.;
run;
ods html close; ods listing;
```

REFERENCES

1. Bessler, LeRoy. “Graph Macro Tools You Can Use in SAS V9.3, With or Without SAS/GRAPH”, *Proceedings of the June 2012 Wisconsin Illinois SAS Users Conference*. Milwaukee, WI: Software User Services, 2012. Proceedings & Tools CD available from Le_Roy_Bessler@wi.rr.com
2. Bessler, LeRoy. “The Most Communication-Effective and Most Usable Information Delivery”, *Proceedings of the SAS Global Forum 2010 Conference*. Cary, NC: SAS Institute Inc. 2010. http://support.sas.com/resources/papers/proceedings10/231-2010.pdf
3. Bessler, LeRoy. “All the Ways to Display Multiple Trend Charts”, *VIEWS News, Issue 52*. UK: VIEWS International SAS Programmer Community. 2011. http://www.sascommunity.org/wiki/VIEWS_News
4. Bessler, LeRoy and Riley, Alexandra. “All the Ways to Display Multiple Trend Charts: Three Selected Examples and Their Code”, *VIEWS News, Issue 53*. UK: VIEWS International SAS Programmer Community. 2011. http://www.sascommunity.org/wiki/VIEWS_News_backissues
5. Bessler, LeRoy. “Comparison of SAS Graphic Alternatives, Old and New”, *Proceedings of the SAS Global Forum 2012 Conference*. Cary, NC: SAS Institute Inc. 2012. http://support.sas.com/resources/papers/proceedings12/235-2012.pdf
6. Holland, Philip R. “Anscombe's Quartet”, *VIEWS News, Issue 54*. UK: VIEWS International SAS Programmer Community. 2011. http://www.sascommunity.org/wiki/VIEWS_News_backissues

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

LeRoy Bessler PhD
Bessler Consulting and Research
Converting Complexity to Clarity™
Strong Smart Systems™
262-242-1099
Le_Roy_Bessler@wi.rr.com

See Next Page For Appendix 1. Set-Up Code for Time Series Graphs

APPENDIX 1. Set-Up Code for Time Series Graphs

NOTE: The citiday data set is automatically present in the SASHELP data library only if SAS/ETS is licensed. However, you might be able to get five data sets citiday, citiwk, citimon, citiqtr, and citiyr from SAS Technical Support.

```
proc format library=work;
  value MonthNm
  1 = 'January'
  2 = 'February'
  3 = 'March'
  4 = 'April'
  5 = 'May'
  6 = 'June'
  7 = 'July'
  8 = 'August'
  9 = 'September'
 10 = 'October'
 11 = 'November'
 12 = 'December';
run; quit;

libname CITIHELP "D:\SASHELP Sample data sets from SAS ETS";

data work.DowByDayIn1990;
keep Year Month Day Dow date;
format Dow 5.;
set CITIHELP.citiday(keep=date snydjcm where=(snydjcm ne .));
Year = year(date); if Year EQ 1990;
if 1988 LE year LE 1991;
Month = month(date);
Day = day(date);
Dow=round(snydjcm,1);
run;

proc means data=work.DowByDayIn1990 min max noprint;
var Dow;
output out=minmax min=DowMin max=DowMax;
run;

data _null_;
set minmax;
call symput('Ymin_1990'  ,trim(left(put(DowMin  ,5.))));
call symput('Ymax_1990'  ,trim(left(put(DowMax  ,5.))));
run;
```