Data Visualization Tips, Techniques, and Tools: Bessler's Best for SAS Graphs, Web, and Color

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

ABSTRACT

For over thirty years starting with SAS/GRAPH right through today and SAS Version 9.3 with ODS Graphics and the new SG (Statistical Graphics) Procedures, I have been on a quest in pursuit of data insights and visual communication effectiveness, and sharing my ideas and methods with graphics users. My latest deliverables for you are this paper and, available upon email request, some perhaps more easily digested slides to present the highlights of this document, as well as to provide some information and examples from earlier work that it does not cover.

ODS Graphics and the SG (Statistical Graphics) Procedures are now built into Version 9.3 of Base SAS, and entail no additional software licensing charge. However, this technology comes with options and features that require optimization by you to actually get communication-effective output if you want more than simply the default results that come with no added thought or effort.

This paper prescribes principles of communication-effective graphic design, and shows you how to implement them. For certain charts, macros are provided to make getting desired results easy and quick. Less code means less opportunity for coding error. The paper also emphasizes and delivers widely usable solutions that were impossible or less well rendered with SAS/GRAPH. The new tools manifest some unexpected behaviors, which do not even announce themselves in the SAS log. The paper points these out, and shows how to cope with them.

After twenty-five years of sharing my SAS/GRAPH solutions with SAS users, I have become a convert to the new graphics technology. Read on and learn how to get the best from it.

INTRODUCTION

This tutorial shows you easy ways to get beyond taking the defaults, to get better results with minimum time and effort. It covers pie, bar, and line 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 graphs, and shows how to create them. For pie charts and bar charts it has long been possible to display the associated precise numbers as part of the image, but for graphs with plot points on multiple lines and/or with point-dense single lines, it has not. For time series or line charts, the paper demonstrates three ways to address this: (1) ALT text (a.k.a. "data tips", mouseover text, flyover text, hover text, floatover text, tool tips, or pop-up text—which is my name for it); (2) a companion spreadsheet linked to the plot; and (3) a table imbedded in the plot image. The first two options can be used only with web-enabled plots. The third option is available for static plots that are printed or deployed in slides or electronic documents, and for web-enabled plots, and is easier to do with ODS GRAPHICS than with SAS/GRAPH.

My focus 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 1.

For some of the coding examples, macro-based solutions are also provided to make it easier to create the results. For some other solutions, use of macro code is mandatory to make them possible.

UNEXPECTED THINGS ODS GRAPHICS DOES WITHOUT ANY WARNING MESSAGES

1. It can omit tick mark values that are essential (categorical character values, not values of a continuous numeric variable) or in cases where you have explicitly specified the numeric or date values to be listed. This astonishing (and frustrating) feature is described in great detail in the paper, and circumvention options are presented.
2. It creates line breaks in your titles and footnotes when font size is too large for amount of text to fit on one line.

MY WISH FOR ODS GRAPHICS—the ability to force DESIRED line breaks with coding so that specification of text characteristics (font, size, color, etc.) does not have to be repeated in every TITLE statement.
ALSO MISSING IN ODS GRAPHICS—the ability to specify default characteristics for all text in the graph for which characteristics are not explicitly specified in code. Using GTL to create a custom style is extra and complex work.

EFFECTIVE VISUAL COMMUNICATION WITH COLOR

I have written several papers about color, including one award-winning paper. This section is, in part, drawn from my latest, and most comprehensive, *Communication-Effective Use of Color for Web Pages, Graphs, Tables, Maps, Text, and Print* (Reference 3). It was a long paper, too big to simply paste in here, and it is still relevant. Most of it is actually NOT specific to SAS software, and applies to any use of color.

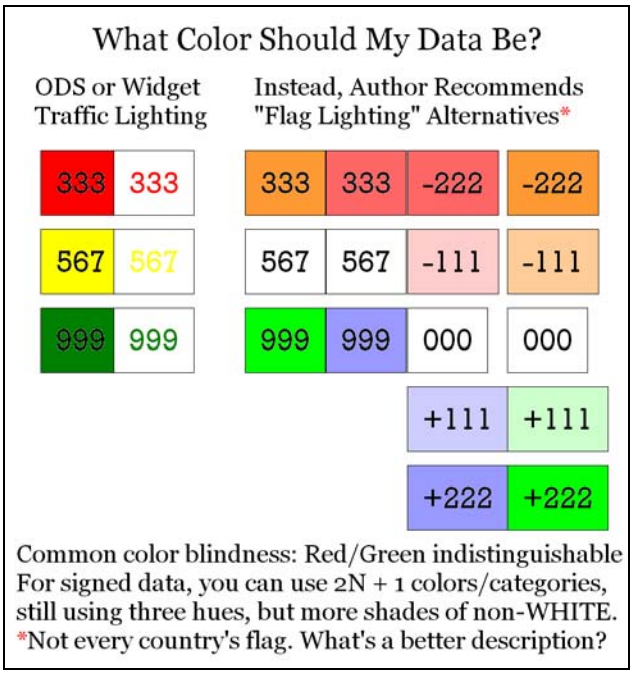
That paper was created with SAS 8.2. There have been enhancements (and a few changes) to how SAS products use color, but none of them diminish the usefulness of that paper.

Below are only six guidelines, excerpted from Reference 3. For more information, please see that paper. Most of the guidelines are simply presented verbatim. A few are commented on here.

1. For Those Who Can't See a Color Difference, There Is None

The commonest color blindness cannot distinguish red and green.

Despite this, there are many well-intended, but misguided, examples of papers explaining how to do "traffic lighting".



2. Use Color To Communicate, Not To Decorate

There are two necessary uses of color to communicate:

- a legend of color squares used to match area fills of pie slices, bars, or geographic unit areas
- a legend of colored line segments, colored markers, or a combination of the two used to match plot lines/points

NOTE: If using colored markers for a plot, use solid-filled markers so that the color is more massive—i.e., visible. In ODS Graphics, all of those have a symbol name that ends with the suffix "Filled", as in CircleFilled, SquareFilled, etc.

There is an optional use of color to communicate:

- Color-coding of data entries in a table, BUT NOT TRAFFIC LIGHTING (as noted in Guideline 1)

The color coding can be applied to the foreground text, i.e., the data entry characters themselves (digits and/or alphabetic characters), or instead to the table cell background. Since the color of text is sometimes hard to discern, it is better to color-code the background, but background and text must always be high contrast—see Guideline 4.

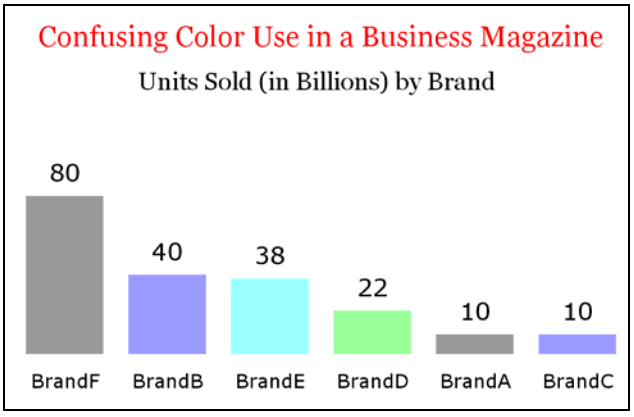
NOTE: Color gradient backgrounds for a graph or web page are always anti-readability, besides being unnecessary.

3. Use of Color Can Confuse, Rather Than Communicate

Viewers attribute significance to your use of color, even when none is intended, proving the wisdom of Guideline 2. Be careful what you do whenever you use color. Using color without a design objective can disorient, confuse, or even mislead the viewer. Failed communication is always the fault of the transmitter, not the receiver.

The content of the example below is different from a magazine illustration that I saw, but its misuse of color is exactly parallel. There is NO relationship between BrandF and BrandA, and none between BrandB and BrandC. So, what does this use of color mean? Absolutely nothing!

Actually, after looking at other color graphs in that magazine article, I realized what had happened. All of the illustrations were limited (why I do not know) to a palette of only four colors. However, this bar chart would have been communication-effective, rather than message-confusing, if rendered in just ONE color.



4. Maximize Color Contrast between Text and Background

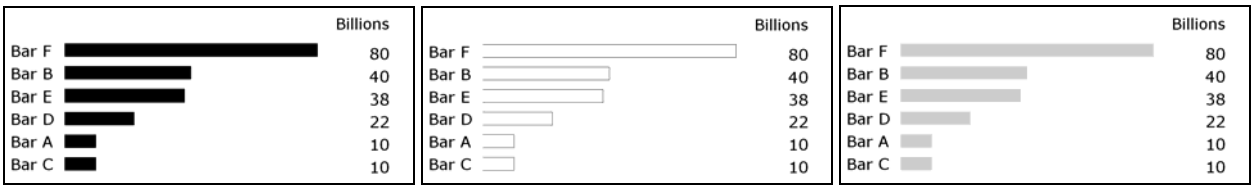
It is no accident that books, newspapers, magazines, and scholarly journals are published with black text on white paper. The second most readable color combination is black and yellow. One of the worst is yellow and white. I have seen traffic-lit tables with (extremely hard to read) yellow numbers in a white box. There are worse choices that come to mind: white on white, black on black, etc. ☺ See Reference 3 for samples of text-background color combinations and for a macro tool to create your own samples.

5. Make Colored Text and Lines Thicker, Colored Plot Point Symbols and Colored Legend Elements Bigger

The color of thin text, thin lines, small plot markers, and small legend elements is difficult to distinguish. Excel pie chart legends suffer from this problem, and all legends with color squares created with ODS Graphics or the SG procedures suffer from this usability defect. See examples of communication-INeffective color legends in Reference 3 and also later in this paper. SAS/GRAPH provides absolute control of legend element size.

6. A Light Color Might Be the Right Color

This pertains to bar charts with area fill, especially ones with numerous bars. Filling the bars with a full-strength color causes the bars to dominate the image. Leaving them empty can cause visual confusion between bars and spaces.



EFFECTIVE USE OF THE WEB FOR DATA VISUALIZATION

Easy Linking of ODS-Created Web Pages (See also the note at the bottom of this page.)

When creating a table or graph, in its TITLE statement or FOOTNOTE statement use this code:

```
LINK="&FileNameToLinkTo..html" "Your Description"
```

Image and Numbers

For certain kinds of graphs, the web is mandatory if you want to get at the detail numbers. A graph enables quick and easy inferences from or revelations about data. However, for reliable inferences or revelations, you also need the precise numbers, not just the picture.

As you will see, for horizontal bar charts and for pie charts, it is always possible to deliver both visuals and numbers in the same graphic image. For a simple, single-line trend chart, with not too many data points, hard (i.e., permanent) annotation of the plot points is usually possible. However, with multiple lines and/or with data points too close together, attempts to annotate suffer overlaps between labels and the plot line(s) and/or between labels and labels.

For trend lines that cannot be annotated, the simplest solution is ALT text (also known as data tips, mouseover text, floatover text, hover text, flyover text, tool tips, or pop-up text). There are several limitations of ALT text:

- You cannot inspect more than one point at a time.
- They can require dexterity to get it to pop up as desired, especially if the plot points are small. When they are too close together, you can wonder: "Which point am I looking at? Is this the point that I want?"
- They are unavailable if you print the web page (or the graph image file it presents).

As a side note, you should be aware that while SAS/GRAPH provides you as much absolute control over content, layout, and formatting of the box of ALT text as you are willing to code, ODS Graphics makes it very easy to get ALT text with minimal coding, but permits no control over what is delivered. ALT text is discussed again later in this paper, and several ODS Graphics examples of its use are presented.

The ability to hyperlink from a trend chart to a table (or, better, to a spreadsheet) makes the precise numbers available. If provided in a spreadsheet (which can be hyperlinked BACK to the chart), the data can then be reformatted, and/or used further in any way desired, with a tool that is universally available and familiar.

Easiest Tip to Implement: Use TITLE=

```
ODS HTML . . . BODY="YourWebPageFileName.html"(TITLE="Your web page title text");
```

If you omit TITLE=, the default is "SAS Output". The title text that you supply above is used in these ways:

1. title bar (title tab) for the browser window (browser tab)
2. default text for Internet Explorer Favorite Bookmark
3. web page browser History list entry
4. captured by search engines

If You Want To Keep Control of Fonts in Graph Titles and Footnotes

If you let ODS put your graph titles and footnotes in the HTML source code that imbeds your graph image file, what it will look like in your or someone else's web browser is unpredictable. The font face might not be what you expect and the size will be whatever. The HTML source in this case contains the name of the font you specified, but it might not be available on the viewing computer. The HTML source will contain an HTML font size (a digit in the range 1 to 7), selected by ODS using an algorithm that you do not control, and the web browser user has access (assuming it the browser is Internet Explorer, but other browsers presumably have something similar or identical) to a choice of five different sizes to rescale the HTML font size (Largest, Larger, Medium, Smaller, Smallest). On Internet Explorer, this size scaling is accessed on the Menu Bar with View > Text Size.

To imbed your title and footnote text in your graph image file, use this:

```
ODS HTML . . . GTITLE GFOOTNOTE;
```

NOTE: If you wish to use LINK= in a title (footnote) to link the web page to another web page, and you are using ODS GRAPHICS (an SG procedure), then you must use NOGTITLE (NOGFOOTNOTE) instead.

Scrolling

Another concern with trend plots is that a time period which is very, very long (i.e., has too numerous data points) cannot be adequately viewed in a single static display area. A web graph can be scrolled horizontally.

The same consideration applies to a horizontal bar chart with too many bars to fit in the display area. A web graph can be scrolled vertically.

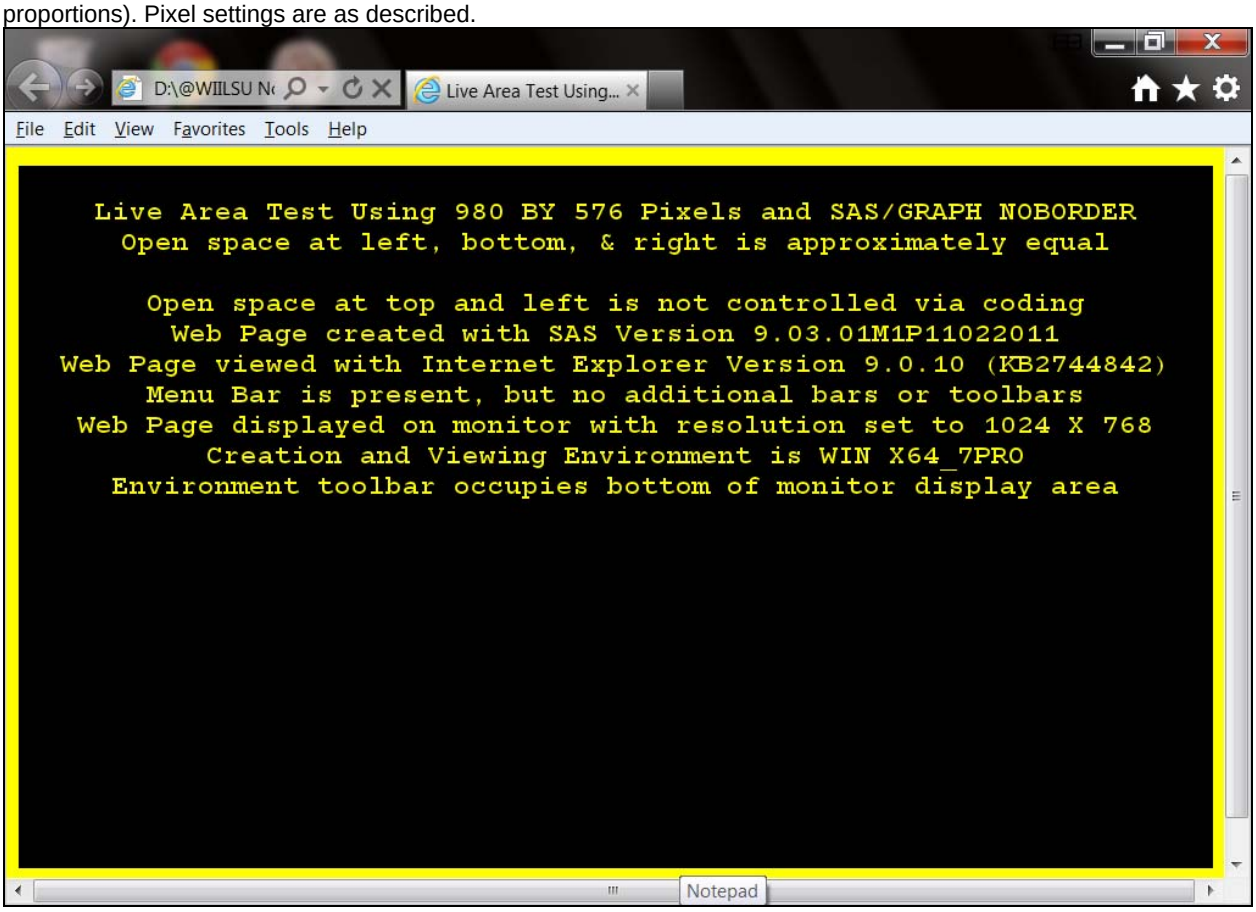
Scrolling Avoidance

Above I emphasized the benefits of scrolling in cases where it is a functional requirement for effective information delivery. However, web design and construction should PREVENT the need for scrolling whenever possible. If unneeded, scrolling is anti-communicative. I.e., it is needlessly denying the viewer the entire picture, and reliable visual comparison of all of its parts simultaneously is impossible. When picking the pixel dimensions of your output web graphs, you need to make some assumptions about the environment where your graphs will be viewed.

Factors that come into play include screen shape, which was fixed and therefore always known before the arrival of wide screen laptops, which I suspect are optimized for viewing movies. All meeting room/site projection screens are still in the traditional proportions of 4 : 3, being the same as those of old-fashioned laptops and CRT monitors.

An obvious factor is monitor screen resolution. On old laptops, a common, if not the commonest, setting is 1024 X 768. Allegedly, the commonest xpixel count is now a bit wider, taking it out of proportion to the screen proportions. I would not design for a non-existent shape.

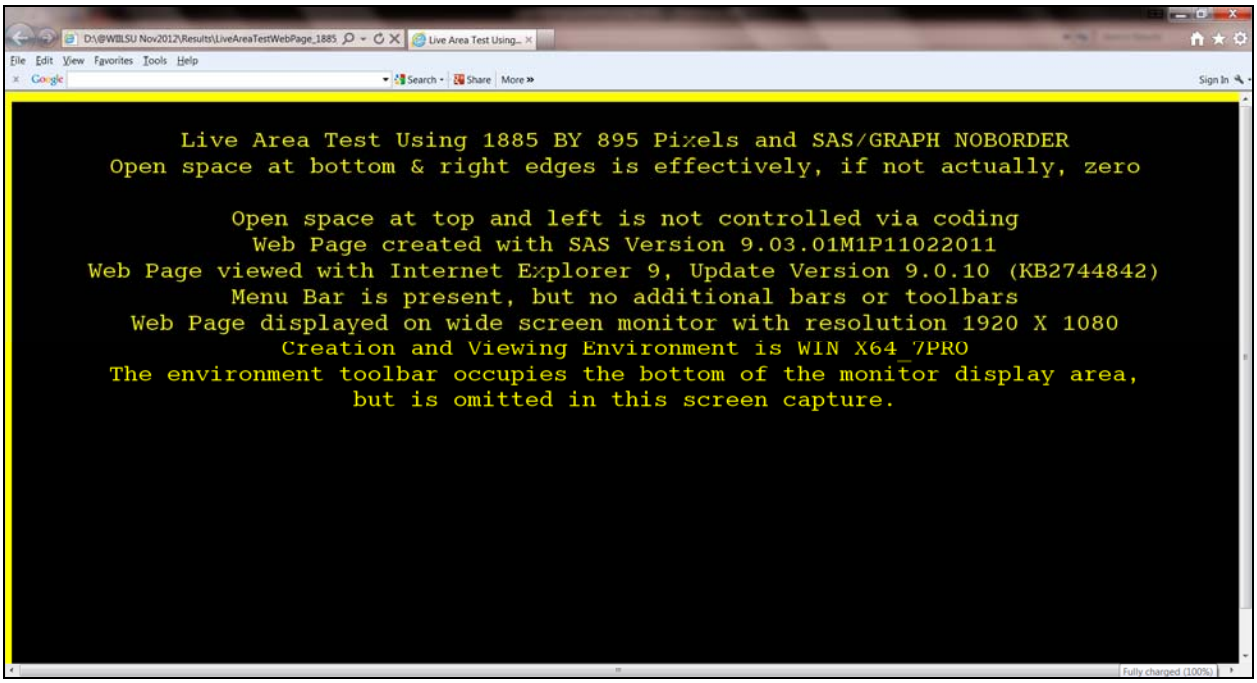
Another factor is web browser. If the statistics I saw are accurate, Google Chrome dominates, followed by FireFox. The space left when you subtract the Windows framing and the browser framing is the available display space, often called the live space or live area. Below is a demonstration, using SAS/GRAPH PROC GSLIDE, of how to determine the live space for a web graph, with my screen resolution set to 1024 X 768 which is the recommended and most commonly used resolution for traditional monitor screens with 4 : 3 proportions (wide screens have 16 : 9 proportions). Pixel settings are as described.



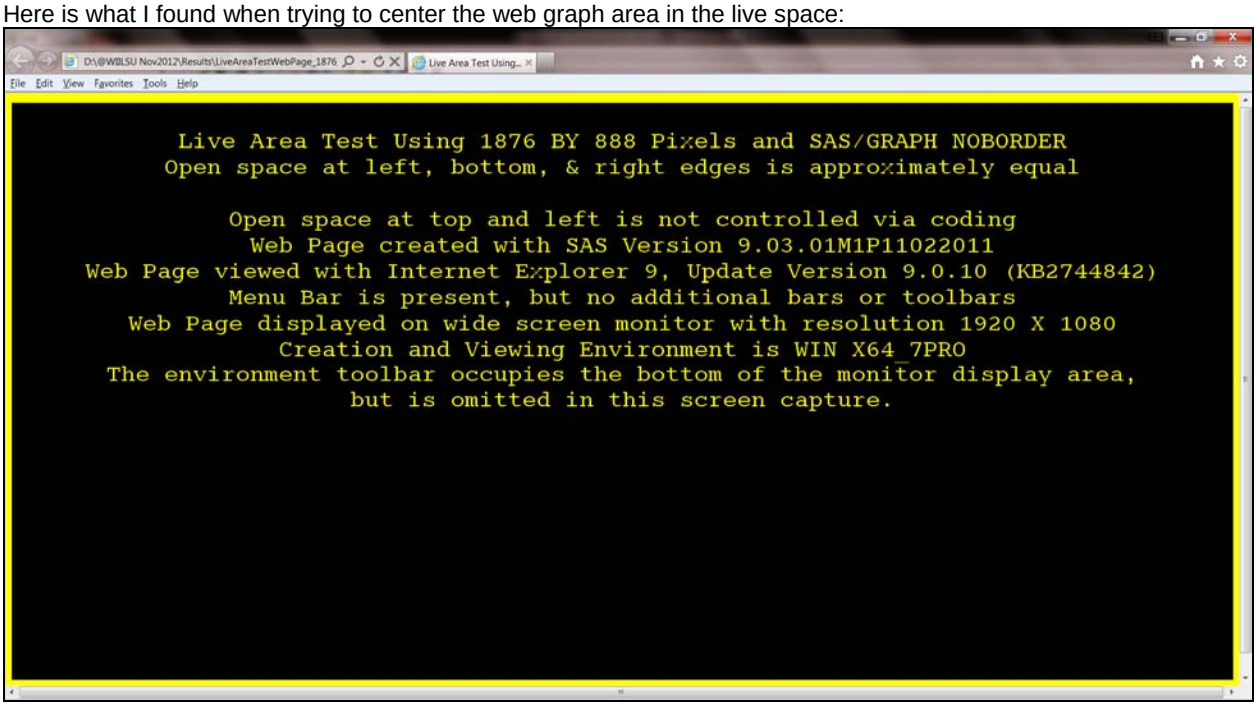
Graph image sizing is a quagmire in my judgment. Your safest policy is to design for the worst case (smallest live space), but you have to decide what that worst case is. I recently did a project to develop web-deployed information delivery where the audience and display environment were known. In another case, I did not have such knowledge, but managed to create a design that drew no complaints, despite being for an important, highly used deliverable.

When you do know your target environment, it is possible and useful to determine the maximum live area. Even if you do not intend to use the maximum, it's important to know what it is so that you have a framework that you can either maximally exploit, or instead within which you can know how much you are safely under-sizing your web graphs.

With my screen resolution set to 1920 X 1080, which is the typical maximum size for a wide screen laptop, below is what I found as maximum graph display capacity:



Here is what I found when trying to center the web graph area in the live space:



If you use GOPTIONS BORDER, you must reduce the GOPTIONS xpixels and ypixels counts to fit it.

If you would like to experiment with determining live space, below is the code I used for the demonstrations above.

NOTE: This cannot be done with ODS Graphics, because it has no PROC GSLIDE analogue.

```
proc template;
edit Styles.Minimal AS Styles.Minimal_CXFFFF00;
style body / background = CXFFFF00; /* browser-safe yellow */
end; run;

%macro LiveAreaTestWebPage(xpixels=,ypixels=,TitleLine2=);

ODS LISTING CLOSE;

ODS HTML PATH="D:\@WIILSU Nov2012\Results" (URL = NONE)
    FILE="LiveAreaTestWebPage_&xpixels._BY_&ypixels._NOBORDER.html"
    (TITLE=" Live Area Test Using &xpixels BY &ypixels Pixels and SAS/GRAPH NOBORDER")
    STYLE=Styles.Minimal_CXFFFF00
    GTITLE GFOOTNOTE;

GOPTIONS RESET=ALL;
GOPTIONS DEVICE=GIF;
GOPTIONS NOBORDER;
GOPTIONS CBACK=CX000000; /* black */
GOPTIONS XPIXELS=&xpixels YPIXELS=&ypixels;

proc gslide;
title1
  height=4 PCT   /* each title text line to be four percent of image height */
  color=CXFFFF00 /* browser-safe yellow */
  " "            /* blank space at top of web page */
justify=center
  "Live Area Test Using &xpixels BY &ypixels Pixels and SAS/GRAPH NOBORDER"
justify=center "&TitleLine2"
justify=center " "
justify=center "Open space at top and left is not controlled via coding"
justify=center "Web Page created with SAS Version &sysvlong4"
justify=center "Web Page viewed with Internet Explorer 9, Update Version 9.0.10
(KB2744842)"
justify=center "Menu Bar is present, but no additional bars or toolbars"
justify=center "Web Page displayed on wide screen monitor with resolution 1920 X 1080"
justify=center "Creation and Viewing Environment is &sysscp &sysscpl"
justify=center
  "The environment toolbar occupies the bottom of the monitor display area,"
justify=center "but is omitted in this screen capture.";
run; quit;

ODS HTML CLOSE;

ODS LISTING;

%mend LiveAreaTestWebPage;

%LiveAreaTestWebPage(
xpixels=1876
,ypixels=888
,TitleLine2=%str(Open space at left, bottom, & right edges is approximately equal));

%LiveAreaTestWebPage(
xpixels=1885
,ypixels=895
,TitleLine2=
    %str(Open space at bottom & right edges is effectively, if not actually, zero));
```

PRINCIPLES OF COMMUNICATION-EFFECTIVE GRAPHICS

I have recently been using *Principia Graphika* as a title for concise handouts on the Principles of Communication-Effective Graphics. They are some of the ideas and techniques that I have been implementing, advocating, and writing/speaking about for more than two decades.

So, what are the Principles? Well, there are a lot of them, previously reported by me elsewhere. In this paper, the focus is on these:

- Deliver image plus precise numbers: image for quick easy inference, precise numbers for reliable inference
- Provide ordering—Show them what's important
- Subset the content where appropriate—Show them what's important
- Provide a reliably usable legend—Seems like something that a user should be able to take for granted as built-in
- Suppress and avoid graphic frills—Let your data talk
- Use color in a communication-effective manner—See the discussion above

Communication-Effective Graphic Design Principle 1: Deliver image plus precise numbers

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 (less frequent 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, whenever they are the acceptable information delivery method. 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.

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 the pop-up data tips that ALT text provides are helpful for any web user without visual impairment.

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 option was previously implemented with SAS/GRAPH (see Reference 2), but it is extended here to ODS GRAPHICS and PROC SGPLOT. However, unlike the case when using SAS/GRAPH PROC GPLOT, **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, if they are very point-dense or multi-line, it can be difficult, if not impossible, to provide annotation for the data points that does not suffer collisions between point labels and/or between lines and point labels. For them, web enablement with ALT text and/or a companion table is essential.

Bar charts and pie charts can be created with all detail in the image area. Some 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 or SERIES statements to make bars or points drillable. URL= specifies a variable to contain the bar- or point-appropriate hyperlink.

Available as a solution for both static plots and web-enabled plots is a table imbedded in the image, or using ODS to package the image of a plot with a report procedure table concatenated below it. ODS Graphics makes table imbedding easy, as is shown in this paper. However, the possibility of compound ODS packaging is a well-known and long-ago well-documented capability, and will not be covered here.

Communication-Effective Graphic Design Principle 2: Provide ordering

One of my favorite titles from past papers that I have written is—

Show them what's important: Solutions for a finite work day in an era of information overload

The default ordering provided by SAS graphics software has always been alphabetical ordering of label text: labels of bars in bar charts, labels of pie slices in pie charts, or labels of legend entries.

If you are doing a horizontal bar chart of measures for, say, all 50 states in the USA, or all of the counties in one of the states, or all of the over 200 countries of the world, and you want the chart to be used as an easy look-up tool, rather than primarily as a tool to quickly identify the entity with the most or least impactful measure, then alphabetical order of the labels is the right choice.

However, if largeness (or smallness) of the measure of interest in a bar chart or pie chart is important, then it is best to order the bars or slices from largest to smallest (or smallest to largest). This technique is even more effective when combined with use of Principle 3.

Less obvious kinds of ordering that I have also used (but not here) for the legend of a multi-line plot is to order the entries by:

- the value of the ending y-value for each line
- the average y-value for each line
- the peak y-value for each line

Communication-Effective Graphic Design Principle 3: Subset the content where appropriate

One of my favorite titles from past papers that I have written is—

Show them what's important: Solutions for a finite work day in an era of information overload

The default action performed by SAS graphics software, rightly, is to use all of the input data that you provide it.

If you are doing a horizontal bar chart of measures for, say, all 50 states in the USA, or all of the counties in one of the states, or all of the over 200 countries of the world, and you want the chart to be used as a comprehensive look-up tool, rather than primarily as a tool to quickly identify the entity with the most or least impactful measure, then presenting all of the possible bars is the right choice.

However, if largeness (or smallness) of the measure of interest in a bar chart, then you can choose to present only a subset of the ranked bars from largest to smallest (or smallest to largest). This technique combines use of Principles 2 and 3.

The idea of subsetting the information was concisely and best described many years ago in a recommendation of Jim White in an article on effective communication with print:

Let part stand for the whole.

It is a fact that often an overwhelming share—80%, 90%, or more—of the total measure of interest is often accounted for by a small number of contributors to the total.

Focus attention where attention is merited, rather than overwhelming the graph viewer with all possible information.

When subsetting the information, it can be useful to also make it easy to get to The Big Picture for anyone who really wants to or needs to see it all. With web-enabled graphs, as is shown later, you can provide a pair of linked-by-a-click graphs: one for the "Top N" (N is your choice of 10, 25, or whatever) and the other for "All".

Other ways to subset the data are with a simple cut-off filter, or (a method demonstrated by me in prior work, but not here) to use only enough ranked observations to account for at least P% of the grand total of the measure of interest. The last method answers this question: "Which of the most impactful of these entities are responsible for, say, 90% of the measure that I am concerned about?"

Many years ago, when I was agonizing over preparing a report for executive management, Kenneth J. Wesley advised me thus:

If it doesn't fit on one page, they won't read it.

Subsetting the data lets you present your data on one paper page certainly, and, if sufficiently subsetted, in one web browser window without scrolling.

Communication-Effective Graphic Design Principle 4: Provide a reliably usable legend

In graphs, legends are typically used to decode the identity of colored plot lines and markers or of color-filled bars or pie slices.

The usability of a color-based legend is related to these insufficiently recognized facts of color communication.

If you want to be able to reliably distinguish colors, then
- **colored text must be thick enough**
- **colored lines must be thick enough**
- **colored plot point markers must be large enough**
- **colored legend samples, whether area fill, line segment, or plot point marker, must be large enough**

ODS Graphics does provide controls for thickness of lines and size of plot point markers. And that thickness and size also carry over into the legend for a plot.

Unfortunately, there are no ODS Graphics controls for the size of the colored sample patches in bar chart legends. SAS/GRAPH provides that support. ODS Graphics does not. This same functional deficiency has always been present in Excel pie chart legends. The lack of control in ODS Graphics is disappointing.

Communication-Effective Graphic Design Principle 5: Suppress and avoid graphic frills

Let your data talk.

ODS Graphics, unlike SAS/GRAPH, somewhat limits the user's ability to inflict needless non-communicative or anti-communicative decoration (e.g., no 3D features for what are really two-variable 2D graphs). However, it does retain the standard graphic paraphernalia from the days when laboratory data was plotted with ink on grid paper. Coding to remove that paraphernalia is presented here later.

Note about Color. Though it might not be thought of as such, a too common, and often counter-productive, graphic frill is color. See this paper's opening section "EFFECTIVE VISUAL COMMUNICATION WITH COLOR".

Why should you care about clutter? Here are miniature excerpts of some of my favorite slides to answer this.

Complexity
Distorts, Impedes, Delays
Communication

Simplicity
- Powerful
- Elegant

Elegant
- everything needed
- only the needed

sparse image
focuses attention

sparse graph
more easily
and
more quickly
interpreted

When the fourth text item above is displayed as a slide, it is a powerful demonstration of what it says.

CLUTTER SUPPRESSION

Before we FINALLY get to some graphs, let me quickly cover the easy how-to technical details of clutter suppression.

Non-communicative graphic elements that I always remove are:

- axis labels that can be in title or subtitle
- axis lines
- tick marks for tick values, especially if using grid lines
- frame around graph detail area (which is not the Border around the entire graph)

An axis of dates, times, or datetimes (unless exotically rendered in a hard-to-interpret format) NEVER needs a label.

Coding methods to accomplish removal are:

```
yaxis display=(nolabel); /* no axis label */
yaxis display=(noline) ; /* no axis line  */
yaxis display=(noticks); /* no tick marks */

yaxis display=none;      /* no label, no line, no tick marks */
```

Above also apply to xaxis, rowaxis, and colaxis.

```
hbar . . . / . . . nooutline ; /* no bar outline */
```

Above also applies to vbar.

The choice **`display=none`** might seem counterintuitive, but, if you have a bar chart with each bar end labeled with the response value (using the simple option of DATALABEL), there is absolutely No Reason to include a response axis. Nevertheless, this superfluity commonly appears in bar charts.

ODS Graphics automatically encases the bars of a bar chart or the plot area of a plot in a frame. Unlike SAS/GRAPH, in the current version (9.3) of ODS Graphics there is no way to turn off the frame. Until the option is provided, if you wish to suppress the frame, you need to use a customized ODS style. If you do not routinely specify an ODS style for your output, you need to know which default style is being applied by SAS to your ODS output:

HTML – Styles.Default (formerly)
HTML – Styles.HTMLblue (V9.3)
RTF – Styles.Rtf
PDF – Styles.Printer
LISTING – Styles.Listing (NEW)

If you are creating a graph on disk for later use, you need to use the ODS LISTING destination. If your base style is one of the above or one of the non-default choices supported by ODS, to be able to remove the graph work area frame, you need to first create a derived style with code like this:

```
proc template;
define style styles.YourStyleName;
  parent=styles.SomePreExistingStyle;
  class graphwalls / frameborder=off;
end;
run;
```

When creating the graph, you must apply the style like this (for a graph on disk):

```
ODS LISTING GPATH="YourTargetFolderLocation" STYLE=Styles.YourStyleName;
ODS GRAPHICS ON / . . . IMAGENAME="YourDesiredFileName";
  /* your SG procedure code goes here */
ODS LISTING CLOSE; ODS LISTING;
```

NOTE: It is important to always close the LISTING destination (and to reopen it as shown). If you omit the close, the graph will be written to disk, but there can be unexpected effects in the next graph created in the same session.

PIE CHARTS: A Deservedly Popular Tool for Graphic Communication

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

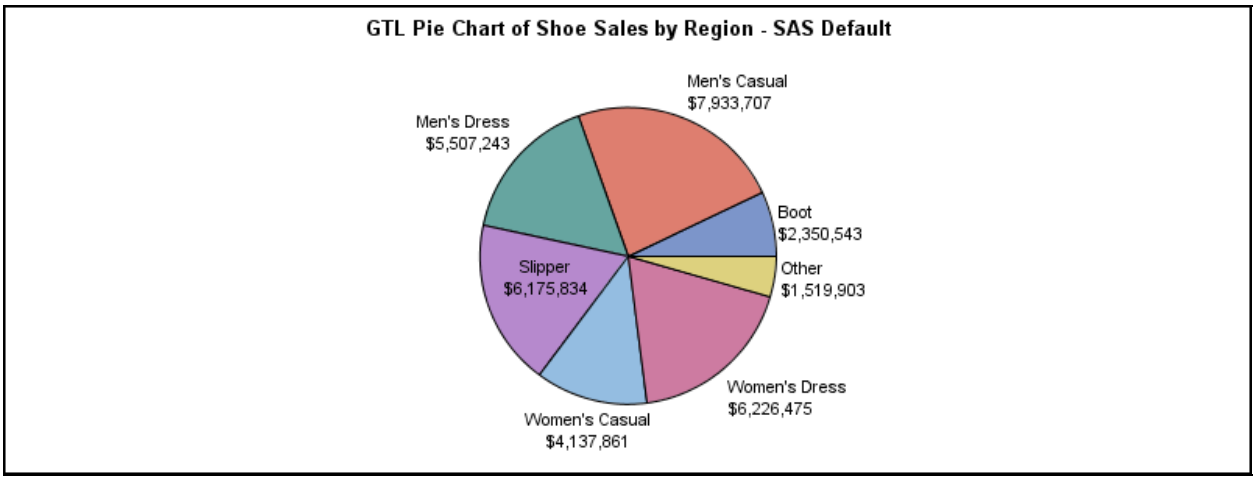
Well, it is a fact that pie charts are one of the commonest graphic delivery instruments, regardless of ideological abstention in some venues and despite misconceptions about what is possible.

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.

NOTE: For my most recent work on pie charts, but in the context of SAS/GRAPH, see Reference 4.

Unacceptable Default Pie Chart: NOT Communication-Effective for four 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'?") But see "Exploiting the Extremes of 'Other'" in Reference 4.

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.

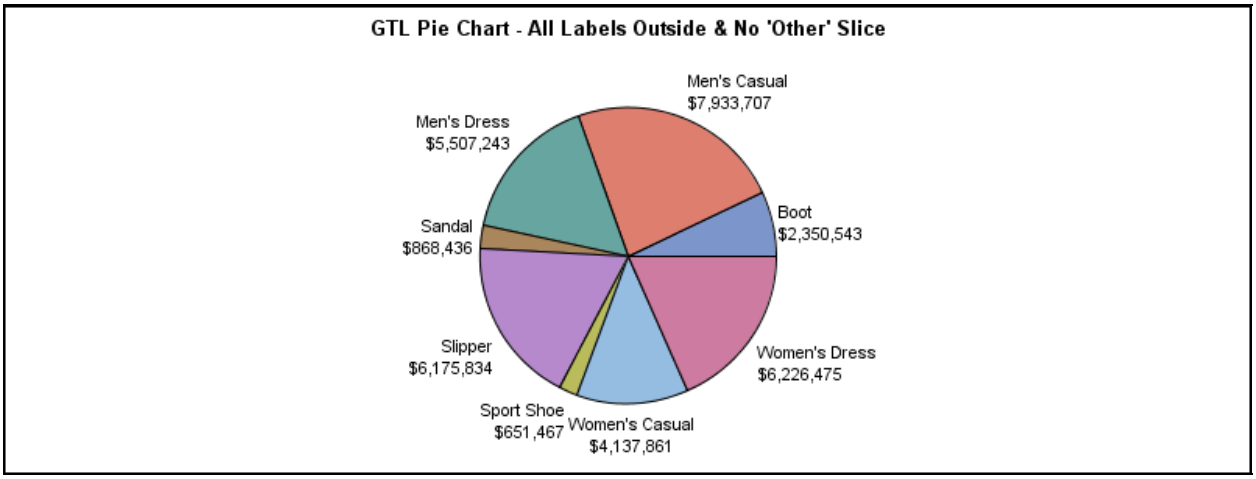
(b) Slices are always more constricted than the while space outside the pie. This increases the difficulty of also supplying the numeric percent of the whole.

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 Nov2012\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;
ods listing close; ods listing;
```

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

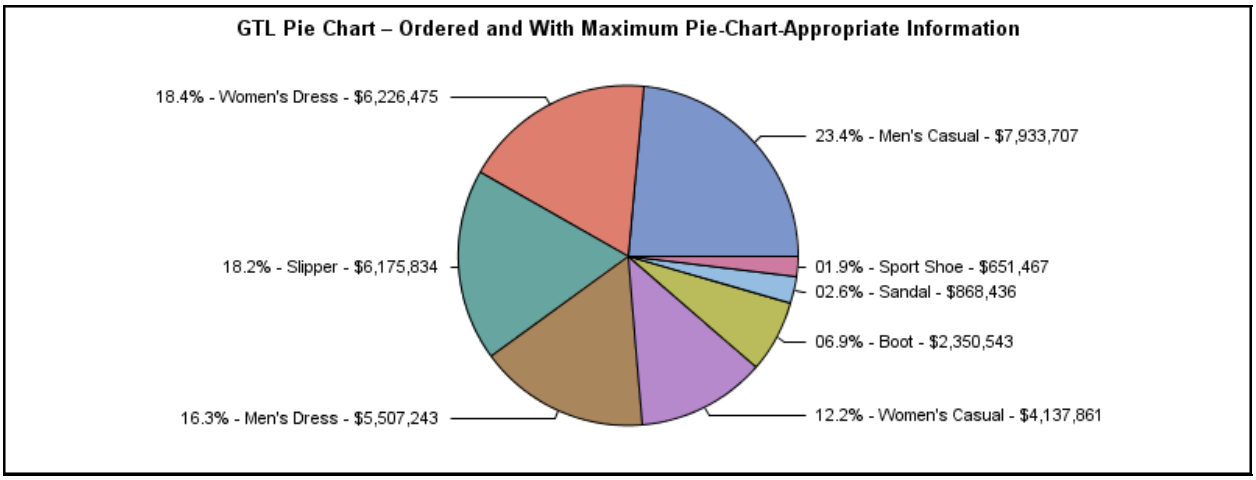


Better: All slice labels are now outside the pie, and all shoe styles are now explicitly identified—no “Other” permitted.

Here is the code, now including two added simple 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 Nov2012\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;
ods listing close; ods listing;
```

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



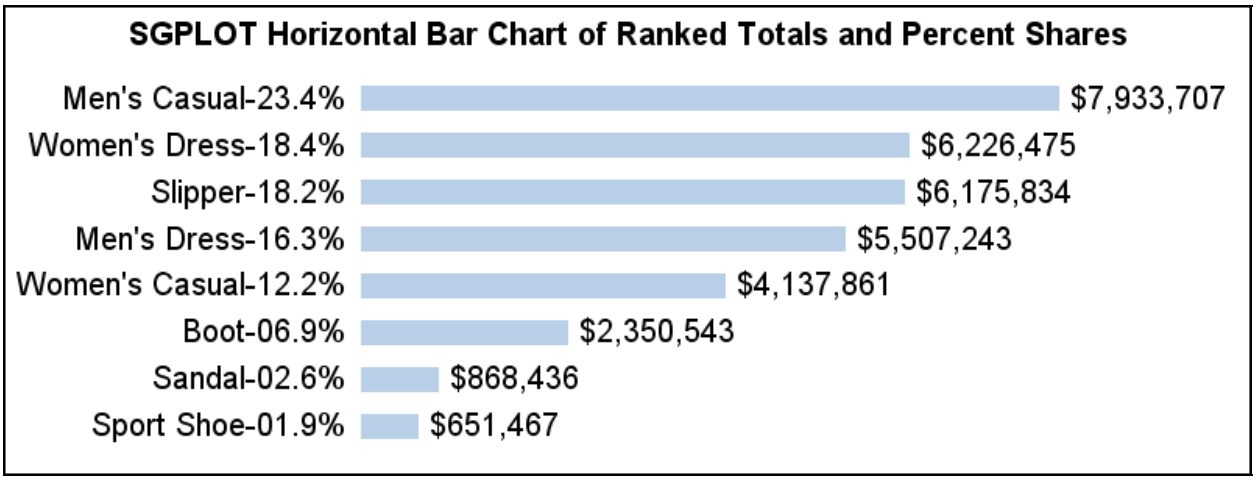
Below is the more complicated code needed to create the pie chart above. **Later in this paper, a macro solution is provided to make this easier and less vulnerable to possible error.**

```
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 Nov2012\Results";

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

proc sgrender data=ToChart template=BesslerBestPieChart31May2012;
run;
ods listing close; ods listing;
```

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 macro solution is provided later in this paper to make the coding 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.ListingWithNoFrame; /* remove a useless box around the bars */
  parent=styles.Listing;
  class graphwalls / frameborder=off;
end; run;

ods listing gpath="D:\@WIILSU Nov2012\Results" style=styles.ListingWithNoFrame;
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;
ods listing close; ods listing;
```

BAR CHART OF SUMS WITH TWO LEVELS OF CATEGORIZATION:

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

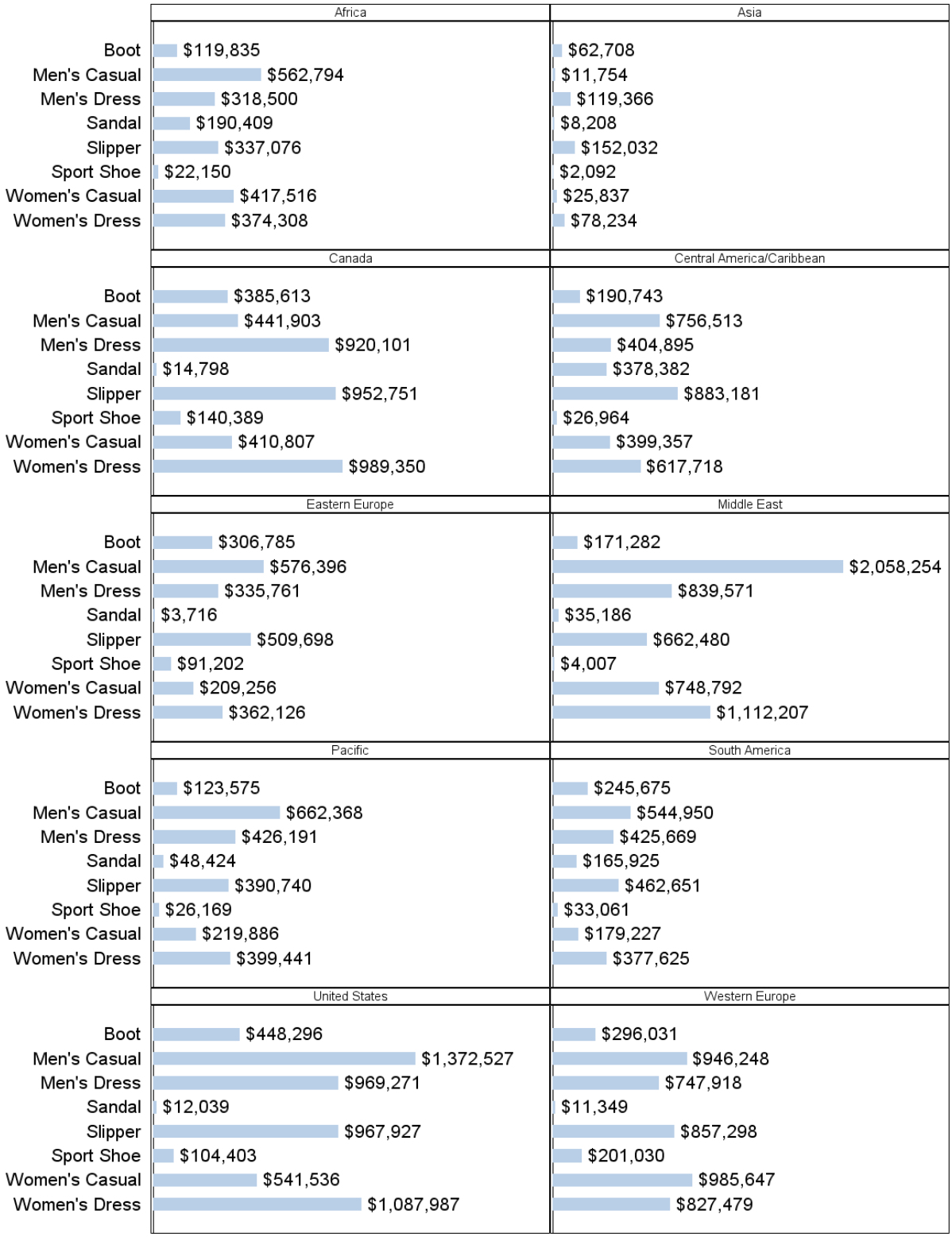
Here is the code used to handle that situation, and to easily create the chart on the following page:

```
ods listing gpath="D:\@WIILSU Nov2012\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
       "Easy To Create, but no control available for size of panel cell header text,";
title3 height=16pt
       "and the shared row header for columns makes CATEGORYORDER unusable.";
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;
ods listing close; ods listing;
```

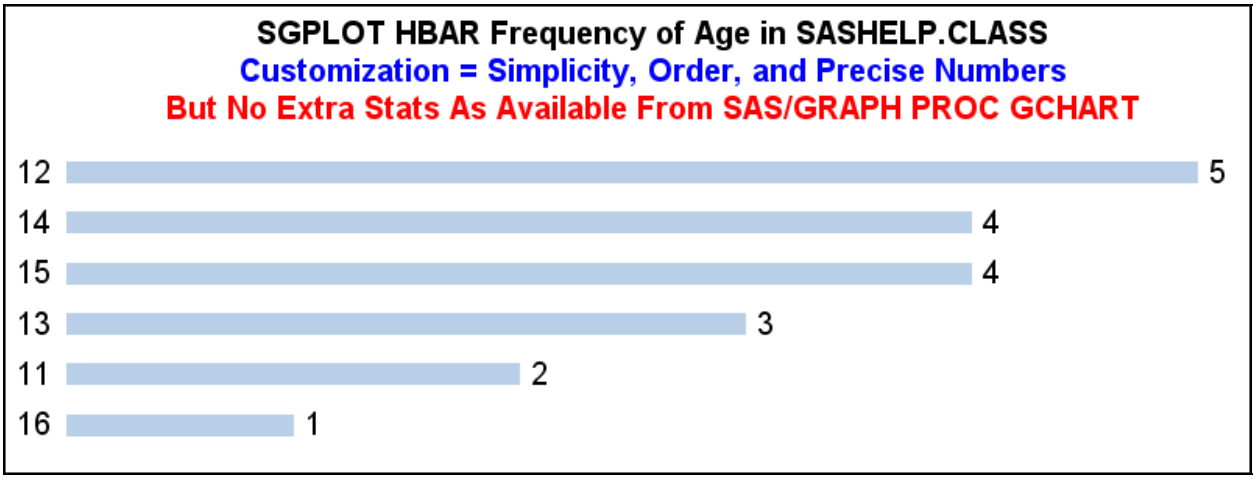
The main disadvantage of this graphic solution is that it is unable to order the bars within each region based upon the Sales By Product. This is due to the fact that the bar labels are provided only at the left margin, rather than for each cell of the panel.

Also, the ability to remove the line that separates the panel cell header from the graphic content of the cell would make it unambiguous as to whether the descriptor pertains to the graph below or the graph above.

SGPANEL Sales by Product within Region in SASHELP.SHOES
Easy To Create, but no control available for size of panel cell header text, and the shared row header for columns makes CATEGORYORDER unusable.



BAR CHART OF FREQUENCY DISTRIBUTION




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

ods listing gpath="D:\@WIILSU Nov2012\Results" style=styles.ListingWithNoFrame;
ods graphics on / reset=all border=on height=300px width=800px
                  imagename='SGPLOThbarFreqChart_Bessler_Customization';
title1 height=16pt "SGPLOT HBAR Frequency of Age in SASHELP.CLASS";
title2 height=16pt
       color=blue "Customization = Simplicity, Order, and Precise Numbers";
title3 height=16pt
       color=red "But No Extra Stats As Available From SAS/GRAPH PROC GCHART";
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;
ods listing close; ods listing;
```

With PROC SGPLOT, it is advantageous to have the counts automatically appended to the bar end without having to do extra programming to annotate the bars. In SAS/GRAPH PROC GCHART, it is easy to get a listing of the values for all of the bars in a column at the right margin, but with a large number of bars, especially when some of them are short, it becomes ambiguous as to which value belongs to which bar. In that case, the extra effort of annotation is necessary, and the column is suppressed. PROC GCHART actually provides three other additional statistics at the right margin: Percent of Total Frequencies, Cumulative Frequency, and Cumulative Percent. You can turn off whatever you do not wish to present. For a large number of bars, doing the needed preprocessing of the data to compute them and providing all of the four statistics via annotation is possible, but probably not desirable. Instead, after the same preprocessing, all of the statistics could be imbedded in expanded bar labels as demonstrated in the Subsetted and Ranked Horizontal Bar Chart later in this paper.

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 5 & 6, 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 7, I provided more information about such plots and tools.

One of the tools omitted in those prior works 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, first developed with SAS/GRAPH in Reference 2 and adapted to ODS GRAPHICS later in this paper, 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 SAS/GRAPH 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, the plot point values in any format to any precision, and times, datetimes, or dates in any format that you prefer (e.g., dates formatted 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, as in this section. 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 is SASHELP.CITIDAY, which is shipped with SAS/ETS. It contains daily history for financial market and other economic information. If you don't have SAS/ETS, you can ask SAS Technical Support 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 index in these web-enabled plots is from the pop-up ALT text, in these graphs 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 are 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 an ODS GRAPHICS SG procedure 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:

1. increase the width of the image if the xaxis has the problem
2. specify **`FitPolicy=Stagger`**
3. specify **`FitPolicy=StaggerRotate`** (first try Stagger, if no help, use Rotate)
4. specify **`FitPolicy=RotateStagger`** (first try Rotate, if no help, use Stagger)
5. 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 THE SERIES STATEMENT, WITH PROC SGPANEL OR PROC SGPLOT

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.

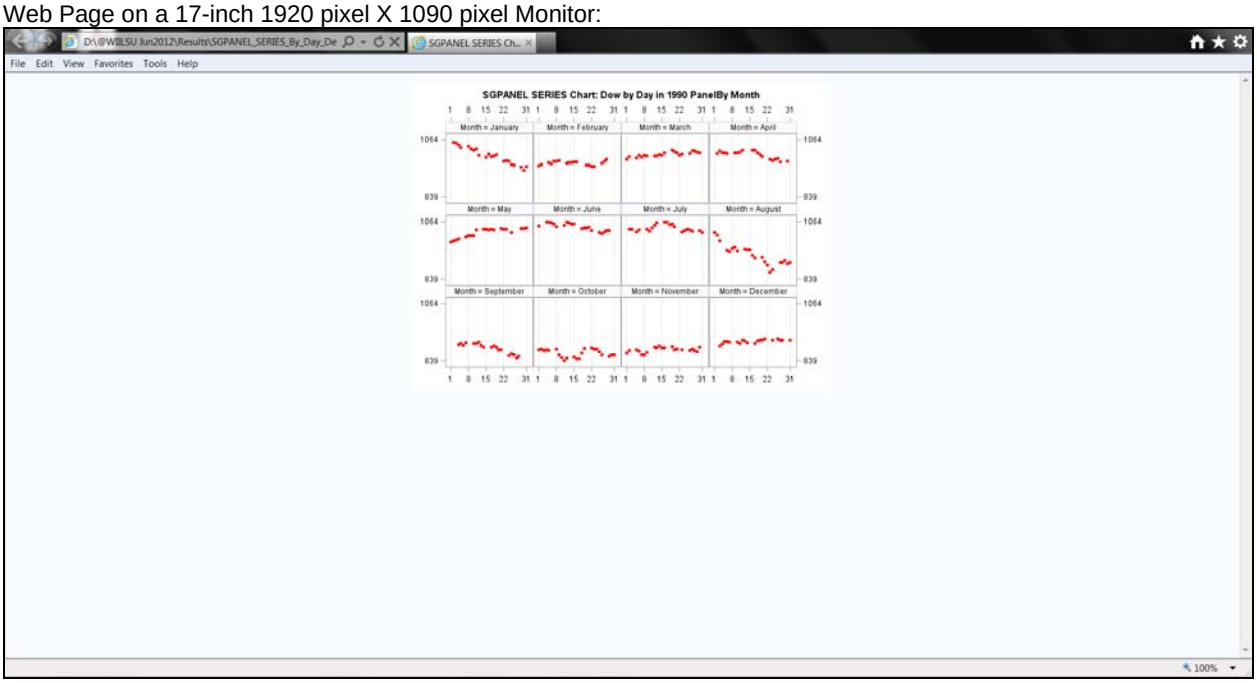
Without GROUP=MONTH, the data is plotted as one line.

The last example in this section 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.

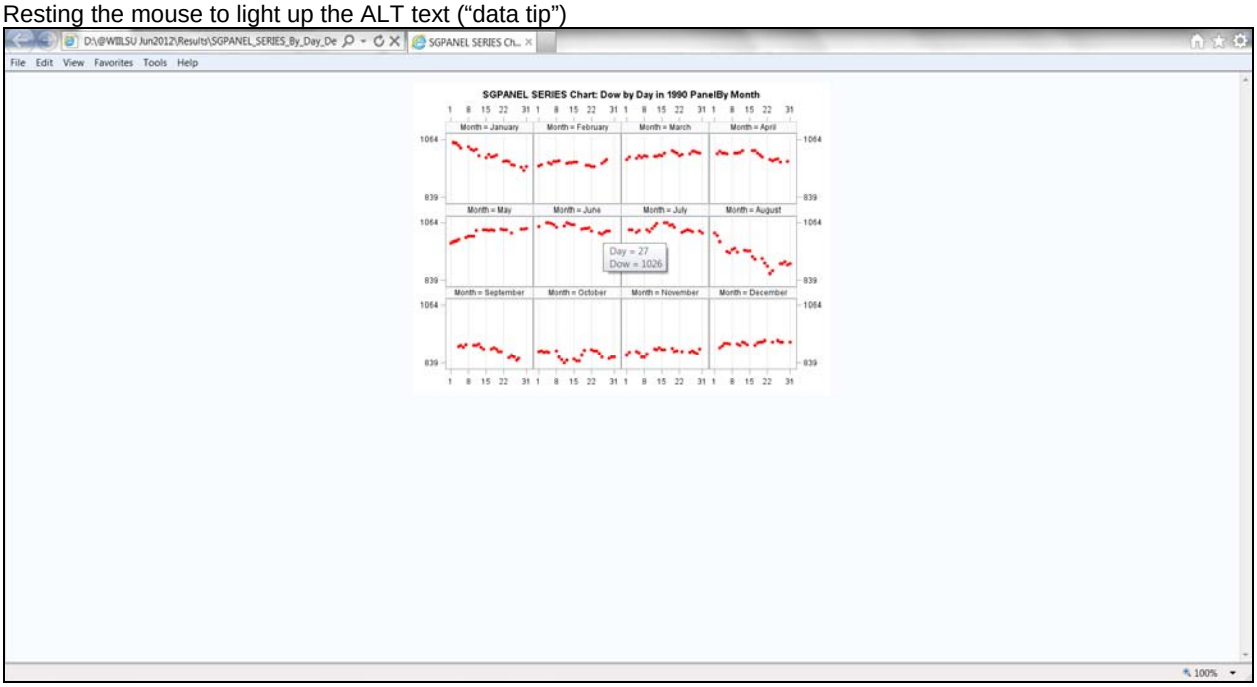
For non-web-enabled plots, the section after this one shows how to imbed the precise y values inside a plot image area by using SGPLOT with a VLINE statement instead of the SERIES statement, which is the focus of this section. The y values can be imbedded as either a table between the plot lines and x axis or as direct plot-point annotation.

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

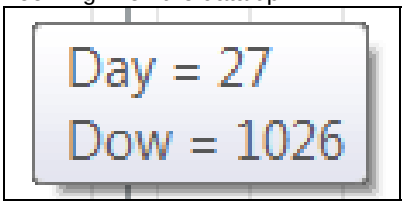
Web Page on a 17-inch 1920 pixel X 1090 pixel Monitor:



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

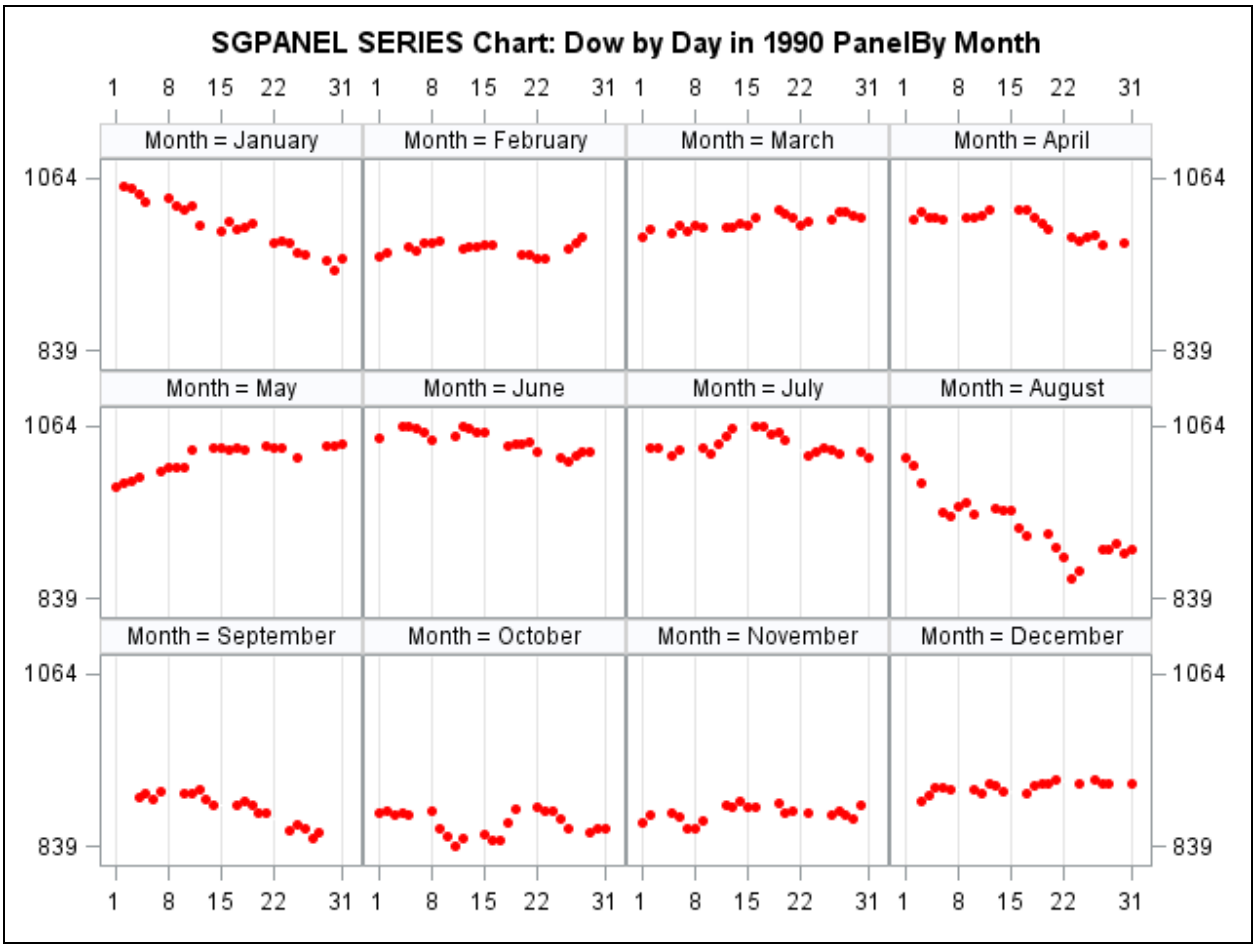


Zooming in on the data tip:



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.

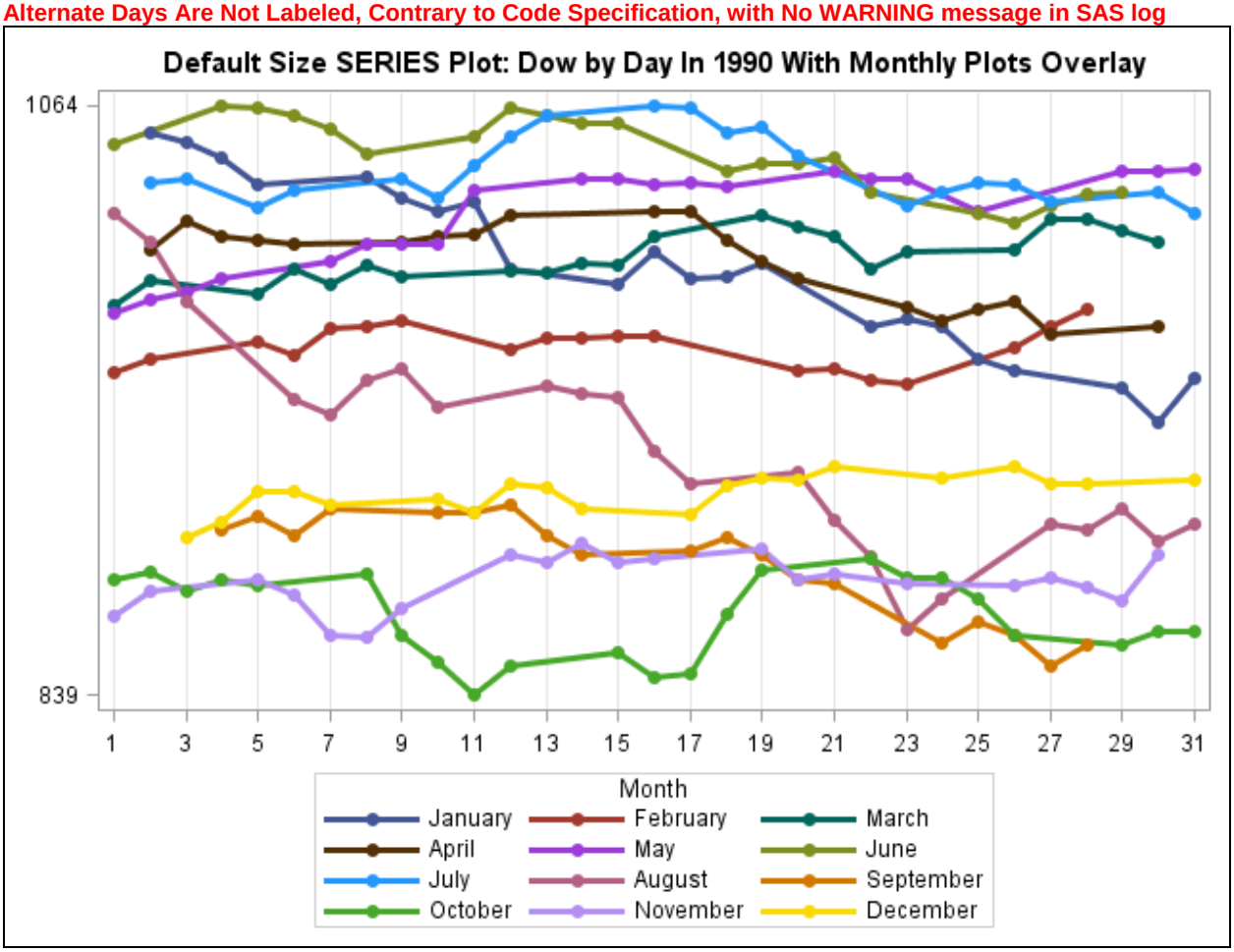


Below is the code to create this web-deployed graph. See predecessor processing in Appendix 1.

```
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 Nov2012\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, with No WARNING message in SAS log

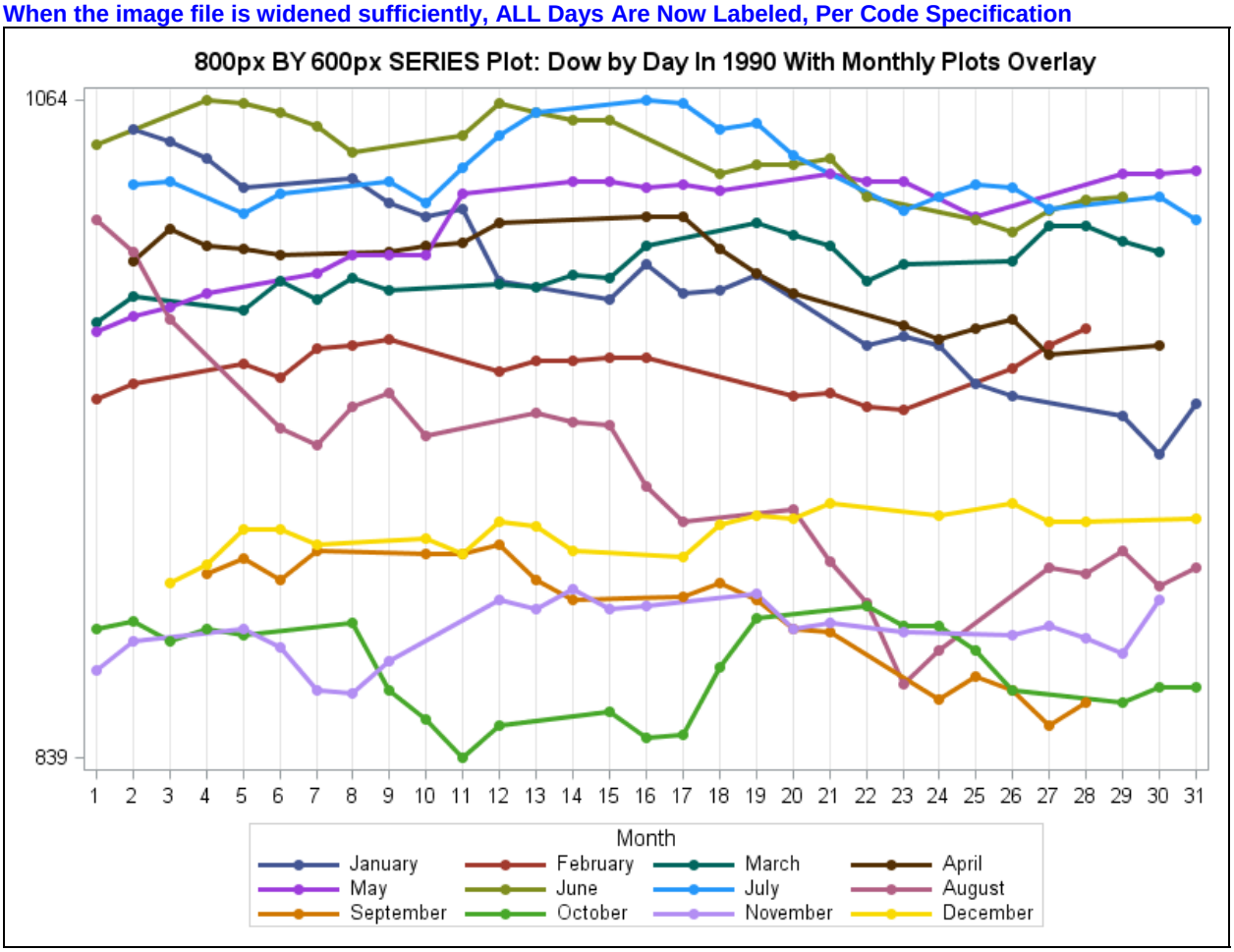


Below is the code to create this web-deployed graph.

```
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 Nov2012\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 (**Shrunk by Microsoft Word To Fit Page Width**) for SGPANEL SERIES Plot By Day (custom 800px X 600px) With Monthly Lines Overlaid

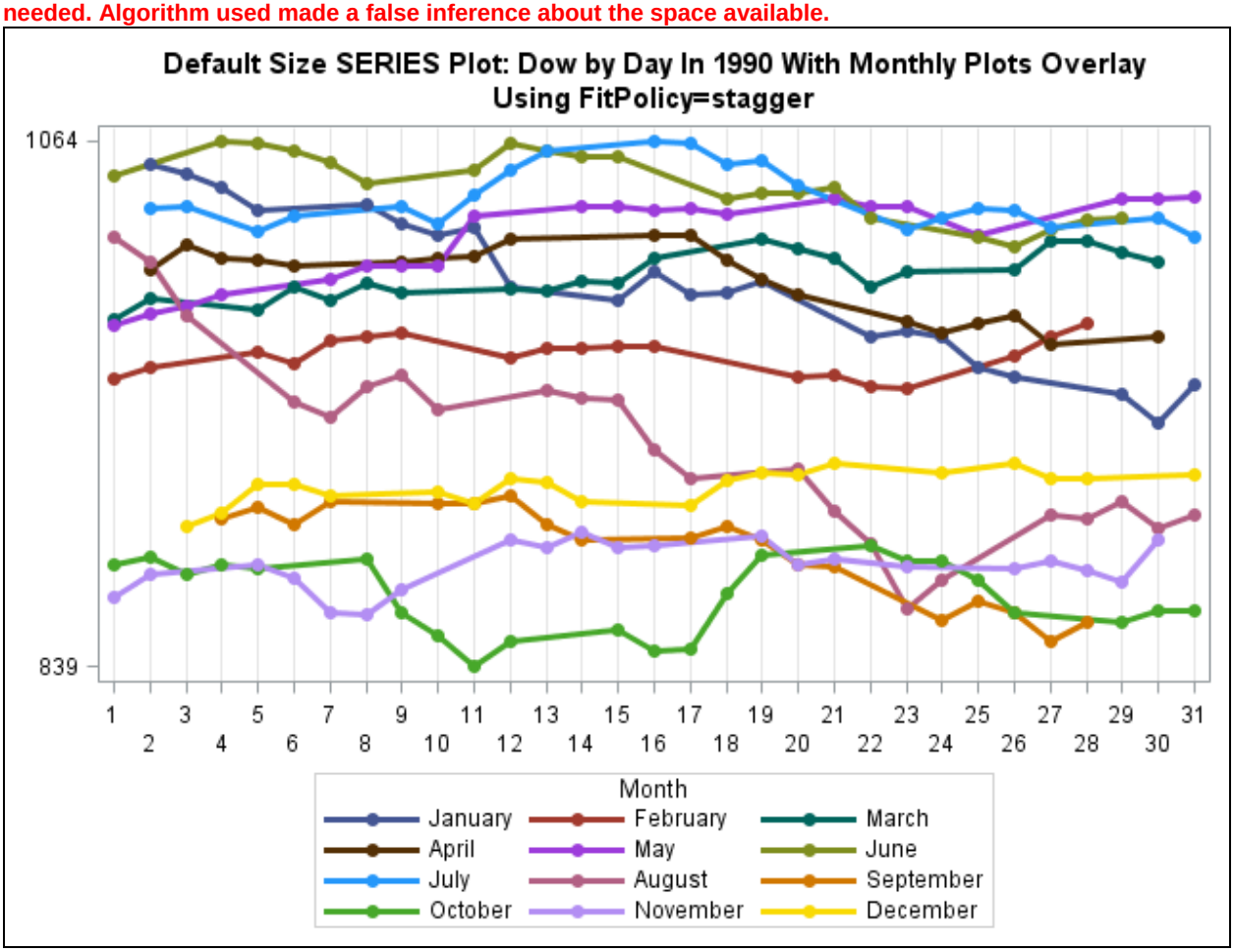
When the image file is widened sufficiently, ALL Days Are Now Labeled, Per Code Specification




```
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 Nov2012\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

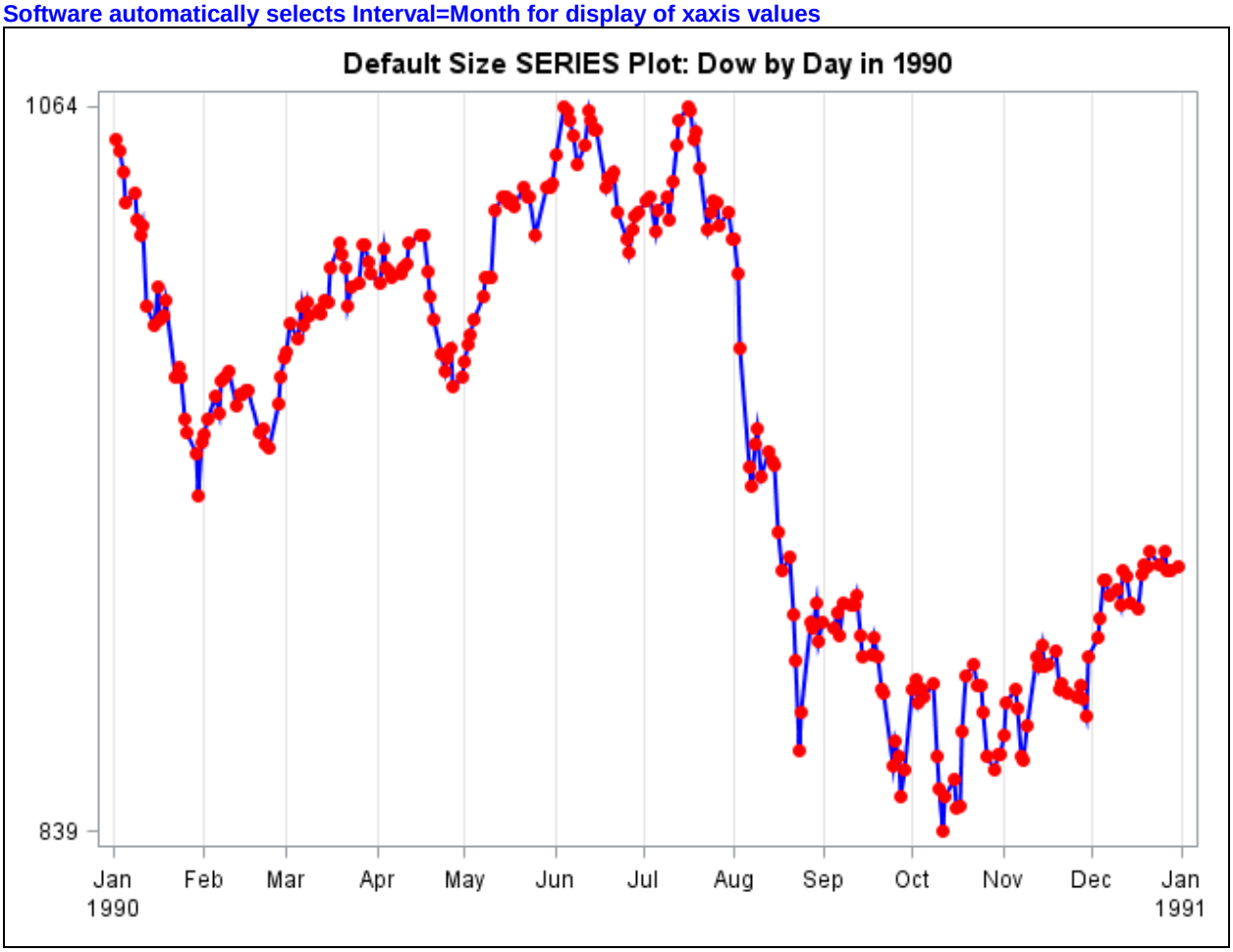
Even for the default width image file, ALL Days Can Be Labeled, But With Alternate Day Values Staggered On Two Lines, Per FitPolicy=Stagger. Therefore, trimming at default width, without Stagger, obviously was NOT needed. Algorithm used made a false inference about the space available.




```
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 Nov2012\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

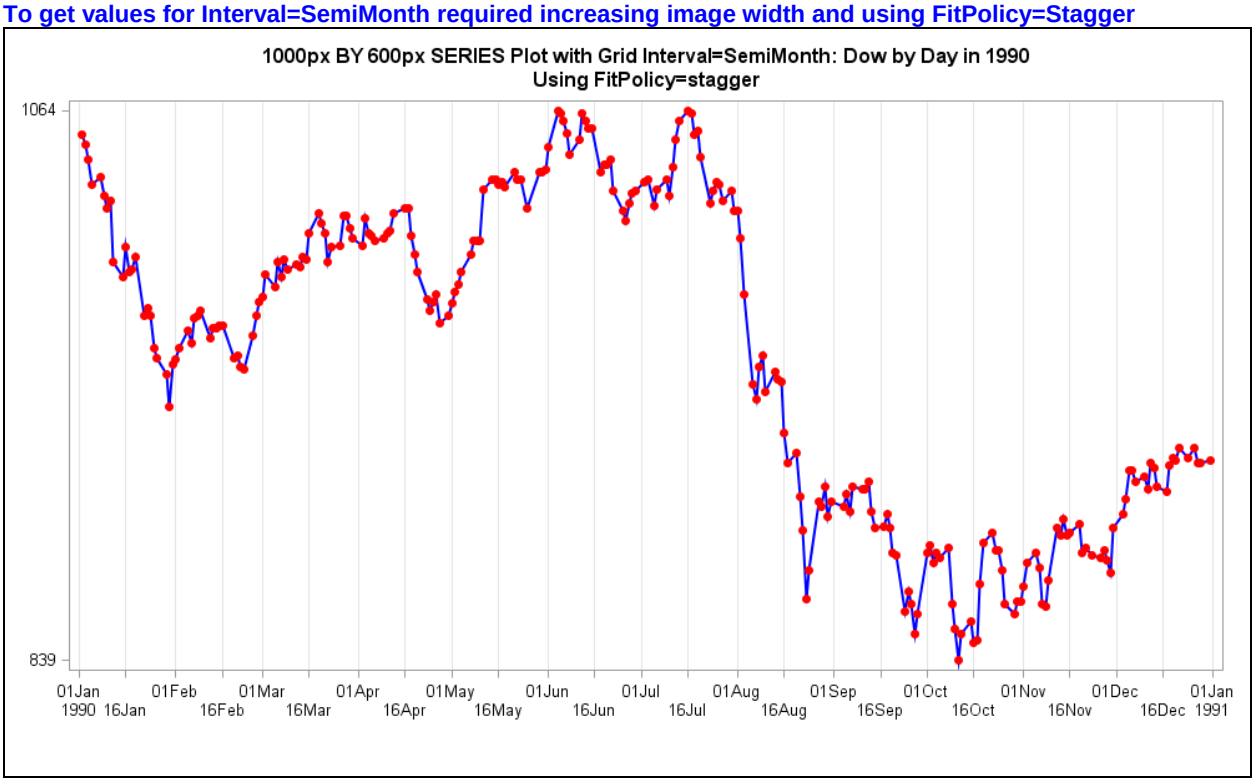
Software automatically selects Interval=Month for display of xaxis values




```
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 Nov2012\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 (**Shrunk To Fit Page Width**) for SGPLOT SERIES Plot By Day (Custom Size 1000px X 600px)

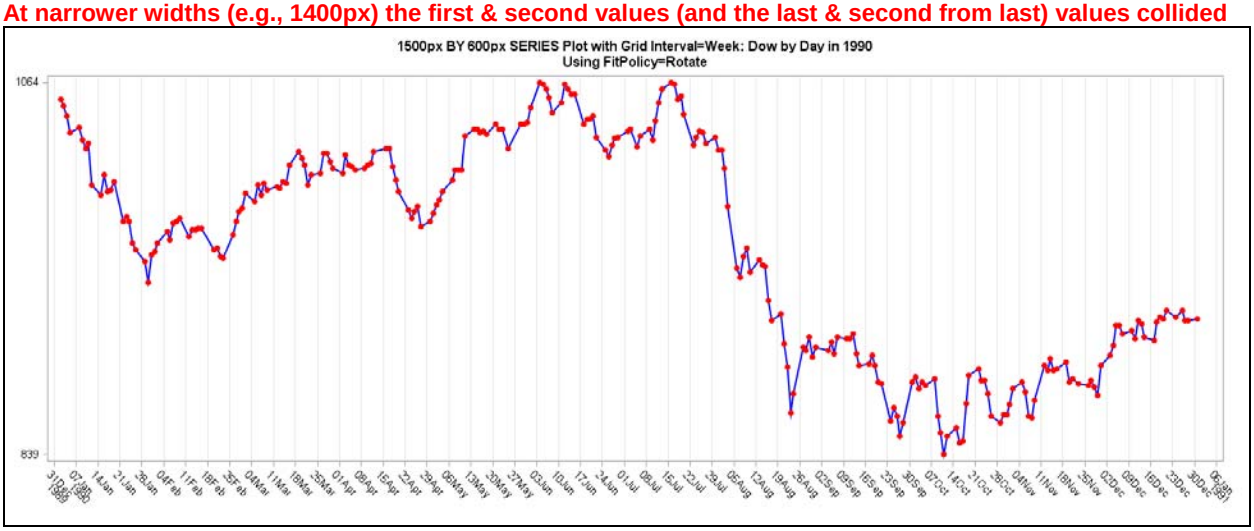
To get values for Interval=SemiMonth required increasing image width and using FitPolicy=Stagger




```
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 Nov2012\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 (**Shrunk To Fit Page Width**) 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

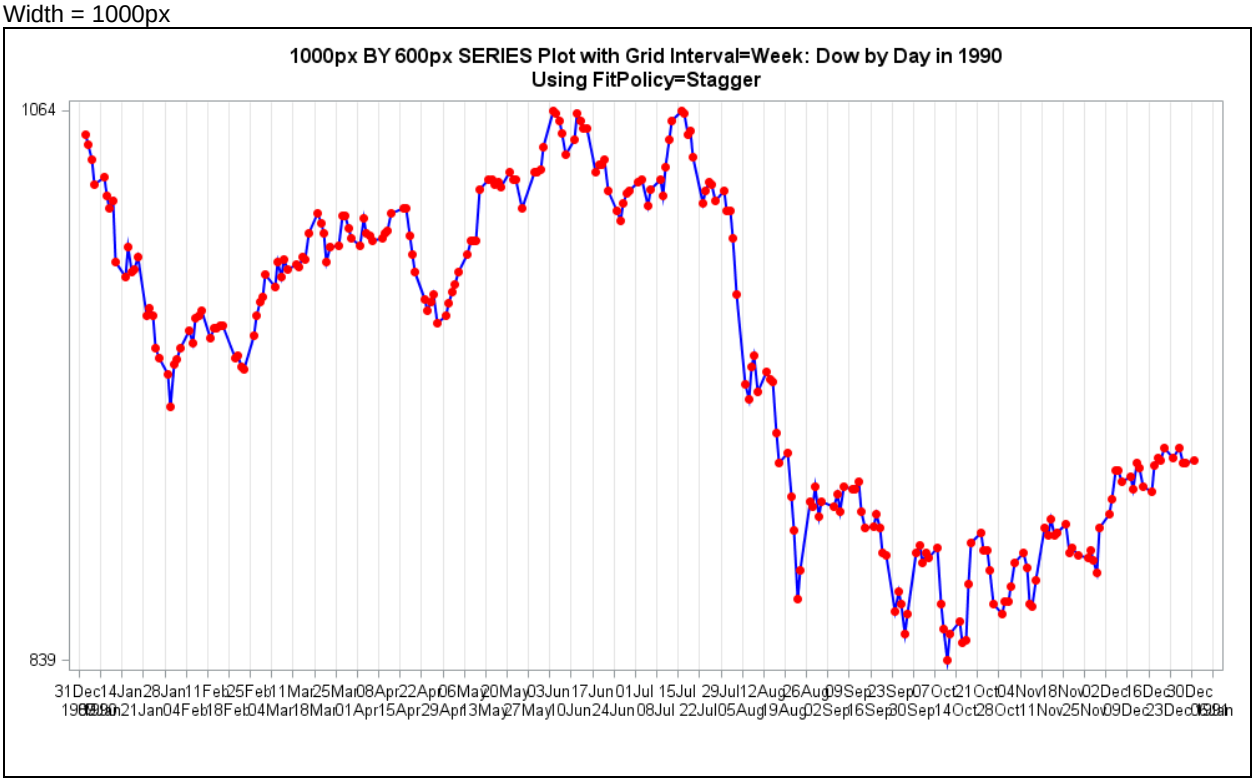



```
ods noresults; ods listing close;
ods graphics on / reset=all border=off antialiasmax=2500 tipmax=2500
    width=1500px height=600px
    imagemap=on imagename=
    'SGPLOT_SERIES_By_Day_1500pxBY600px_IntervalEqualWeek_FitPolicyEqualRotate';
ods html path="D:\@WIILSU Nov2012\Results" (url=none)
 body='SGPLOT_SERIES_By_Day_1500pxBY600px_IntervalEqualWeek_FitPolicyEqualRotate.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;
```

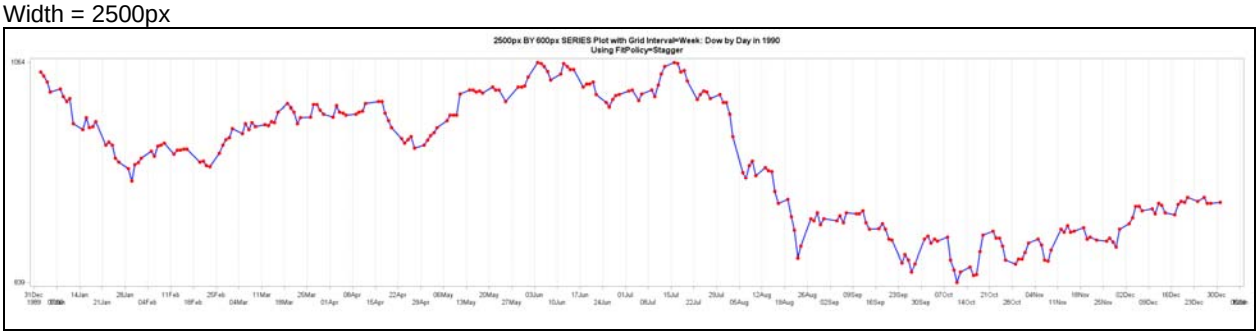
Image Files (**Shrunk To Fit**) for SGPLOT SERIES Plot By Day With Interval=Week and FitPolicy=Stagger

NOTE: 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 first & second values (and last & second from last) values. Failure to recognize Stagger overlap for Interval=Week and to thin the values is a known software problem.

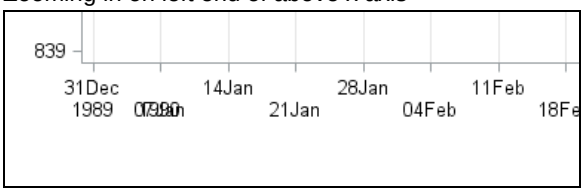
Width = 1000px



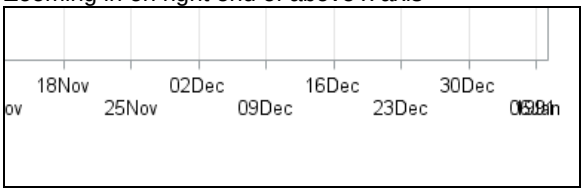
Width = 2500px



Zooming in on left end of above x axis



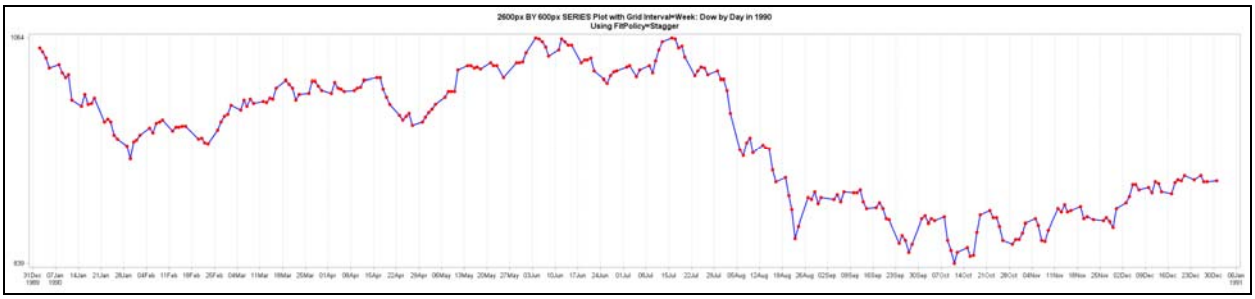
Zooming in on right end of above x axis



Code used to create the above example is the same as that on the following page except for different image width specified in the ODS GRAPHICS statement and mentioned in the title.

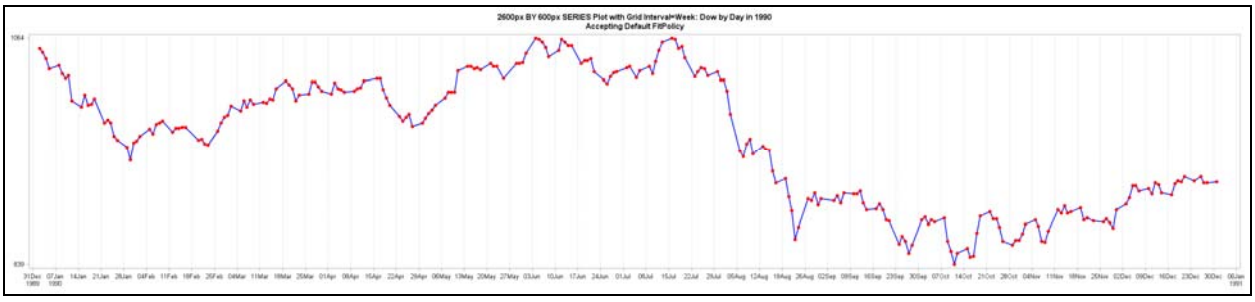
Image File (**Shrunk To Fit Page Width**) for SGPLOT SERIES Plot By Day (Custom Size 2600px X 600px)

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 Nov2012\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;
```

Interval=Week at image width 2600px and accepting default FitPolicy yields same result as above.



This was created with same code as above, except different title2 text, filename changes, and this XAXIS statement:

```
xaxis display=(nolabel) grid interval=week;
```

Simplifying the Appearance of the SERIES Plot

Though necessary tick mark values, and sometimes grid lines, provide communication value, the little tick marks themselves, the 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, but 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.

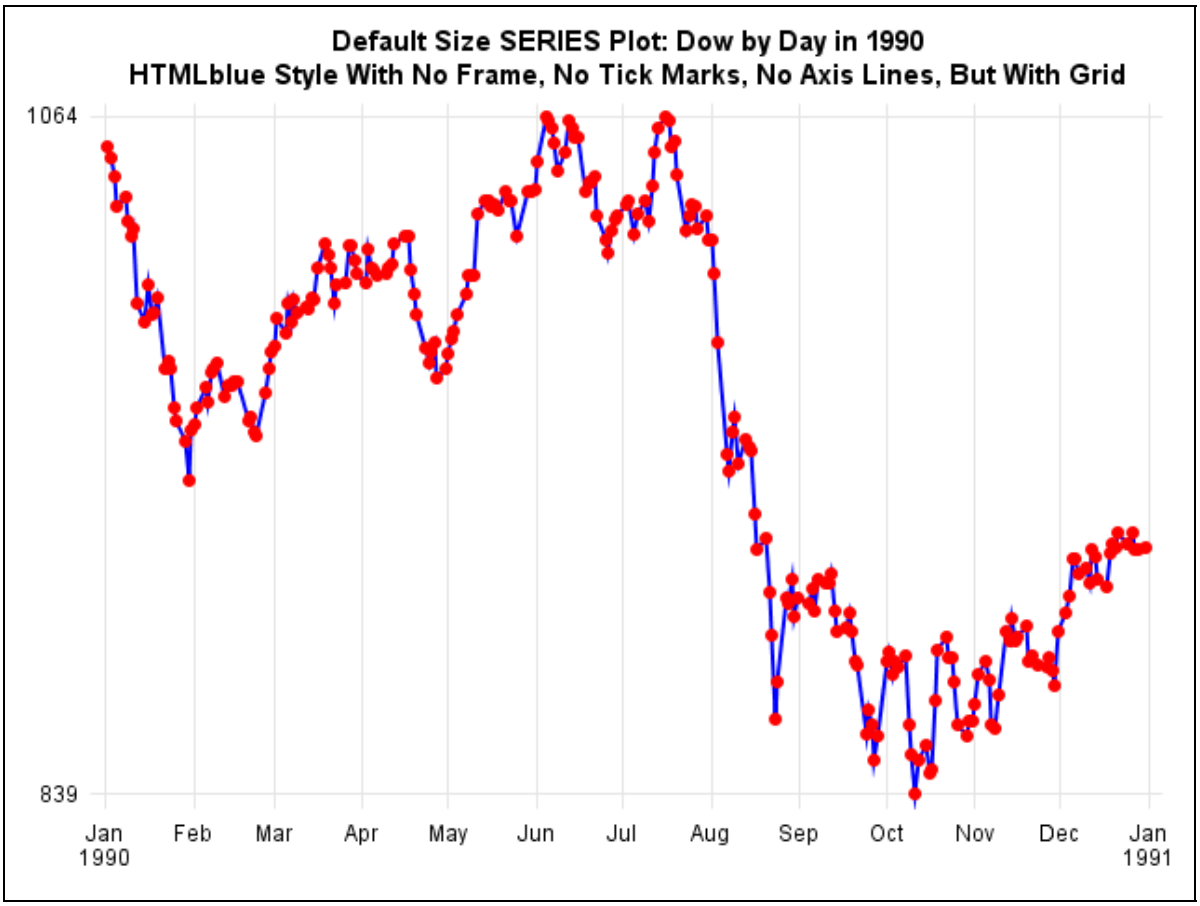
Here is the code used to do the frame removal for the web graph displayed on the following page:

```
proc template;
define style styles.HTMLblueWithNoFrame; /* remove useless box around the plot area */
  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 Nov2012\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;
```

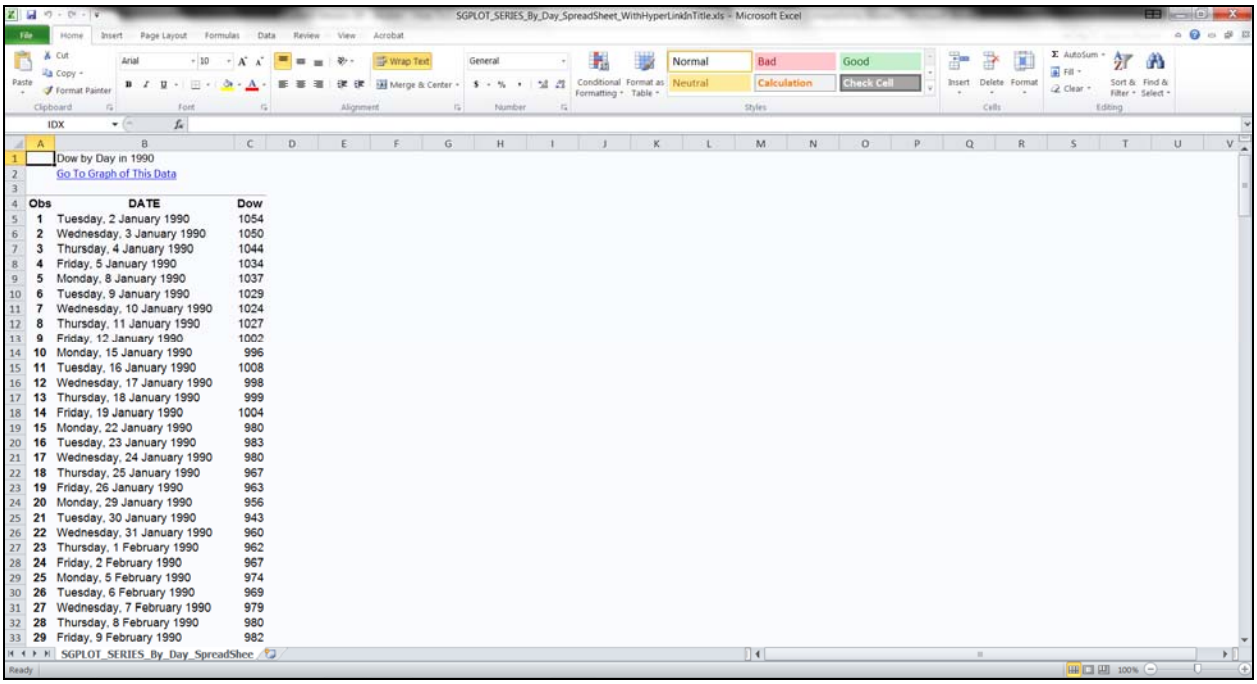
Web Page (**Shrunk To Fit**) for SGPLOT SERIES Plot By Day With Frame Around Plot Display Area Suppressed

Other default features are also suppressed. 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)



The Most Communication-Effective, 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

Below is the code used to create the interlinked plot and spreadsheet. The ODS HTML code block to create the graph accepts the default style (HTMLblue), rather than using the custom HTMLblueWithNoFrame style used in the preceding example. Choice of style has no material effect on the interlinking function being demonstrated here.

```
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';
```

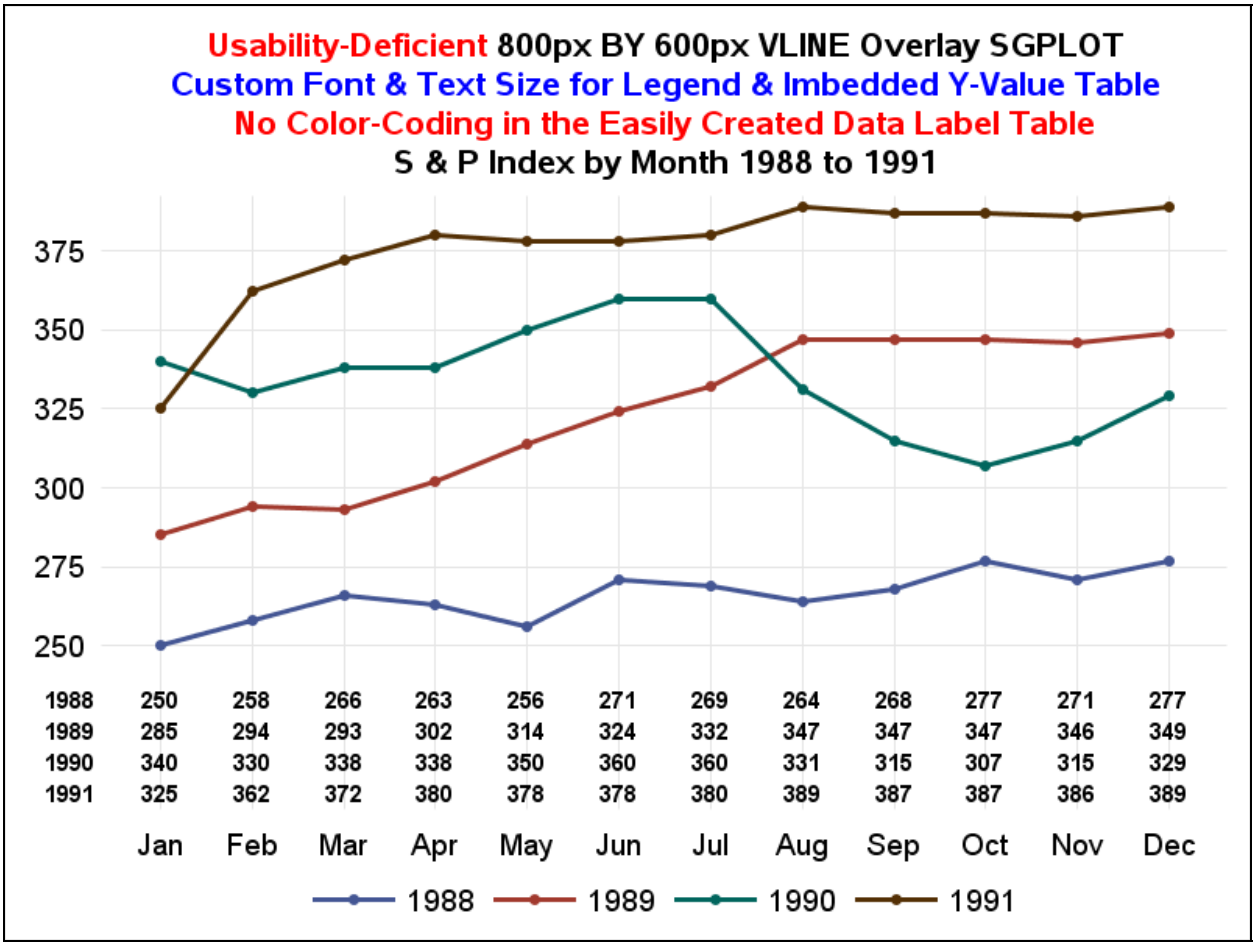
```
ods html path="D:\@WIILSU Nov2012\Results" (url=none) nogtitle
    body='SGPLOT_SERIES_By_Day_800pxBY600px_WithHyperLinkInTitle.html'
    (title='800px BY 600px SERIES Plot With Hyperlink In Title - Dow by Day in 1990');
title
link=
'D:\@WIILSU Nov2012\Results\SGPLOT_SERIES_By_Day_SpreadSheet_WithHyperLinkInTitle.xls'
'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 Nov2012\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;
```

COMMUNICATION-EFFECTIVE MULTI-LINE PLOT WITH DETAIL DATA IMBEDDED: HOW TO PROVIDE IMAGE PLUS DETAIL FOR RTF/Word, PDF, PowerPoint, or PRINT

The y values for a plot are easily imbedded in the plot with the DATALABEL option. You are able to control their text characteristics (font, size, color, etc.) with the DATALABELATTRS option. The labels are automatically annotated adjacent to the plot points, but they can sometimes, as you will see, overlap the plot line. (For SAS/GRAPH, in Reference 8, I presented an algorithm to reliably annotate a plot line without such overlaps.) And when the lines and/or the plot points per line are too numerous and too close, collisions between labels and/or between labels and lines are inevitable. A reliably successful solution removes the y value labels out of the plot area and puts them either in a table separately created with, say, PROC PRINT, and ODS-appended below the graph, or in a table that is part of the graph but below the plot area. **PROC SGPLOT supports time series plots with either the SERIES statement or the VLINE statement, but only the VLINE statement supports the positioning option of DATALABELPOS=BOTTOM, which produces the table that you see in the graph below.**

In earlier work with SAS/GRAPH, I showed that you can use tick mark labels to create, at each x value, a stack of the y values for a multi-line plot. In production, that solution worked well for as many as seven lines and 37 values of x. The solution has no theoretical limit to the number of plot lines or x values, but sufficiently many x values will require the y values to be so small as to be unreadable. That limitation would also apply to the method used below.

NOTE: The **usability deficiency** in the solution below is that the table entries are not color-coded to match the plot lines (unlike my SAS/GRAPH solution). To associate a y value with a plot point means going from the plot point to the legend to translate color into year and then going to that year's entry under that plot point, keeping in mind which is the month of interest. However, if you do not create the table with DATALABELPOS=BOTTOM, the plot-point y value annotations are color-coded to match the line and marker. So, it is obvious that the software could easily be enhanced to likewise do the color-coding here. Complex coding alternatives for better results are provided later.



NOTE: Titles are centered by the software above the plot area, not within the full graph image width.
It is interesting to note that though the starting y axis tick mark value is less than or equal to the lowest y value, the

ending y axis tick mark value is NOT greater than or equal to the highest y value. This does make better use of the available image height, and does not make the graph unusable.

NOTE: The title for the legend is absent because I omitted use of TITLE= in the KEYLEGEND statement below. I used the KEYLEGEND statement only to remove the legend frame and to control the font and size of the legend text. When TITLE= is omitted, the legend title is not defaulted to the variable name or variable label for the GROUP variable. The SAS OnlineDoc describes the TITLE parameter on the KEYLEGEND statement as optional. It is optional, in the sense that omitting it causes no ERROR message, but its omission has the unexpected result of no title text at all. If you omit using KEYLEGEND, then the legend title does default to "Year", the font and text size for title and legend values are defaults, and the legend is displayed in a box.

Code in this section outputs the graphs to disk. From disk, graphs can be manually inserted in Word, PowerPoint, or Excel files. For output direct to RTF or PDF, needed ODS wrapping statements are standard and well explained in other published resources, including those at support.sas.com or in SAS OnlineDoc.

Here is the code used to create the graph above:

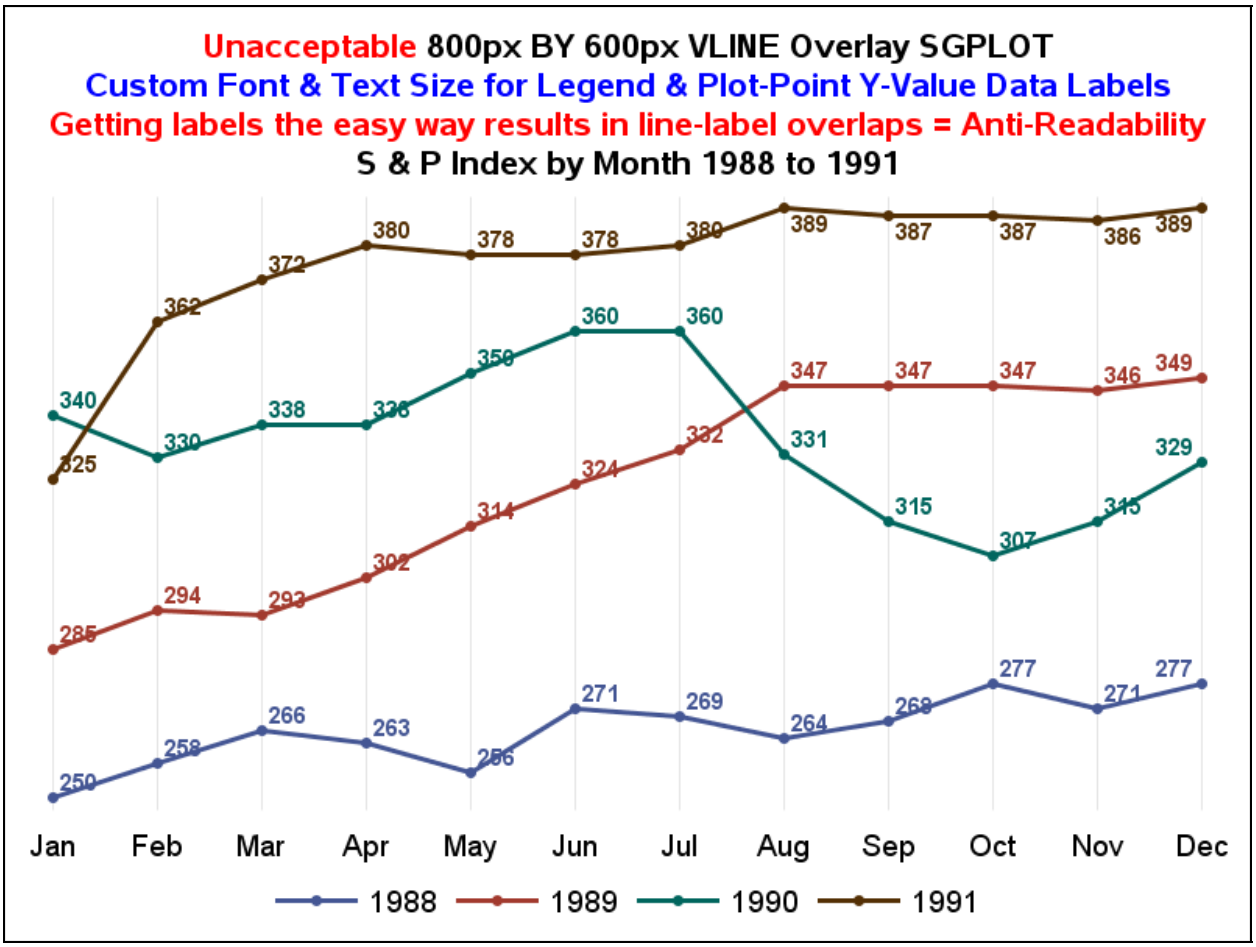
```
ods listing gpath="D:\@WIILSU Nov2012\Results" style=Styles.htmlblueWithNoFrame;
ods graphics on / reset=all border=on labelmax=48
  width=800px height=600px /* override default size 480px X 360px */
  imagename=
'SGPLOT_VLINE_By_Month_OneResponseVar_YearAsGroupVarForOverlay_WithYvalueDataLabelsInT
able_AutomaticColorsForLinesAndMarkersAndLabels';
title1 FONT='Albany AMT/Bold' HEIGHT=14 PT
  COLOR=Red 'Usability-Deficient '
  COLOR=Black '800px BY 600px VLINE Overlay SGPLOT';
title2 FONT='Albany AMT/Bold' HEIGHT=14 PT COLOR=Blue
  'Custom Font & Text Size for Legend & Imbedded Y-Value Table';
title3 FONT='Albany AMT/Bold' HEIGHT=14 PT COLOR=Red
  'No Color-Coding in the Easily Created Data Label Table';
title4 FONT='Albany AMT/Bold' HEIGHT=14 PT
  'S & P Index by Month 1988 to 1991';
proc sgplot data=work.SandPindexByMon1988to1991;
vline Month / response=SandPindex group=Year
      markers markerattrs=(size=7 symbol=circlefilled)
     lineattrs=(thickness=3 pattern=solid)
     datalabel=SandPindex
     datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold)
     datalabelpos=bottom;
keylegend / noborder
            titleattrs=(family='Albany AMT/Bold' size=12 PT)
            valueattrs=(family='Albany AMT/Bold' size=12 PT);
yaxis display=(nolabel noticks noline) grid
      offsetmin=0.05 /* space between top of datalabel table & bottom of plot area */
      valueattrs=(family='Albany AMT/Bold' size=12 PT);
xaxis display=(nolabel noticks noline) grid
      valueattrs=(family='Albany AMT/Bold' size=12 PT) values=(1 to 12 by 1);
format Month MonthAbbrev.;
format SandPindex 5.;
run;
ods listing close; ods listing;
```

NOTE: This particular data requires that the code in Appendix 2 be run first to prepare the input data.

Below is the result of omitting DATALABELPOS=BOTTOM, changing the titles, using HEIGHT=13pt for TITLE3, and changing the YAXIS statement to

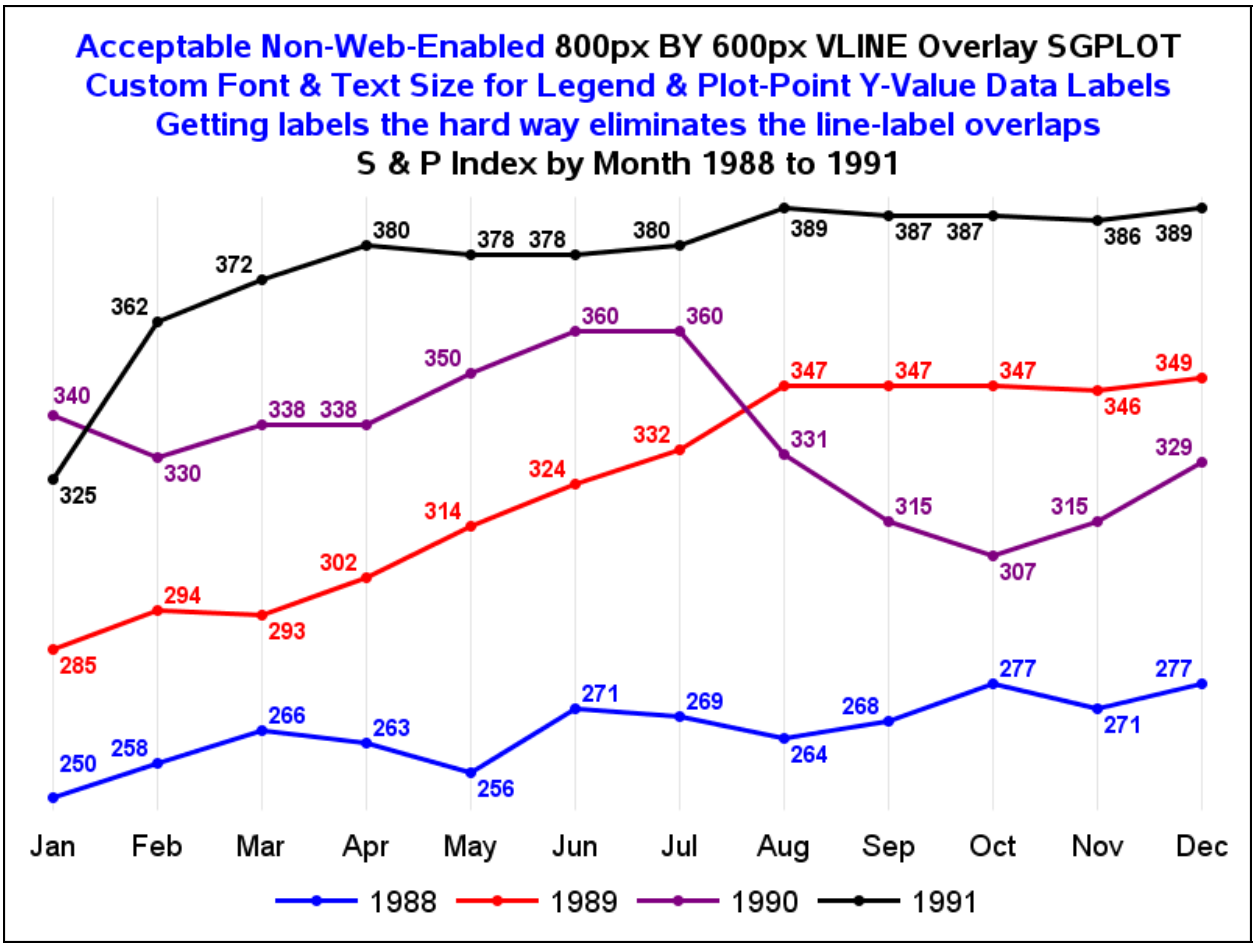
```
yaxis display=none;
```

but otherwise using the same code as for the data label table example above. Since all of the y values are annotated in the plot area itself, there is no value in providing a y axis and tick mark values. **With the y axis omitted, unlike the prior example, the titles are centered within the full graph width.**



When experimenting with the TRANSPARENCY option that is available on the VLINE statement, which documentation says will lighten the lines and markers, I found that the data label text was also lightened to the same degree, thus preserving impaired readability.

Good News: If you do some extra work with preprocessing the data and do some extra coding in the SGPLOT PROC step, it IS possible to remove the line-label overlaps, as is shown on the next page. **What is surprising about the solution is that it also uses PROC SGPLOT and the VLINE statement, but the VLINE DATALABEL option delivers line-label collisions if you create the four lines using one VLINE statement with the GROUP=Year option, but no collisions if the lines are created with four separate, more complicated, VLINE statements with no GROUP option.**



NOTE: It should be realized that if there are enough lines intersecting often enough, or nearly parallel and close enough to each other, and/or sufficiently many data points within the lines, collisions between labels and/or labels and lines are inevitable. The above solution is Acceptable, for this particular set of data, but not Most Reliable for the general case. An imbedded data label table is more likely to be successful, but the text size needs to be reduced sufficiently as the number of y values per line increases.

Here is the code used to create the graph above:

```
data work.SandPindexByMonFourYearVars(drop=Year SandPindex);
set work.SandPindexByMon1988to1991;
if Year EQ 1988
then SandPindex_1988 = SandPindex;
else
if Year EQ 1989
then SandPindex_1989 = SandPindex;
else
if Year EQ 1990
then SandPindex_1990 = SandPindex;
else SandPindex_1991 = SandPindex;
run;

ods listing gpath="D:\@WIILSU Nov2012\Results" style=Styles.htmlblueWithNoFrame;
ods graphics on / reset=all border=on labelmax=48
  width=800px height=600px /* override default size 480px X 360px */
  imagename=
'SGPLOT_VLINE_By_Month_FourResponseVariableOverlay_WithYvaluePlotPointDataLabels_Custo
mColorsForLinesAndMarkersAndLabels';
```

```
title1 FONT='Albany AMT/Bold' HEIGHT=14 PT
  COLOR=Blue 'Acceptable Non-Web-Enabled '
  COLOR=Black '800px BY 600px VLINE Overlay SGPLOT';
title2 FONT='Albany AMT/Bold' HEIGHT=14 PT COLOR=Blue
  'Custom Font & Text Size for Legend & Plot-Point Y-Value Data Labels';
title3 FONT='Albany AMT/Bold' HEIGHT=14 PT COLOR=Blue
  'Getting labels the hard way eliminates the line-label overlaps';
title4 FONT='Albany AMT/Bold' HEIGHT=14 PT
  'S & P Index by Month 1988 to 1991';
proc sgplot data=work.SandPindexByMonFourYearVars;
vline Month / response=SandPindex_1988 nostatlabel legendlabel='1988'
      markers markerattrs=(size=7 symbol=circlefilled color=blue)
      lineattrs=(thickness=3 pattern=solid color=blue)
      datalabel=SandPindex_1988
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=blue);
vline Month / response=SandPindex_1989 nostatlabel legendlabel='1989'
      markers markerattrs=(size=7 symbol=circlefilled color=red)
      lineattrs=(thickness=3 pattern=solid color=red)
      datalabel=SandPindex_1989
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=red);
vline Month / response=SandPindex_1990 nostatlabel legendlabel='1990'
      markers markerattrs=(size=7 symbol=circlefilled color=purple)
      lineattrs=(thickness=3 pattern=solid color=purple)
      datalabel=SandPindex_1990
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=purple);
vline Month / response=SandPindex_1991 nostatlabel legendlabel='1991'
      markers markerattrs=(size=7 symbol=circlefilled color=black)
      lineattrs=(thickness=3 pattern=solid color=black)
      datalabel=SandPindex_1991
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=black);
keylegend / noborder
            titleattrs=(family='Albany AMT/Bold' size=12 PT)
            valueattrs=(family='Albany AMT/Bold' size=12 PT);
yaxis display=none;
xaxis display=(nolabel noticks noline) grid
      valueattrs=(family='Albany AMT/Bold' size=12 PT) values=(1 to 12 by 1);
format Month MonthAbbrev.;
format SandPindex_1988 SandPindex_1989 SandPindex_1990 SandPindex_1991 5.;
run;
ods listing close; ods listing;
```

The above code is more complex. In principle, one could convert it to a macro. It would be straightforward, but a lot of code. I leave it as an exercise for the interested and motivated reader, and would be grateful for a copy of the code.

As a next step, let's see what happens if one simply adds "datalabelpos=bottom" to each of the VLINE statements above, changes the graph titles and filename, provides LABEL statements for each response variable:

```
label SandPindex_1988='1988';
label SandPindex_1989='1989';
label SandPindex_1990='1990';
label SandPindex_1991='1991';
```

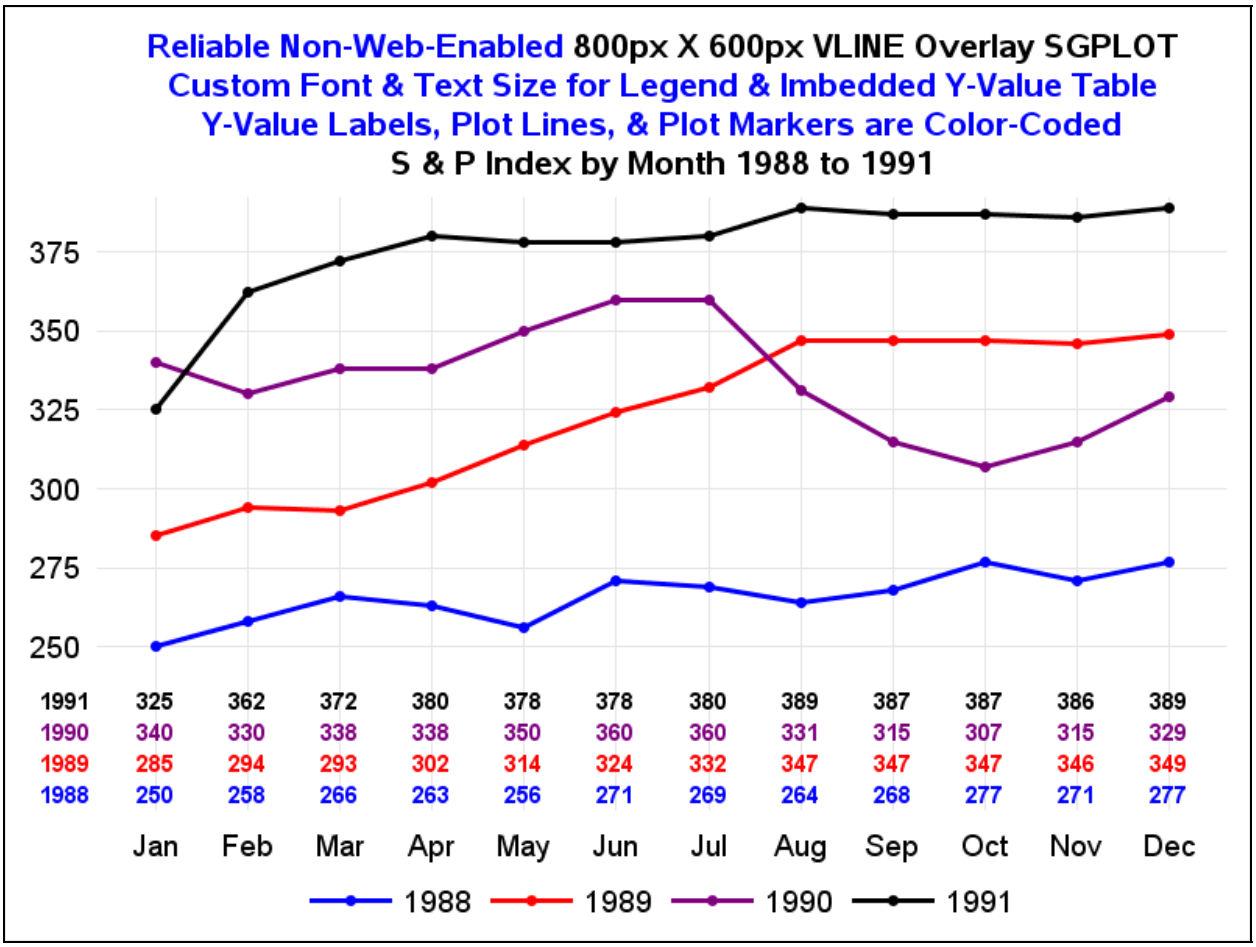
and restores the y axis tick mark values by replacing

```
yaxis display=none;
```

with this statement

```
yaxis display=(nolabel noticks noline) grid offsetmin=0.05
      valueattrs=(family='Albany AMT/Bold' size=12 PT);
```

The result is on the next page.



Here is the code used to create the graph above:

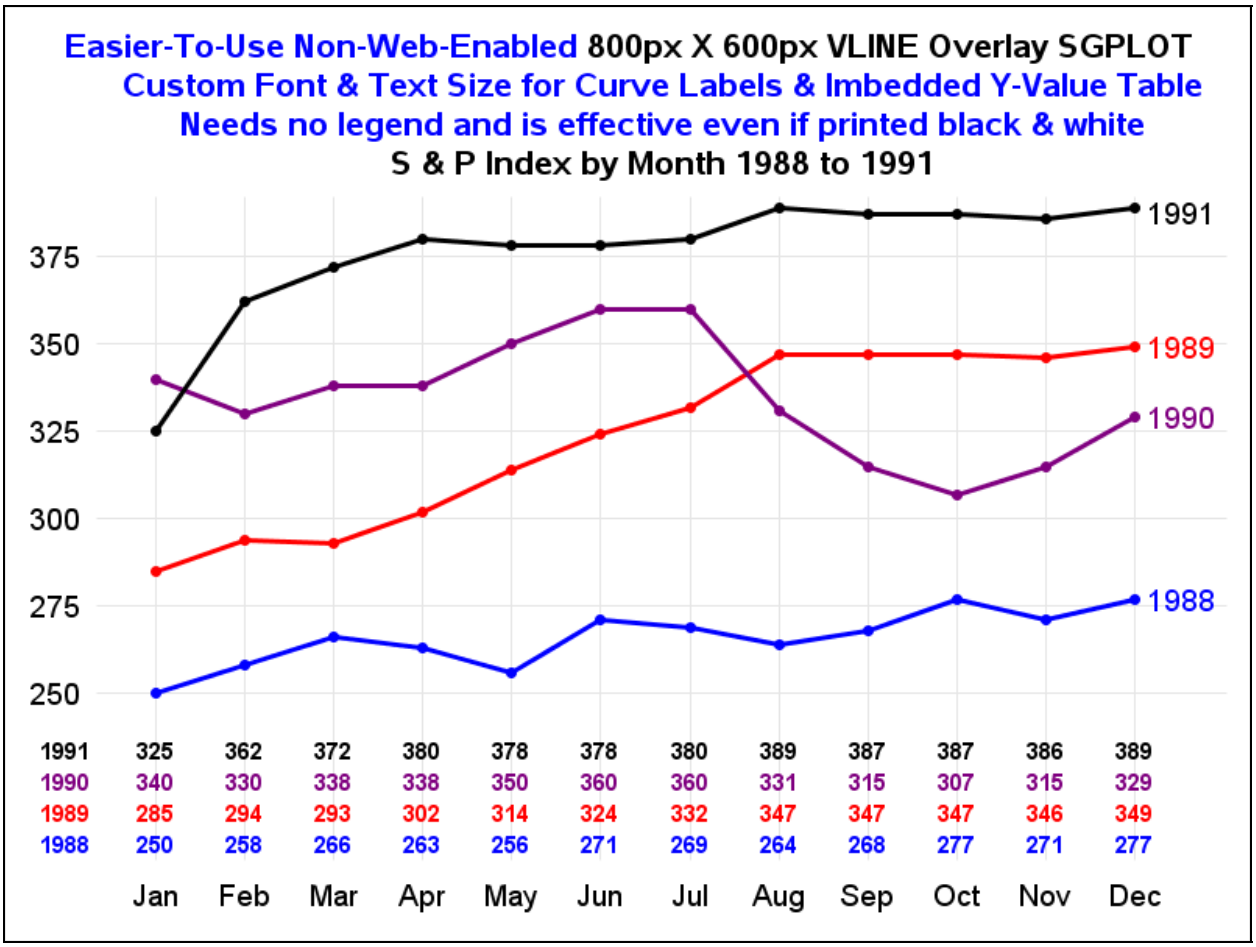
```
data work.SandPindexByMonFourYearVars(drop=Year SandPindex);
set work.SandPindexByMon1988to1991;
if Year EQ 1988
then SandPindex_1988 = SandPindex;
else
if Year EQ 1989
then SandPindex_1989 = SandPindex;
else
if Year EQ 1990
then SandPindex_1990 = SandPindex;
else SandPindex_1991 = SandPindex;
run;

ods listing gpath="D:\@WIILSU Nov2012\Results" style=Styles.htmlblueWithNoFrame;
ods graphics on / reset=all border=on labelmax=48
  width=800px height=600px /* override default size 480px X 360px */
  imagename=
'SGPLOT_VLINE_By_Month_FourResponseVariableOverlay_WithYvalueDataLabelsInTable_CustomC
olorsForLegendAndLinesAndMarkersAndDataLabels';
title1 FONT='Albany AMT/Bold' HEIGHT=14 PT
  COLOR=Blue 'Reliable Non-Web-Enabled '
  COLOR=Black '800px X 600px VLINE Overlay SGPLOT';
title2 FONT='Albany AMT/Bold' HEIGHT=14 PT COLOR=Blue
  'Custom Font & Text Size for Legend & Imbedded Y-Value Table';
title3 FONT='Albany AMT/Bold' HEIGHT=14 PT COLOR=Blue
  'Y-Value Labels, Plot Lines, & Plot Markers are Color-Coded';
```

```
title4 FONT='Albany AMT/Bold' HEIGHT=14 PT
  'S & P Index by Month 1988 to 1991';
proc sgplot data=work.SandPindexByMonFourYearVars;
vline Month / response=SandPindex_1988 nostatlabel
      markers markerattrs=(size=7 symbol=circlefilled color=blue)
      lineattrs=(thickness=3 pattern=solid color=blue)
      datalabel=SandPindex_1988
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=blue)
      datalabelpos=bottom;
vline Month / response=SandPindex_1989 nostatlabel
      markers markerattrs=(size=7 symbol=circlefilled color=red)
      lineattrs=(thickness=3 pattern=solid color=red)
      datalabel=SandPindex_1989
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=red)
      datalabelpos=bottom;
vline Month / response=SandPindex_1990 nostatlabel
      markers markerattrs=(size=7 symbol=circlefilled color=purple)
      lineattrs=(thickness=3 pattern=solid color=purple)
      datalabel=SandPindex_1990
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=purple)
      datalabelpos=bottom;
vline Month / response=SandPindex_1991 nostatlabel
      markers markerattrs=(size=7 symbol=circlefilled color=black)
      lineattrs=(thickness=3 pattern=solid color=black)
      datalabel=SandPindex_1991
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=black)
      datalabelpos=bottom;
keylegend / noborder
            titleattrs=(family='Albany AMT/Bold' size=12 PT)
            valueattrs=(family='Albany AMT/Bold' size=12 PT);
yaxis display=(nolabel noticks noline) grid
      offsetmin=0.05 /* space between top of datalabel table & bottom of plot area */
      valueattrs=(family='Albany AMT/Bold' size=12 PT);
xaxis display=(nolabel noticks noline) grid
      valueattrs=(family='Albany AMT/Bold' size=12 PT) values=(1 to 12 by 1);
format Month MonthAbbrev.;
format SandPindex_1988 SandPindex_1989 SandPindex_1990 SandPindex_1991 5.;
label SandPindex_1988='1988';
label SandPindex_1989='1989';
label SandPindex_1990='1990';
label SandPindex_1991='1991';
run;
ods listing close; ods listing;
```

As was the case for the improved solution for y-value plot-point data labels, the above code is complex. My quest for a way to create a multi-line plot with an imbedded color-coded y-value table was inspired by seeing Philip R. Holland's black-and-white multi-line plot with imbedded table, which relies on use of Graph Template Language (GTL) and PROC SGRENDER. See Reference 9.

I recently heard Sanjay Matange emphasize the benefits of using the curvelabel option to replace the legend. The curvelabels eliminate the need for eye travel from a legend to the plot lines to identify the plots. Of course, if the legend entry text is long, then curvelabels would steal display horizontal space from the plot lines. With a history plot, it is the horizontal space for the time dimension that is usually the critical constraint, unless there are only a small number of dates, times, or datatimes.

On a multi-line plot, putting a label anywhere inside the plot area would incur the risk of overlaps between labels for different curves or between a curve label and one of the other curves. I prefer the option curvelabelpos=max, which, per the documentation, "places the label at the part of the curve closest to the maximum X axis value". I initially tried curvelabelpos=end, but the label of each curve was too close to its last plot point. If you have a plot where too many of the last plot points are too close together, overlap of the curve labels might be inevitable, in which case curvelabelpos=min or use of a legend might be the required alternative. Also, multi-line plots with curvelabels can be interpreted even if printed (or rendered) in black-and-white without using different symbol types by line and a legend.

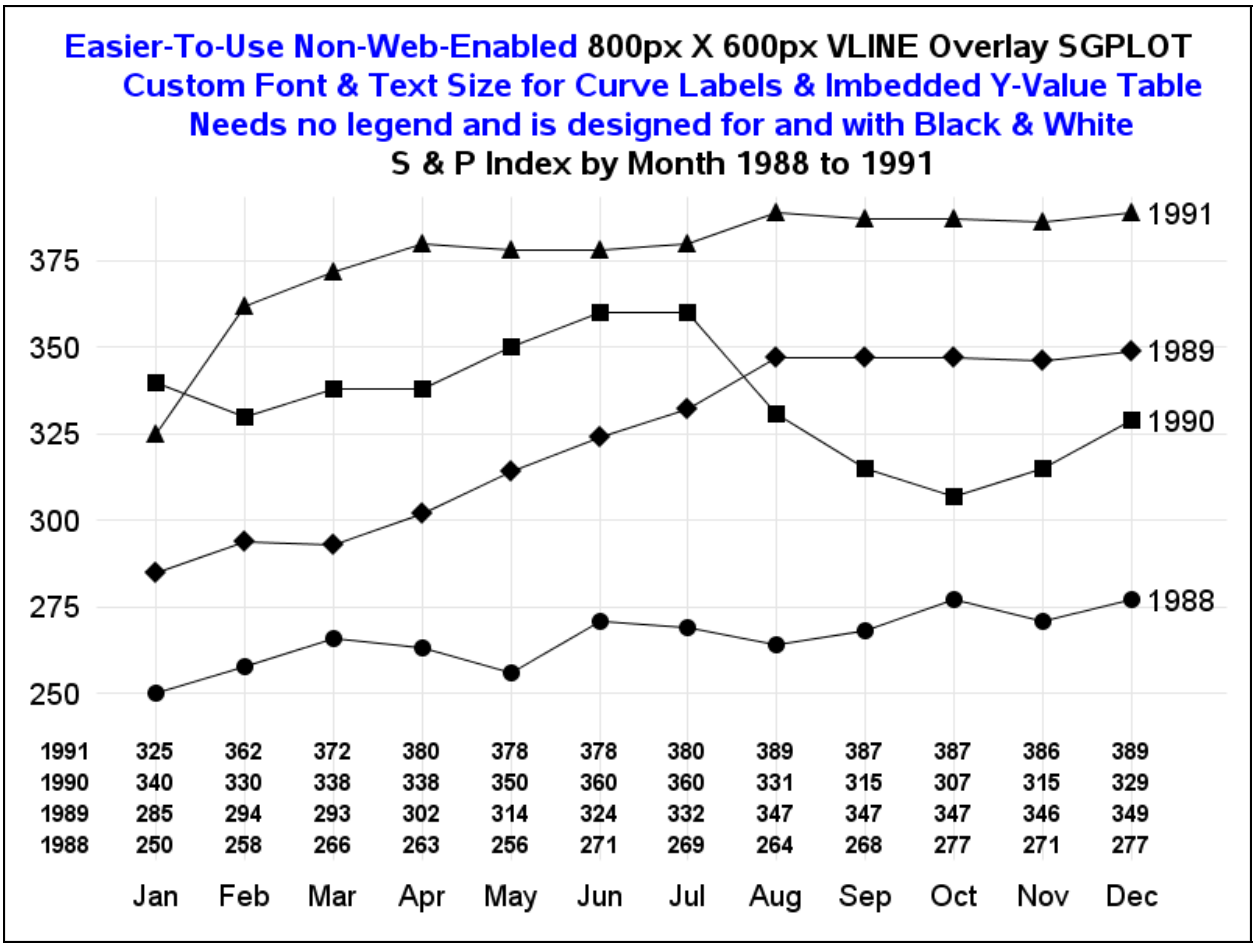


Here is the final code used to create the graph above:

```
data work.SandPindexByMonFourYearVars(drop=Year SandPindex);
set work.SandPindexByMon1988to1991;
if Year EQ 1988
then SandPindex_1988 = SandPindex;
else
if Year EQ 1989
then SandPindex_1989 = SandPindex;
else
if Year EQ 1990
then SandPindex_1990 = SandPindex;
else SandPindex_1991 = SandPindex;
run;

ods listing gpath="D:\@WIILSU Nov2012\Results" style=Styles.HTMLblueWithNoFrame;
ods graphics on / reset=all border=on labelmax=48
  width=800px height=600px /* override default size 480px X 360px */
  imagename=
'SGPLOT_VLINE_By_Month_FourResponseVariableOverlay_WithYvalueDataLabelsInTableAndCurve
LabelAtEnd_CustomColorsForLinesAndMarkersAndDataLabelsAndCurveLabels';
title1 FONT='Albany AMT/Bold' HEIGHT=14 PT
  COLOR=Blue 'Easier-To-Use Non-Web-Enabled '
  COLOR=Black '800px X 600px VLINE Overlay SGPLOT';
title2 FONT='Albany AMT/Bold' HEIGHT=14 PT COLOR=Blue
  'Custom Font & Text Size for Curve Labels & Imbedded Y-Value Table';
title3 FONT='Albany AMT/Bold' HEIGHT=14 PT COLOR=Blue
  'Needs no legend and is effective even if printed black & white';
```

```
title4 FONT='Albany AMT/Bold' HEIGHT=14 PT
  'S & P Index by Month 1988 to 1991';
proc sgplot data=work.SandPindexByMonFourYearVars;
vline Month / response=SandPindex_1988 nostatlabel
      markers markerattrs=(size=7 symbol=circlefilled color=blue)
      lineattrs=(thickness=3 pattern=solid color=blue)
      datalabel=SandPindex_1988
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=blue)
      datalabelpos=bottom
         curvelabel='1988'
      curvelabelattrs=(family='Albany AMT/Bold' size=12 PT color=blue)
      curvelabelpos=max;
vline Month / response=SandPindex_1989 nostatlabel
      markers markerattrs=(size=7 symbol=circlefilled color=red)
      lineattrs=(thickness=3 pattern=solid color=red)
      datalabel=SandPindex_1989
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=red)
      datalabelpos=bottom
         curvelabel='1989'
      curvelabelattrs=(family='Albany AMT/Bold' size=12 PT color=red)
      curvelabelpos=max;
vline Month / response=SandPindex_1990 nostatlabel
      markers markerattrs=(size=7 symbol=circlefilled color=purple)
      lineattrs=(thickness=3 pattern=solid color=purple)
      datalabel=SandPindex_1990
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=purple)
      datalabelpos=bottom
         curvelabel='1990'
      curvelabelattrs=(family='Albany AMT/Bold' size=12 PT color=purple)
      curvelabelpos=max;
vline Month / response=SandPindex_1991 nostatlabel
      markers markerattrs=(size=7 symbol=circlefilled color=black)
      lineattrs=(thickness=3 pattern=solid color=black)
      datalabel=SandPindex_1991
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=black)
      datalabelpos=bottom
         curvelabel='1991'
      curvelabelattrs=(family='Albany AMT/Bold' size=12 PT color=black)
      curvelabelpos=max;
yaxis display=(nolabel noticks noline) grid
      offsetmin=0.05 /* space between top of datalabel table & bottom of plot area */
      valueattrs=(family='Albany AMT/Bold' size=12 PT);
xaxis display=(nolabel noticks noline) grid
      valueattrs=(family='Albany AMT/Bold' size=12 PT) values=(1 to 12 by 1);
format Month MonthAbbrev.;
format SandPindex_1988 SandPindex_1989 SandPindex_1990 SandPindex_1991 5.;
label SandPindex_1988='1988';
label SandPindex_1989='1989';
label SandPindex_1990='1990';
label SandPindex_1991='1991';
run;
ods listing close; ods listing;
```

To create a usable multi-line plot in black and white requires use of different plot symbols for each plot line. On the following page is an illustration of such a plot, with curvelabels. (For a black and white plot WITHOUT curvelabels, given the illustration and creation code above for a four-color graph without curvelabels and with a legend, you should be able to visualize and produce one yourself, by modifying the code below the illustration on the next page.)



Here is the code used to create the graph above:

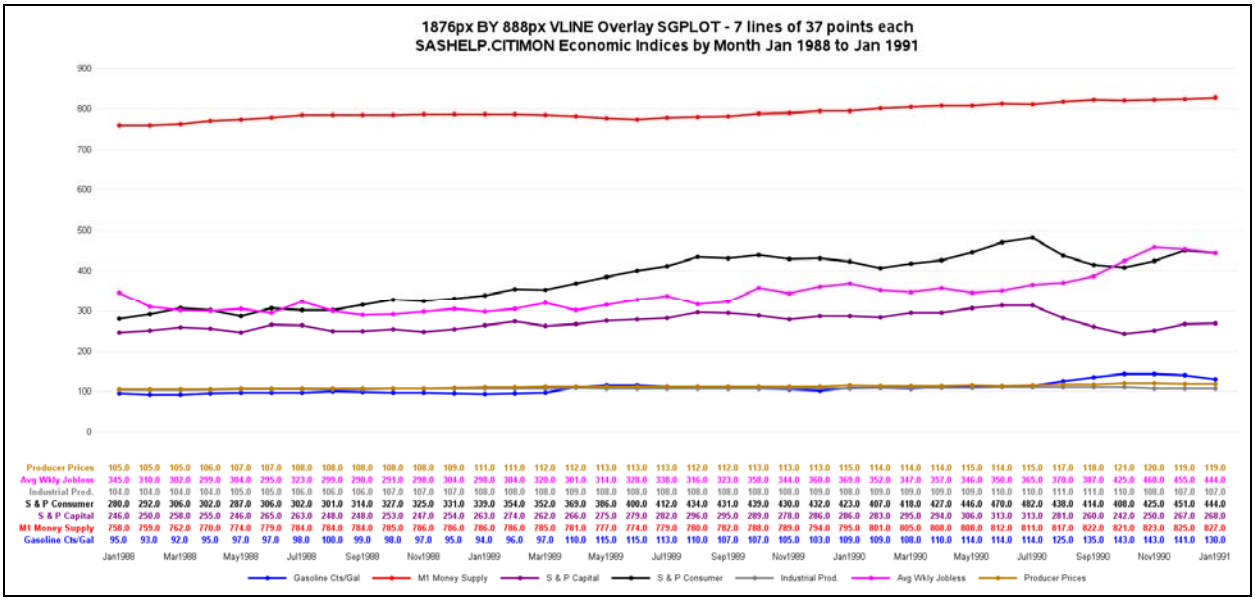
```
data work.SandPindexByMonFourYearVars(drop=Year SandPindex);
set work.SandPindexByMon1988to1991;
if Year EQ 1988
then SandPindex_1988 = SandPindex;
else
if Year EQ 1989
then SandPindex_1989 = SandPindex;
else
if Year EQ 1990
then SandPindex_1990 = SandPindex;
else SandPindex_1991 = SandPindex;
run;

ods listing gpath="D:\@WIILSU Nov2012\Results" style=Styles.HTMLblueWithNoFrame;
ods graphics on / reset=all border=on labelmax=48
  width=800px height=600px /* override default size 480px X 360px */
  imagename=
'SGPLOT_VLINE_By_Month_FourResponseVariableOverlay_WithYvalueDataLabelsInTableAndCurve
LabelAtEnd';
title1 FONT='Albany AMT/Bold' HEIGHT=14 PT
  COLOR=Blue 'Easier-To-Use Non-Web-Enabled '
  COLOR=Black '800px X 600px VLINE Overlay SGPLOT';
title2 FONT='Albany AMT/Bold' HEIGHT=14 PT COLOR=Blue
  'Custom Font & Text Size for Curve Labels & Imbedded Y-Value Table';
title3 FONT='Albany AMT/Bold' HEIGHT=14 PT COLOR=Blue
  'Needs no legend and is designed for and with Black & White';
```

```
title4 FONT='Albany AMT/Bold' HEIGHT=14 PT
  'S & P Index by Month 1988 to 1991';
proc sgplot data=work.SandPindexByMonFourYearVars;
vline Month / response=SandPindex_1988 nostatlabel
      markers markerattrs=(size=10 symbol=circlefilled color=black)
      lineattrs=(color=black)
      datalabel=SandPindex_1988
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=black)
      datalabelpos=bottom
         curvelabel='1988'
      curvelabelattrs=(family='Albany AMT/Bold' size=12 PT color=black)
      curvelabelpos=max;
vline Month / response=SandPindex_1989 nostatlabel
      markers markerattrs=(size=10 symbol=diamondfilled color=black)
      lineattrs=(color=black)
      datalabel=SandPindex_1989
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=black)
      datalabelpos=bottom
         curvelabel='1989'
      curvelabelattrs=(family='Albany AMT/Bold' size=12 PT color=black)
      curvelabelpos=max;
vline Month / response=SandPindex_1990 nostatlabel
      markers markerattrs=(size=10 symbol=squarefilled color=black)
      lineattrs=(color=black)
      datalabel=SandPindex_1990
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=black)
      datalabelpos=bottom
         curvelabel='1990'
      curvelabelattrs=(family='Albany AMT/Bold' size=12 PT color=black)
      curvelabelpos=max;
vline Month / response=SandPindex_1991 nostatlabel
      markers markerattrs=(size=10 symbol=trianglefilled color=black)
      lineattrs=(color=black)
      datalabel=SandPindex_1991
      datalabelattrs=(family='Albany AMT/Bold' size=10 PT weight=Bold color=black)
      datalabelpos=bottom
         curvelabel='1991'
      curvelabelattrs=(family='Albany AMT/Bold' size=12 PT color=black)
      curvelabelpos=max;
yaxis display=(nolabel noticks noline) grid
      offsetmin=0.05 /* space between top of datalabel table & bottom of plot area */
      valueattrs=(family='Albany AMT/Bold' size=12 PT);
xaxis display=(nolabel noticks noline) grid
      valueattrs=(family='Albany AMT/Bold' size=12 PT) values=(1 to 12 by 1);
format Month MonthAbbrev.;
format SandPindex_1988 SandPindex_1989 SandPindex_1990 SandPindex_1991 5.;
label SandPindex_1988='1988';
label SandPindex_1989='1989';
label SandPindex_1990='1990';
label SandPindex_1991='1991';
run;
ods listing close; ods listing;
```

The y-value label table concept continues to work well for more lines and more data points per line. As the horizontal density of the table rows increases, the software automatically reduces the font size without any message in the SAS log. The font family and text color specified in the datalabelattrs parameter are respected during that reduction and the text size is maintained for the the row labels at the left margin. If you wish to prevent rotation of the x-axis tick mark labels, fitpolicy=thin will accomplish that. It is useful and wise to reduce the size specification in the valueattrs parameter of the x-axis statement since the software does not do automatic reduction of the tick mark labels.

Below is an illustration of a seven-line overlay plot for 37 months of data, i.e., a three-year month-to-same-month trend. In this case, the font size in the data label table incurred no reduction. This graph was designed to use the maximum live space in the web browser on my 17-inch diagonal laptop with resolutions set to 1920 X 1080. Below it has been reduced into a 6.5 inch width space. As a point of information, I was able to fit and create a readable plot for up to 97 months of data. At that point, I had reduced the y-value format from 5.1 to 3.

NOTE: Since the width of the image below is less than half the original width, its readability has been impaired. You can approximately experience the original readability by using the PDF zoom feature.



The code used to create the graph above is simply an extension, from four VLINE statements to seven VLINE statements, of the equivalent four-line overlay plot above with three exceptions:

- values=(0 to 900 by 100) was appended to the yaxis statement
- fitpolicy=thin was appended to the xaxis statement
- title statements are different and fewer

The row labels for the y-value data label table are derived from the SAS variable label for the corresponding response variable which feeds that row. And that label also serves as the corresponding legend entry label. Since the length of the row label consumes precious width in the graph display area, it is advantageous to keep its length at a reasonable minimum. In a case like that above, for practical use of this graph, it would be essential to define the units for each of the economic indices. That could have been accomplished by using the legendlabel parameter on each of the seven VLINE statements. The legendlabel affects only the legend entry descriptions, and not the row labels in the y-value data label table.

GRAPH MACRO TOOLS YOU CAN USE IN SAS V9.3

My quest since 1981 for communication-effective graphs has yielded solutions that require extra coding. Macros can do that coding work for you, and give you simple parameters to specify instead, similar to using a SAS procedure. In this section, I share and show how to use such macros for common uses with high-value results.

Two of the macros, one for a pie chart and the other for a bar chart alternative to serve as a pie chart alternative when a pie chart is infeasible, overcome limitations in the new ODS GRAPHICS and SG procedures which are now an alternative to SAS/GRAPH. With those macros you need not use SAS/GRAPH to compensate for what is missing in the new technology. Their output charts are written to disk to later be inserted into Microsoft PowerPoint, Word, or Excel, or simply printed, but they could be web-enabled and packaged with ODS HTML. Earlier in this paper, corresponding non-macro graphic solutions were provided to meet these exact same needs. Those solutions are made easier and less error vulnerable by use of macros provided below.

The third macro, for a Maximally Informative Subsetted and Ranked Horizontal Bar Chart, does create web-enabled output.

The fourth macro, also for that Maximally Informative Subsetted and Ranked Horizontal Bar Chart, is intended to let you custom format the chart for post-creation use in Word or PowerPoint by manual insert from disk.

The remaining macros (provided both for the new technology, and for SAS/GRAPH—because SAS/GRAPH performs better as explained there) are provided to enable you to handle ultra-wide trend charts in a very communication-effective manner.

Some of the macro code is very lengthy and complex, but once you have copied it out of here and filed it in a macro library, its use is easy and quick. Sample invocations of the macro are used here to create output examples.

You do not need to read any of macro source code provided here. If you do, obviously you can use it as a starting point to create a derivative or alternative that you like better.

How To Use the Macros

Copy the code for each macro into NotePad, save the result with filetype sas in any folder location convenient to you where you have write access as well as read access. The filename that you use must match the string after the %MACRO prefix in the first line of the macro.

To run the macros, you must, as shown in the macro invocation example code, precede the macro invocation with this statement:

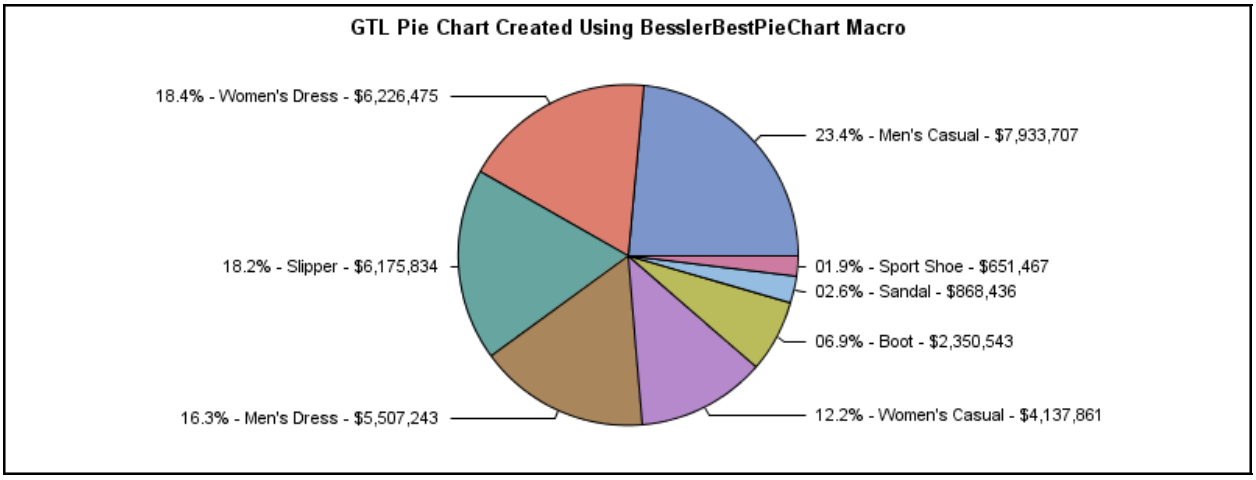
```
OPTIONS SASAUTOS=( ThePathToYourFolderGoesHere SASAUTOS);
```

An example of **ThePathToYourFolderGoesHere** could be:

```
"C:\MySASmacros"
```

When you submit code to run the macro, SAS will first look for it in your folder, and then it will look for it in your default macro library location, which has path reference SASAUTOS. For every macro that is invoked during your session or in your batch program execution, until you override the above OPTIONS statement, SAS will always look in your macro library first.

PIE CHART WITH ORDERING AND ALL RELEVANT NUMERIC INFORMATION



There is NOTHING ELSE you could need visually or numerically to have complete understanding of the information at this level, and ordering makes it easy to quickly assess the relative size of the shares of the whole.

Within the ODS GRAPHICS environment of Base SAS, there is no SG procedure to create a pie chart. Instead, a pie chart requires use of the cumbersome, and otherwise unnecessary, GTL (Graph Template Language). Below is the code for invocation of a macro to simplify the task of creating the pie chart above:

```
options sasautos=("D:\MySASmacros" sasautos);
/* First look in D:\MySASmacros for any macro to be invoked.
   If Not Found, then look in the default sasautos,
   which is the macro library shipped by SAS Institute,
   and maybe additional local SAS macro library(ies) at your site,
   which has (have) been linked to your SAS software installer.
   You can reverse the order of the search list above.
   You can include additional custom macro libraries in the search list above.
   If the same macro name is used in multiple libraries,
   the first one found is used. */

options mprint;

%BesslerBestPieChart27Jun2012
(Data=sashelp.shoes
,SliceLabelVar=Product
,SliceMeasureVar=Sales
,SliceMeasureFormat=dollar10.
,Order=Descending
,ChartTitle=GTL Pie Chart Created Using BesslerBestPieChart27Jun2012 Macro
,ChartFileName=PieChartCreatedUsingBesslerBestPieChart27Jun2012Macro
,ChartFolderName=D:\@WIILSU Nov2012\Results
,ChartHeight=300px
,ChartWidth=800px);
```

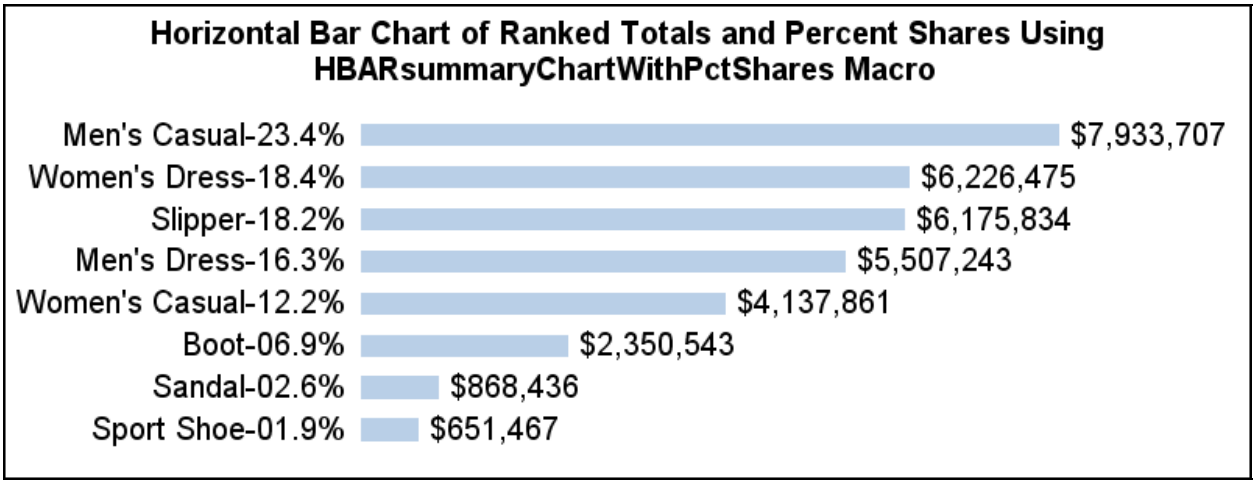
Here is the macro that was stored in **D:\MySASmacros** with filename **BesslerBestPieChart27Jun2012.sas** .

```
%macro BesslerBestPieChart27Jun2012
(Data=
,SliceLabelVar=
,SliceMeasureVar=
,SliceMeasureFormat=
,DecimalPositionsForPercents=1 /* can be 0 or any integer */
,Order=       /* valid values are descending or ascending */
,ChartTitle=
```

```
,ChartFileName=
,ChartFolderName=
,ChartHeight=
,ChartWidth=
);

* All parameters above, except Order are mandatory                  *;
* If Order is not specified, the SAS PROC SORT default is ascending *;

proc summary data=&Data nway;
class &SliceLabelVar;
var &SliceMeasureVar;
output out=ToPrep sum=TotalByClass;
run;

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

data ToChart;
length SliceNameWithPercentAndValue $ 256; /* over-sized, but that is harmless */
set ToPrep;
SliceNameWithPercentAndValue =
  trim(left(
  put(((TotalByClass / &GrandTotal) * 100),z4.&DecimalPositionsForPercents)
  )) ||
  '% - ' ||
  trim(left(&SliceLabelVar)) ||
  ' - ' ||
  trim(left(put(TotalByClass,&SliceMeasureFormat)));
run;

proc sort data=ToChart;
by &Order TotalByClass;
run;

proc template;
define statgraph BesslerBestPieChart27Jun2012;
  begingraph;
    entrytitle "&ChartTitle";
    layout region;
      piechart category=SliceNameWithPercentAndValue
               response=TotalByClass /
               datalabelcontent=(category)
               datalabellocation=callout
               otherslice=FALSE;
    endlayout;
  endgraph;
end;
run;

ods listing gpath="&ChartFolderName";
ods graphics on / reset=all
                  border=on
                  height=&ChartHeight
                  width=&ChartWidth
                  imagename="&ChartFileName";

proc sgrender data=ToChart template=BesslerBestPieChart27Jun2012;
run;
ods listing close; ods listing;

%mend BesslerBestPieChart27Jun2012;
```

BAR CHART OF SUMS WITH PERCENT SHARES: ALTERNATIVE FOR WHEN 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.



Just as the pie chart labels included values, not just slice names and percent shares, bar chart labels created in this macro COULD include values also, and the values at bar ends could be dropped. I leave that macro enhancement to the reader. See the code for the Subsetted and Ranked Horizontal Bar Chart later in this paper for another way to compose multi-function bar chart labels.

The main advantage of SGPLOT bar charts is that you can get the values at the bar ends with no extraordinary effort. A PROC GCHART bar chart by default places those values in a stack or table at the right margin of the graph area. To get them at the bar ends requires use of the ANNOTATE facility, which is more work—best avoided if possible.

Below is the code for invocation of a macro to simplify the task of creating the bar chart above:

```
options sasautos=("D:\MySASmacros" sasautos);
/* First look in D:\MySASmacros for any macro to be invoked.
   If Not Found, then look in the default sasautos,
   which is the macro library shipped by SAS Institute,
   and maybe additional local SAS macro library(ies) at your site,
   which has (have) been linked to your SAS software installer.
   You can reverse the order of the search list above.
   You can include additional custom macro libraries in the search list above.
   If the same macro name is used in multiple libraries,
   the first one found is used. */
options mprint;
%HBARsummaryChartWithPctShares
(Data=sashelp.shoes
,BarLabelVar=Product
,BarMeasureVar=Sales
,FontSize=16pt
,BarWidth=0.5
,Order=Descending
,ChartTitle=Horizontal Bar Chart of Ranked Totals and Percent Shares Using
HBARsummaryChartWithPctShares Macro
,ChartFileName=BarChartCreatedUsingHBARsummaryChartWithPctSharesMacro
,ChartFolderName=D:\@WIILSU Nov2012\Results
,ChartHeight=303px  /* NOTE: At 300px, every other bar label is omitted by SAS. */
,ChartWidth=800px); /* See discussion of FitPolicy=THIN earlier in this paper.  */
```

Here is the macro that was stored in **D:\MySASmacros** with filename **HBARsummaryChartWithPctShares.sas** .

```
%macro HBARsummaryChartWithPctShares
(Data=
,BarLabelVar=
,BarMeasureVar=
,BarMeasureFormat=
,FontSize=
,BarWidth=
,DecimalPositionsForPercents=1 /* can be 0 or any integer */
,Order= /* valid values are descending or ascending */
,ChartTitle=
,ChartFileName=
,ChartFolderName=
,ChartHeight=
,ChartWidth=
);

%if %upcase(&Order) EQ DESCENDING
%then %let Order = respdesc;
%else %let Order = respasc ;

proc summary data=&Data nway;
class &BarLabelVar;
var &BarMeasureVar;
output out=ToPrep sum=TotalByClass;
run;

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

data ToChart;
length BarNameWithPercent $ 256; /* over-sized, but that is harmless */
set ToPrep;
BarNameWithPercent =
  trim(left(&BarLabelVar)) || '-' ||
  trim(left(put(((TotalByClass / &GrandTotal) * 100),z4.1))) || '%';
run;

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

ods listing gpath="&ChartFolderName" style=styles.ListingWithNoFrame;

ods graphics on / reset=all
    border=on height=&ChartHeight width=&ChartWidth imagename="&ChartFileName";

title height=&FontSize "&ChartTitle";
proc sgplot data=ToChart;
hbar BarNameWithPercent / response=TotalByClass categoryorder=&Order
     datalabel datalabelattrs=(size=&FontSize) barwidth=&BarWidth nooutline;
yaxis display=(nolabel noline noticks) valueattrs=(size=&FontSize);
xaxis display=none;
run;
ods listing close; ods listing;

%mend HBARsummaryChartWithPctShares;
```

SOLUTIONS FOR A FINITE WORK DAY IN AN ERA OF INFORMATION OVERLOAD: SHOW THEM WHAT IS IMPORTANT . . . WITH THE MAXIMALLY INFORMATIVE SUBSETTED AND RANKED HORIZONTAL BAR CHART

For over a quarter century I have advocated and exploited use of the subsetted and ranked horizontal bar chart. I have long been hooked on the idea of trying to deliver only the most important. The most important can usually fit on one sheet of paper, and frequently, if not almost always, on one web page without having to scroll. Such limitation of information volume reminds me of the wisdom of Kenneth J. Wesley, my staff who once counseled me, when I was agonizing over a report for executive management, that "If it doesn't fit on one page, they won't read it." And I always remember the wisdom of Jim White, an expert on print document design, who said, "Let part stand for the whole."

In that spirit, I developed a macro that allowed the user to point to a data set and create a bar chart subsetted in any one of three ways: (1) Top N (where N was any integer); (2) all values above a cut-off; or (3) enough of the top values to account for the Top P Percent of the total measure of interest. Whereas I have long maintained that what will fit on a page, say, anywhere from the Top 10 to the Top 40 or 50, will usually account for 80 to sometimes 99% of the total measure of interest, a reliable approach to take when subsetting is to stop reporting as soon as the chart bars account for as much of the total as you feel is important to show. YOU pick the percent target with Option 3 above. Recently, I decided I wanted a client's reporting system users to be able to optionally look at the WHOLE list.

The macro that I am sharing here does NOT support that four-option capability. What was delivered for the client was actually a macro that not only created four versions of the ranking report, but also interlinked the four web graphs with hyperlinks—very cool and maximally convenient.

Here I provide a more limited function macro that successfully sizes the height of the image file for ANY number of the Top N bars **up to a maximum of 50** and links it forwards and backwards with an image sized large enough to display ALL of the bars. Of course, if you do not really want the ALL Bars companion chart, it is easy to modify the macro to omit that capability. (I could have built in the option, but the macro is complicated enough As Is.) If the Top N number of bars requested is not less than the total number of bars possible, only one web page is produced.

For this macro to perform reliably regardless of the number of Top N bars (up to 50), the SGPLOT HBAR chart image file must be sufficiently tall and/or the font used for bar labels must be significantly small to avoid automatic thinning of the bar labels. (See the earlier discussion in this paper of thinning tick mark values in line plots.) At the same time, the image file must be sufficiently small to fit in the browser window so that all of the Top N bars can be viewed without need for vertical scrolling. (The ALL Bars chart will require scrolling unless ALL is less than 51.)

So, there is a trade-off between: (a) making the bar label font small enough so that the desired number of Top N bars will not experience thinning; (b) making the bar label font small enough to that the Top N bar chart will not require vertical scrolling; and (c) making the bar label font large enough to be easily readable.

The viewing capacity of the expected target user depends on the brand and version number of the web browser, how many toolbars are in use in the web browser, and the vertical resolution of the monitor. My development and testing of the examples shown here were done using a monitor with 1920 X 1080 resolution, Internet Explorer Version 9, and no (what I regard as) extra toolbars.

The default font, which can be overridden by the macro user, is Albany AMT/Bold. This font is shipped with SAS software. The font size used by the macro is controlled internally, and varies by the value of TopN. The algorithm in the macro that controls font size also determines the appropriate number of Y pixels per bar based on the value of TopN. There is an overhead number of Y pixels which is independent of the value of TopN. The overhead is a macro parameter defaulted to what I found appropriate for the default choice of bar label font. The macro dynamically determines the total number of Y pixels needed. Macro internals must be revised if using a different font for the bar labels. It required experimentation and testing to determine the relationship between TopN, font size, Y pixels per bar, and Y pixels overhead—in pursuit of preventing the dreaded automatic thinning of bar labels.

The macro default font size for the title lines is 16 pt. Reducing it might facilitate fitting your chart in a browser window on a lower resolution monitor. The title lines are OUTSIDE of and above the image file. To save vertical space, you can avoid use of the optional fifth title line. That line can be used for optional diagnostic information during testing. When not testing, you can optionally provide any text, with any color of your choice (default is black). With or without such optional text, you can include (only in black) the run day, date, and time on title line 5.

NOTE: I later provide a non-web/non-HTML version of the macro, which can create an image file on disk to be imbedded manually in Microsoft Word or Microsoft PowerPoint, and perfectly sized for whichever use you need.

Adaptive SGPLOT Horizontal Bar Chart Macro Adjusts Image Height Based On Bar Count

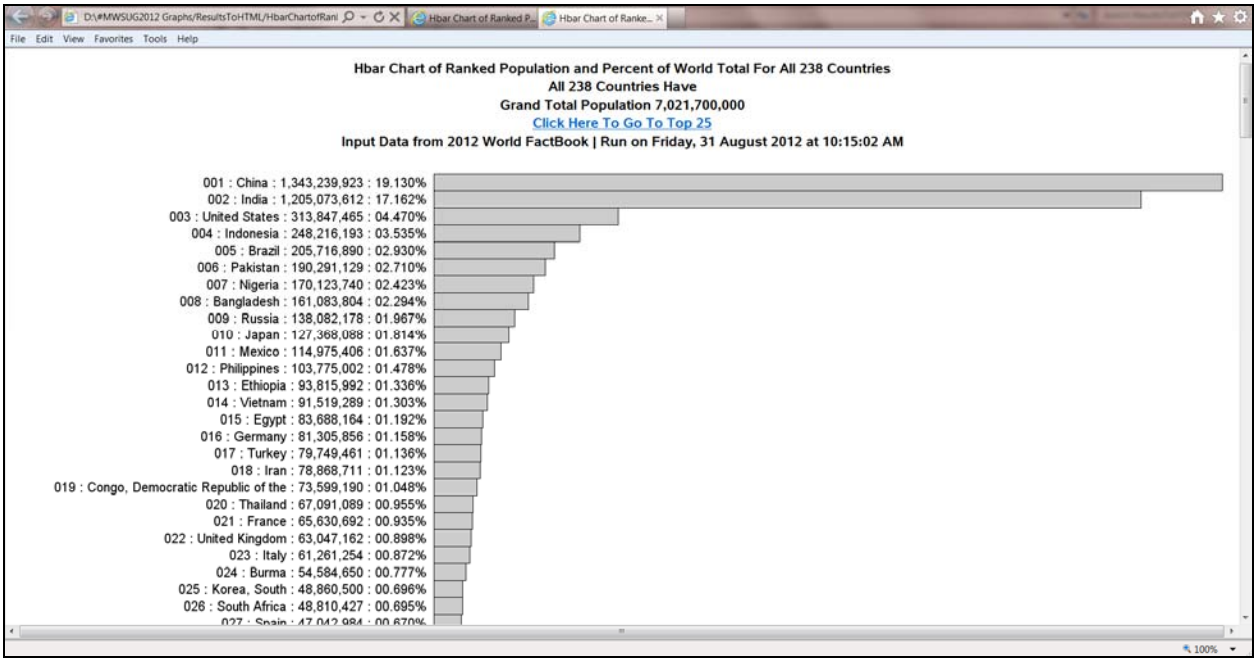
When producing a horizontal bar chart with a default size image file, software makes default use of the space. When using SAS/GRAPH PROC GCHART with a few bars, you get useless white space above and below, and with numerous enough bars you get adjacent bar label overlay. When using ODS Graphics PROC SGPLOT HBAR with a few bars, you get oversized bars to fill the space and disproportionate to the size of the bar labels, and with numerous enough bars some of the bar labels disappear (are "thinned"). The macro presented here right-sizes the image and uses SGPLOT HBAR. **It requires the CATEGORYORDER and VALUEATTRS features of Version 9.3 of SAS.**

As explained earlier, I am a strong advocate of subsetting information delivered in order to focus on what's important, but I recently finally stepped up to the reality of presenting information for which a vertically scrollable web graph of ranked horizontal bars with NO subsetting is desired. When you are presenting a subset of the data, it is useful to anticipate and be able to answer any question about the omitted data. The macro presented here permits you to create a pair of interlinked web graphs for a subset and for all of the data.

To deliver a tall horizontal bar chart (i.e., a very large number of bars), you obviously need to increase the pixel count for the height of the image sufficiently. To do this in a production environment, where no iterative loop of trial and adjustment is available, requires what I have long called "software intelligence", which is my personal favorite term for data-adaptive program design and construction. The key to the solution is a macro that dynamically sets the needed pixel count capacity for the height of the image based on bar count and dynamically sets the font size.

My later examples use the SASHELP.SHOES data set that every SAS user can access, but I first want to use data that involves a larger number of potential bars—i.e., population by country data from the 2012 World FactBook.

NOTE: TITLE5 is optional: the descriptive part is specified with an optional macro parameter, and the run day, date, & time can be omitted. Alternatively TITLE5 can instead deliver diagnostic information—e.g., if desired during testing.

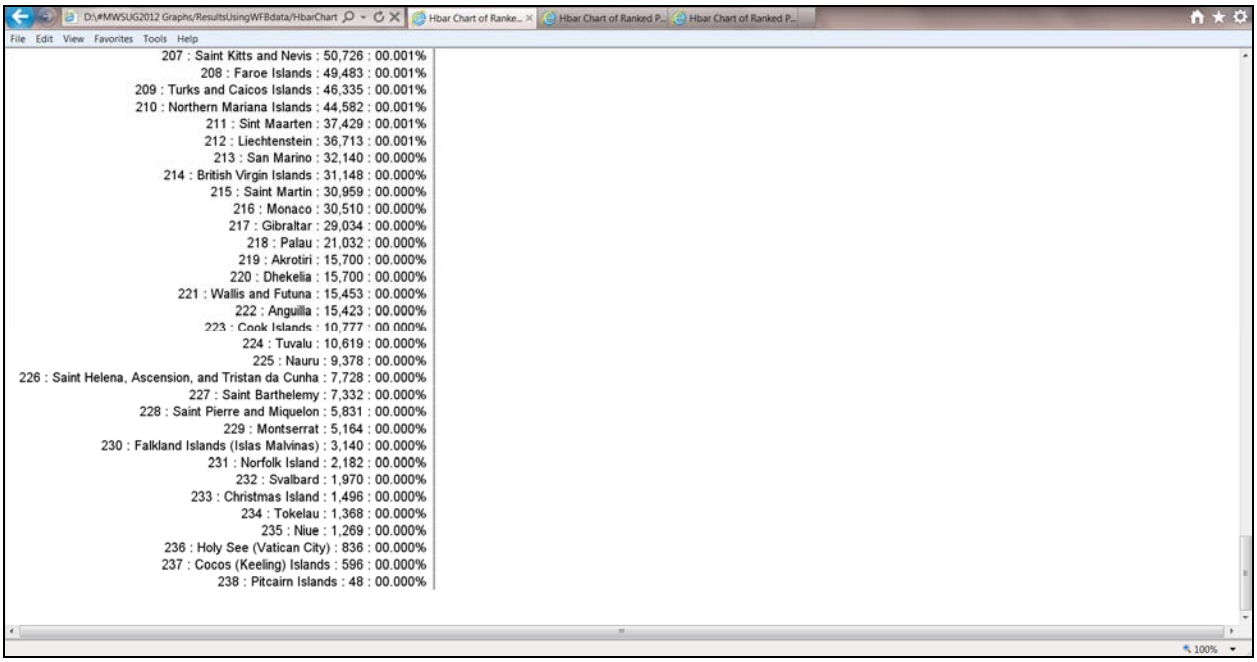


Here is an enlargement of the top four title lines:

Hbar Chart of Ranked Population and Percent of World Total For All 238 Countries
All 238 Countries Have
Grand Total Population 7,021,700,000
Click Here To Go To Top 25

The count of bars and the grand total of the measure of interest are dynamically generated by the macro processing. Note the link to a Top 25 subset of the data. That web page has a link back.

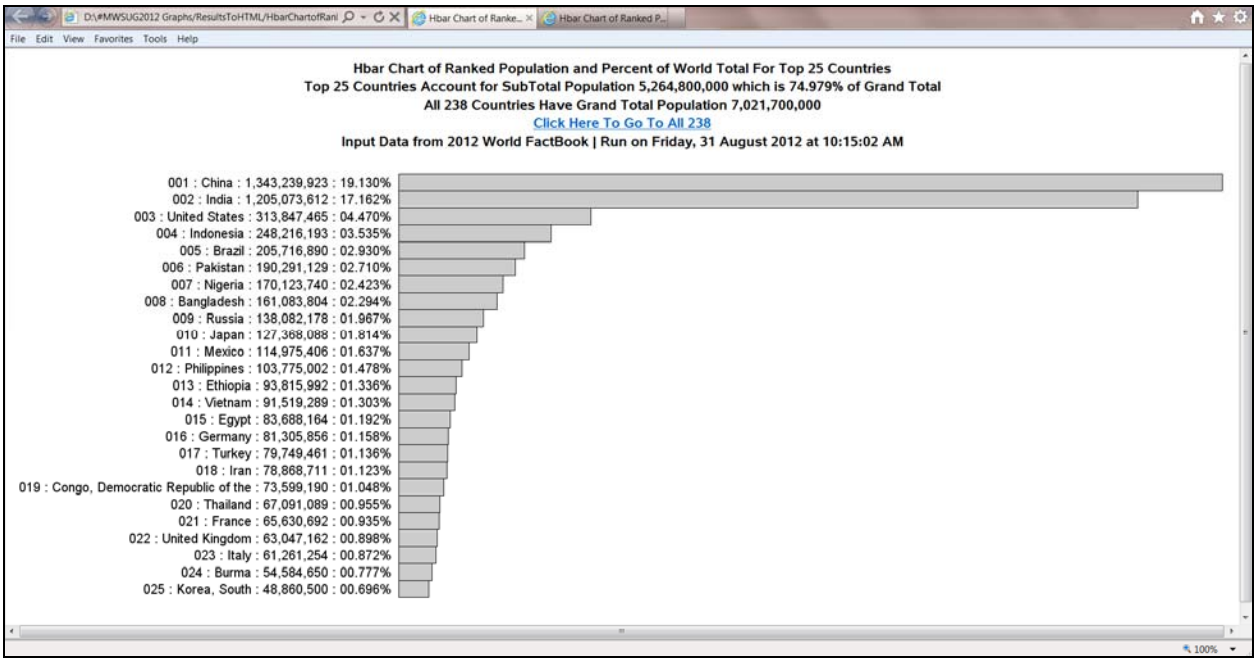
Here is the bottom of the All 238 Countries chart:



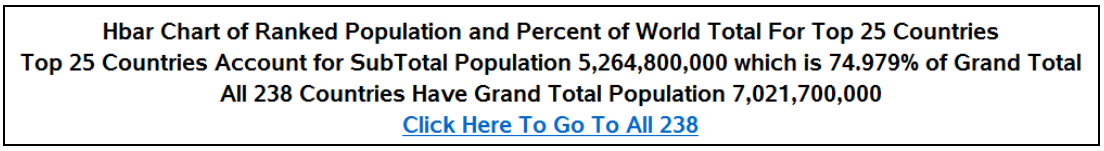
In the enlargement below, the bottom bars are nothing but a vertical line at this point, and percents of the whole, even to three decimal places, are 00.000%, but the country names and populations are still useful information.

233 : Christmas Island : 1,496 : 00.000%	
234 : Tokelau : 1,368 : 00.000%	
235 : Niue : 1,269 : 00.000%	
236 : Holy See (Vatican City) : 836 : 00.000%	
237 : Cocos (Keeling) Islands : 596 : 00.000%	
238 : Pitcairn Islands : 48 : 00.000%	

Here is the Top 25 Countries chart:



Here is an excerpt of the chart, showing only the title, subtitles, and the link back to the All 238 Countries chart:



The first subtitle provides the bar count, the subtotal from those bars, and their percent of the grand total. The second subtitle provides both the total bar count and their grand total. Both subtitles are dynamically generated by the macro.

The code was used to create web graphs for display on a wide-screen monitor with resolution 1920 X 1080. During initial testing with Xpixels=1875 to make full use of the browser window, PROC SGPLOT failed with this message:

```
ERROR: Java virtual machine exception. java.lang.OutOfMemoryError: Java heap space.
```

The problem could be avoided by reducing the chart image file width with Xpixels=1676. However, by doing some investigation at support.sas.com, by use of a search for “java heap space” in the Samples and SAS Notes (on web page http://support.sas.com/notes/index.html), I found http://support.sas.com/kb/31/184.html for which the title is “Problem Note 31184: "Java virtual machine exception" message issued with ODS Graphics and large data sets”. My example DOES NOT involve a “large data set”, but it does create a somewhat large image file. (Of course, what constitutes “large” is inherently ambiguous.) The recommended solution in the Problem Note is to make a change to the SASV9.CFG file. If you are using SAS on a standalone PC, you might be able to make the change. If using SAS on a server, you presumably will need to request the change to be made by the SAS server administrator.

NOTE: Instead of actually encountering the ERROR above, for some large image files, you might instead find this message in your SAS log:

```
WARNING: A very large output size of (1875, 6219) is in effect. This could make Java
VM run out of memory and result in some Java exceptions. You should reduce the output
size or DPI settings.
```

where the width and height pixel counts, here 1875 and 6219 respectively, will be replaced by the values appropriate to YOUR graph. These numbers are determined by parameters on the ODS GRAPHICS statement, namely:

```
      width=1875px height=6219px
```

where 1875 comes from a macro invocation parameter and 6219 is dynamically set by the macro logic.

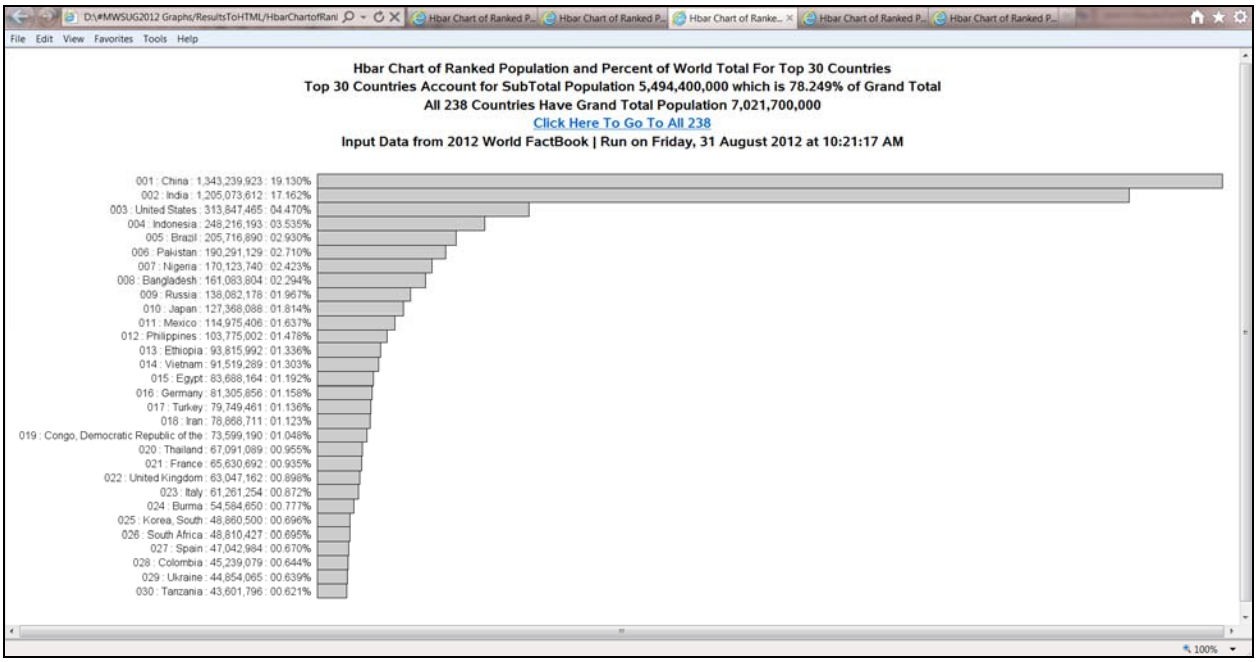
Here is the macro invocation code used to create the two web-interlinked charts shown above:

```
%SubsettedRankingHbarChartsToHTML(
data=DataLib.PopulationByCountryPerWFB2012
,TopN=25
,RptPath=D:\#MWSUG2012 Graphs\ResultsToHTML
,Title1_Prefix=Hbar Chart of Ranked Population and Percent of World Total For
,Title1_Suffix=Countries
,Title5=Input Data from 2012 World FactBook /* specifying Title5 is optional */
/* ,RunDayDateTimeInTitle5=NO */ /* Uncomment prior to turn off macro default YES */
,BarLabelVar=Country
,BarMeasureVar=Population
/* ,BarMeasureDescription=Population AsOf 2012 WFB Estimate */
/* With the above commented out, chart titles use BarLabelVar value as description */
,BarMeasureFormat=comma13.
,Xpixels=1875); /* requires a 1920 pixel width monitor */
```

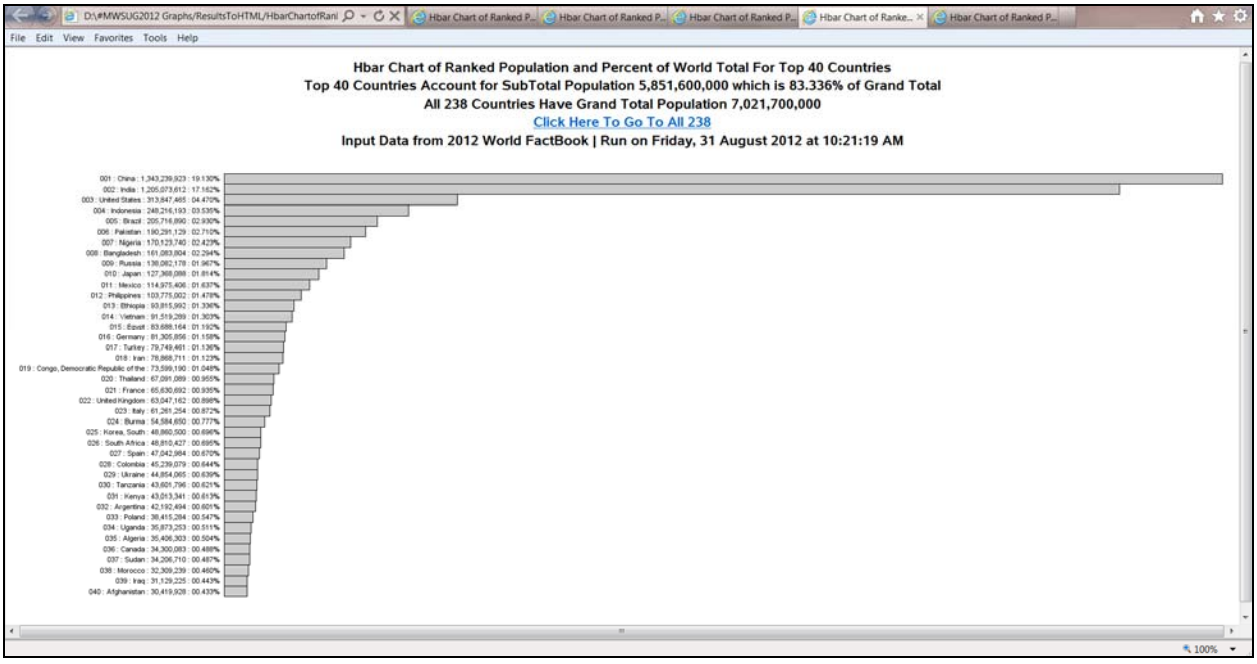
Every macro parameter shown above must be assigned, except **`Title5`** which defaults to null. It is important to realize that the macro provided in this paper does NOT verify that you have assigned all of the mandatory parameters, nor that you have provided appropriate values. It does, however, make some checks, and, if you do not override the macro default selection for TextFont, it selects a point size for the bar labels small enough to prevent thinning by PROC SGPLOT, but not smaller than necessary. Earlier in the paper, I discussed and demonstrated the unfortunate phenomenon of thinning of axis tick mark values, but there in the context of line charts, not bar charts.

Remember that thinning occurs without any WARNING message in the SAS log. If you use a different font for the bar labels, you might need to change macro internal code and the **YpixelsOverhead** macro parameter (which is defaulted to 31) in order to prevent thinning of bar labels. For values of TopN greater than 25, the macro uses progressively smaller point sizes for the TextFont, as shown in the examples below. Values of TopN greater than 50 are infeasible if you want readable labels and to keep all of the bars within the web browser window without scrolling. The macro rejects use of infeasible values for TopN.

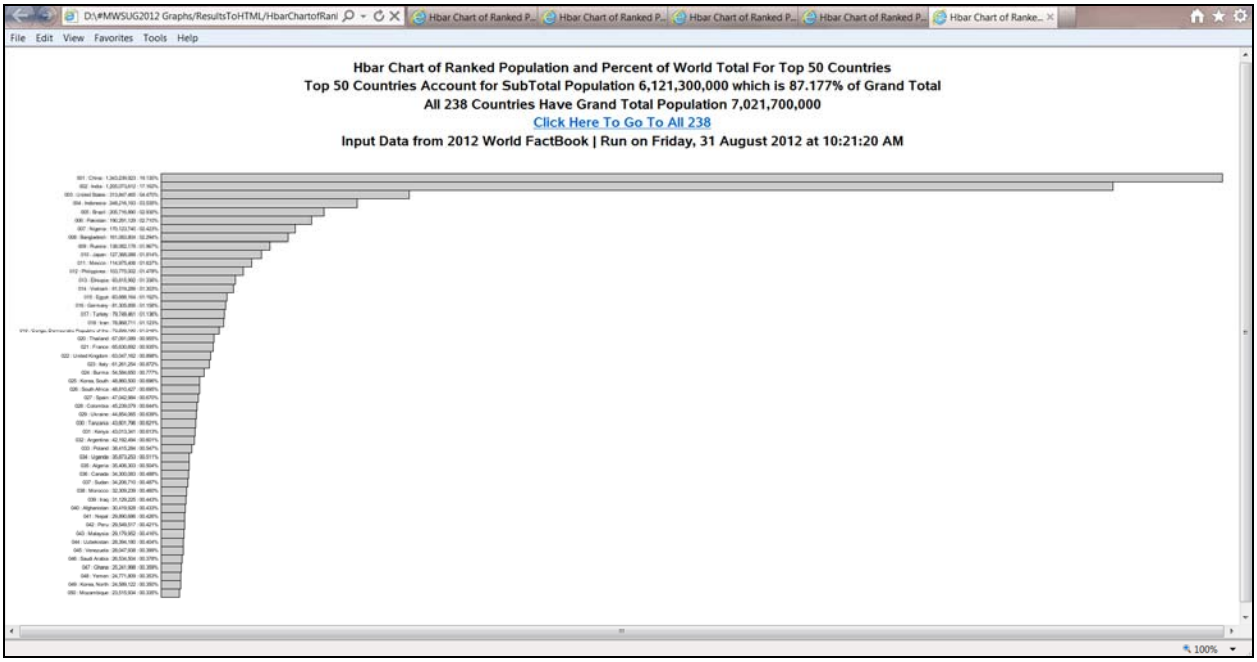
Below is a Top 30 chart. The only macro invocation code change was to set TopN=30.



When shrunk below to fit within 6.5 inches for this paper, the Top 40 chart might, with some difficulty, be readable, especially if you are already familiar with Country names. The digits on the bar labels are even harder to read.



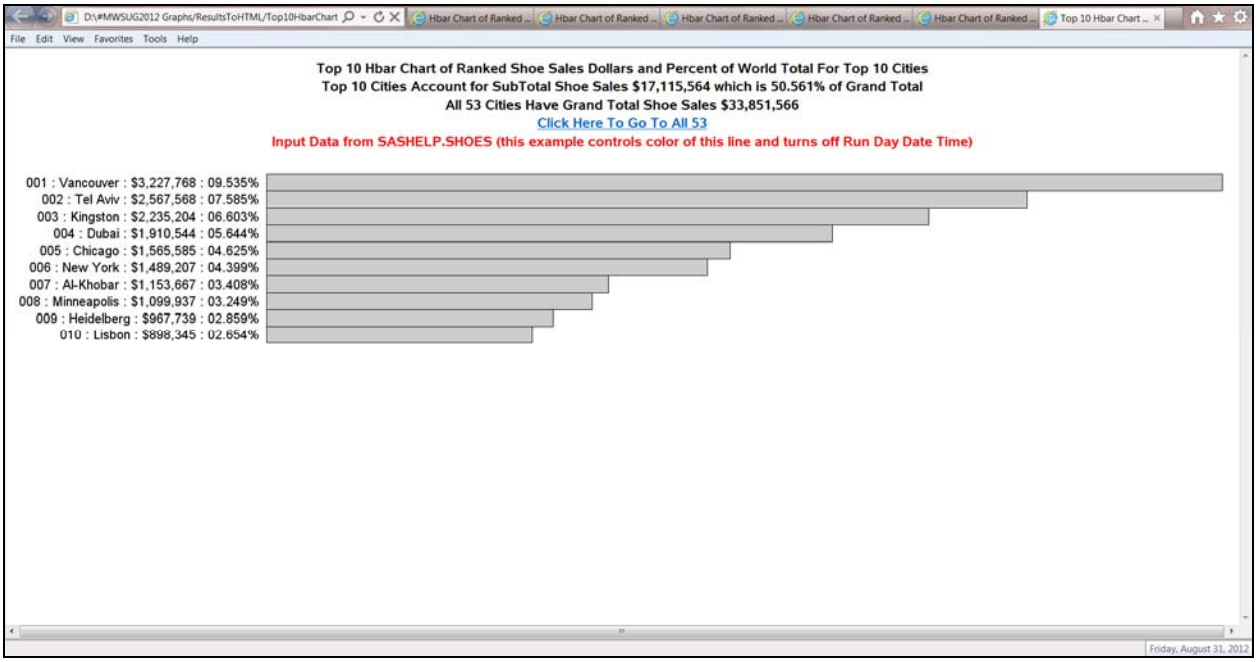
The Top 50 chart was easily readable on the 17-inch diagonal monitor used for development, which has a screen width of 15 inches. Here, shrunk below to fit within 6.5 inches for this paper, the bar labels are unreadable.



The SAS-Institute-developed software-internal code does not adjust the font size. The macro does that.

Below is an example for a smaller TopN value, using the SASHELP.SHOES data set that any SAS user can access.

The Top 10 chart automatically “moves” the bars up the browser window because the macro assigns a shorter height to the image file. Without the macro, the software itself would work with whatever height that is assigned, if any height IS assigned, on the ODS GRAPHICS statement, or with its own default of 480 pixels, and would center the bars vertically within that image file, making the bars as wide as possible in the available space, even if their width would be unnecessarily large and out of proportion to the bar label width. Such results will be shown later in the paper. Below is the result for 10 bars, using the macro.



Here is the macro invocation code used to create the Top 10 Chart:

```
%SubsettedRankingHbarChartsToHTML(
data=SASHELP.SHOES
,TopN=10
,RptPath=D:\#MWSUG2012 Graphs\ResultsToHTML
,Title1_Prefix=Top 10 Hbar Chart of Ranked Shoe Sales Dollars and Percent of World
Total For
,Title1_Suffix=Cities
,Title5=Input Data from SASHELP.SHOES (this example controls color of this line and
turns off Run Day Date Time)
,Title5_Color=red
,RunDayDateTimeInTitle5=NO
,BarLabelVar=Subsidiary
,BarMeasureVar=Sales
,BarMeasureDescription=Shoe Sales
,BarMeasureFormat=dollar11.
,Xpixels=1875); /* requires a 1920 pixel width monitor */
```

Unless you override the macro's default choices for its TextFont or YpixelsOverhead parameters, the SubsettedRankingHbarChartsToHTML macro will deliver your output graph with no risk of missing bar labels due to thinning by the SAS software. There is a limit of 50 on TopN, but, above that, the font size needed to avoid thinning would have to be so small as to be unreadable.

NOTE: If the expected target user monitor has a vertical resolution lower than 1080 pixels, the macro internals and YpixelsOverhead value will need to be revised if you wish the entire bar chart, for bar counts beyond some limit (unique to the monitor resolution), to be viewable without vertical scrolling—which is an important design objective.

Here is the code for the SubsettedRankingHbarChartsToHTML macro:

```
%macro SubsettedRankingHbarChartsToHTML(
data=
,RptPath=
,Title1_Prefix=
,Title1_Suffix=
,UseTitle5ForDiagnostics=NO /* YES could be useful during testing a revised macro */
,Title5= /* Title5 is optional */
,Title5_Color=black
,RunDayDateTimeInTitle5=YES /* If using TITLE5 for diagnostic
                               or for a custom message,
                               it might be wise to turn off Run Day, Date, & Time
                               to prevent wrap of a too long title line, which could
                               push the bottom of chart below the browser window. */
,TitlesJustify=Center /* Center is actually the SAS default.
                         If titles were imbedded in the graph,
                         SGPLOT HBAR chart centering of titles would be defective.
                         Left Justification would be the better choice.
                         ODS HTML NOGTITLE (used below in macro) makes Center OK. */
,TopN= /* subset number of bars to be shown */
,BarLabelVar=
,BarMeasureVar=
,BarMeasureFormat=
,BarMeasureDescription= /* Macro does not retrieve SAS var label even if one exists.
                           If BarMeasureDescription not specified,
                           then SAS var name (BarMeasureVar) is used instead. */
,BarColor=CXCCCCCC /* light grey */
,TitleFont='Albany AMT/Bold'
,TitleFontPointSize=16
,TextFont='Albany AMT/Bold'
,Xpixels=     /* 1875 is maximum to avoid horizontal scrolling on 1920-pixel monitor.
                 Select a value based on the expected viewing monitor.
                 REDUCE Xpixels if java heap space out of memory & SGPLOT failure
```

```
                when bar count of the All Bars companion chart is high.
                The memory problem is announced with these messages in the SAS log:
                ERROR: Java virtual machine exception.
                       java.lang.OutOfMemoryError: Java heap space.
                NOTE: The SAS System stopped processing this step because of errors.
                Alternatively, you might get relief by changing the SASV9.CFG file,
                following the instructions
                in SAS Note http://support.sas.com/kb/31/184.html
                If you are using SAS on a server, you will need to request the change
                be made by the server SAS Administrator. */
,YpixelsOverhead=31 /* 31 YpixelsOverhead is for use with TextFont='Albany AMT/Bold'
                       and the rule inside the macro for choosing
                       TextFontPointSize and YpixelsPerBar based on the value of TopN.
                       YpixelsOverhead, YpixelsPerBar, TopN are used inside the macro
                       to compute the total Y pixels for height of the image file. */
);

%if %upcase(&UseTitle5ForDiagnostics) EQ YES AND %length(&Title5) NE 0 %then %do;
  %put USER ERROR: With the &sysmacroname macro, if you specify a value for Title5,
you must specify UseTitle5ForDiagnostics=NO;
  %GoTo MacExit;
%end;

%if "%upcase(&TextFont)" NE "'ALBANY AMT/BOLD'" %then %do;
  %put USER ERROR: This &sysmacroname macro is currently configured to only work with
the Albany AMT/Bold as TextFont parameter;
  %GoTo MacExit;
%end;

%if %length(&BarMeasureDescription) EQ 0
%then %let BarMeasureDescription = &BarMeasureVar;

%if %eval(&TopN LE 25)
%then %do;
  %let TextFontPointSize = 14;
  %let YpixelsPerBar = 26;
%end;
%else
%if %eval(&TopN LE 30)
%then %do;
  %let TextFontPointSize = 12;
  %let YpixelsPerBar = 21.67;
%end;
%else
%if %eval(&TopN LE 40)
%then %do;
  %let TextFontPointSize = 8;
  %let YpixelsPerBar = 16.25;
%end;
%else
%if %eval(&TopN LE 50)
%then %do;
  %let TextFontPointSize = 6;
  %let YpixelsPerBar = 13;
%end;
%else %do;
  %put USER ERROR: This &sysmacroname macro is currently configured to only work with
values less than or equal to 50 for the TopN parameter;
  %GoTo MacExit;
%end;

DATA _NULL_;
RunDayDateTimeText = 'Run on ' || TRIM(LEFT(PUT(DATE(),weekdatx37.))) ||' at ' ||
```

```
  TRIM(LEFT(PUT(TIME(),timeampm11.)));
CALL SYMPUT('RunDayDateTime',TRIM(LEFT(RunDayDateTimeText)));
RUN;

PROC SUMMARY DATA=&data nway;
CLASS &BarLabelVar;
VAR &BarMeasureVar;
OUTPUT OUT=Summed(drop=_type_ _freq_) SUM=;
RUN;

PROC SQL NOPRINT;
SELECT COUNT(&BarLabelVar) , SUM(&BarMeasureVar)
  INTO :CountOfAllBars     , :BarMeasureGrandTotal
  FROM Summed;
QUIT;

DATA _NULL_;
LENGTH ForSYMPUT 8;
ForSYMPUT = &BarMeasureGrandTotal;
CALL SYMPUT('GrandTotal',TRIM(LEFT(PUT(ForSYMPUT,&BarMeasureFormat))));
RUN;

PROC SORT DATA=Summed;
BY DESCENDING &BarMeasureVar;
RUN;

DATA ALLtoChart_WithBarLabels(drop=BarLabelSuffix);
LENGTH BarLabel $ 256 BarLabelSuffix $ 7;
SET Summed;
BarLabelSuffix =
  TRIM(LEFT(PUT(((&BarMeasureVar / &BarMeasureGrandTotal) * 100),6.3))) || '%';
IF LENGTH(BarLabelSuffix) EQ 6
THEN BarLabelSuffix = '0' || BarLabelSuffix;
BarLabel = TRIM(LEFT(PUT(_N_,z3.))) || ' : ' || /* assumes max possible is 999 bars */
  TRIM(LEFT(&BarLabelVar)) || ' : ' ||
  TRIM(LEFT(PUT(&BarMeasureVar,&BarMeasureFormat))) || ' : ' || BarLabelSuffix;
RUN;

OPTIONS OBS=&TopN;

DATA TOPNtoChart_WithBarLabels;
SET ALLtoChart_WithBarLabels;
RUN;

OPTIONS OBS=MAX;

%macro TopN_OR_All(Which=);

%let Which = %upcase(&Which);

%if &Which EQ TOPN %then %do;

PROC SQL NOPRINT;
SELECT SUM(&BarMeasureVar)
  INTO :BarMeasureSubTotal
  FROM TopNToChart_WithBarLabels;
QUIT;

DATA _NULL_;
LENGTH ForSYMPUT ForSYMPUT1 ForSYMPUT2 8;
ForSYMPUT = &BarMeasureSubTotal;
CALL SYMPUT('SubTotal',TRIM(LEFT(PUT(ForSYMPUT,&BarMeasureFormat))));
ForSYMPUT1 = &BarMeasureSubTotal;
```

```
ForSYMPUT2 = &BarMeasureGrandTotal;
CALL SYMPUT('SubTotalPercentOfGrandTotal',
  TRIM(LEFT(PUT(( ForSYMPUT1 / ForSYMPUT2 ) * 100 , 6.3))));
RUN;

%end;

%if &Which EQ TOPN
%then %let CountOfSelectedBars = &TopN;
%else %let CountOfSelectedBars = &CountOfAllBars;

%let Y_pixels = %eval( %sysfunc(CEIL( %sysevalf(&CountOfSelectedBars * &YpixelsPerBar)
)) + &YpixelsOverhead );

%if &Which EQ ALL
%then %do;
  %let Title1 = &Title1_Prefix.%str( All )&CountOfAllBars &Title1_Suffix;
  %let Title2 = All &CountOfAllBars &Title1_Suffix Have;
  %let Title3 = Grand Total &BarMeasureDescription &GrandTotal;
  %let Title4_LinkDescription = Top &TopN;
%end;
%else %do;
  %let Title1 = &Title1_Prefix.%str( Top )&TopN &Title1_Suffix;
  %let Title2 = Top &CountOfSelectedBars &Title1_Suffix
    Account for SubTotal &BarMeasureDescription &SubTotal
    which is &SubTotalPercentOfGrandTotal% of Grand Total;
    /* Because Title2 is delivered by HTML source code
          due to the specification of ODS HTML . . . NOGTITLE
          any extra blanks between the three parts of Title2
          on three separate lines in the %LET statement above
          will be compressed down to one blank. */
  %let Title3 =
    All &CountOfAllBars &Title1_Suffix Have Grand Total &BarMeasureDescription
&GrandTotal;
  %let Title4_LinkDescription = All &CountOfAllBars;
%end;

%if %eval(&CountOfAllBars GT &TopN) %then %do;

  %let FileName_Temp = %sysfunc(COMPRESS(&Title1,' '));
  %let FileNameLinkTo_Temp =
%sysfunc(COMPRESS(&Title1_Prefix.&Title4_LinkDescription&Title1_Suffix));

  %let FileName = &FileName_Temp._LinkedTo_&FileNameLinkTo_Temp;
  %let FileNameLinkTo = &FileNameLinkTo_Temp._LinkedTo_&FileName_Temp;

%end;
%else %let FileName = %sysfunc(COMPRESS(&Title1,' '));

%if &UseTitle5ForDiagnostics EQ YES %then %do;
  %let Title5ForDiagnostics =
       &CountOfSelectedBars bars, &YpixelsPerBar Ypixels per bar,
       &YpixelsOverhead Ypixels overhead, &Y_pixels Total Ypixels, &Xpixels Xpixels,
          Titles &TitleFontPointSize.pt &TitleFont, Bar Labels &TextFontPointSize.pt
&TextFont;
  %let Title5ForDiagnostics = %sysfunc(compbl(%quote(&Title5ForDiagnostics)));
%end;

ODS GRAPHICS ON / RESET=ALL BORDER=OFF SCALE=OFF
    HEIGHT=&Y_pixels.px WIDTH=&Xpixels.px
    IMAGENAME="&FileName";

FOOTNOTE;
```

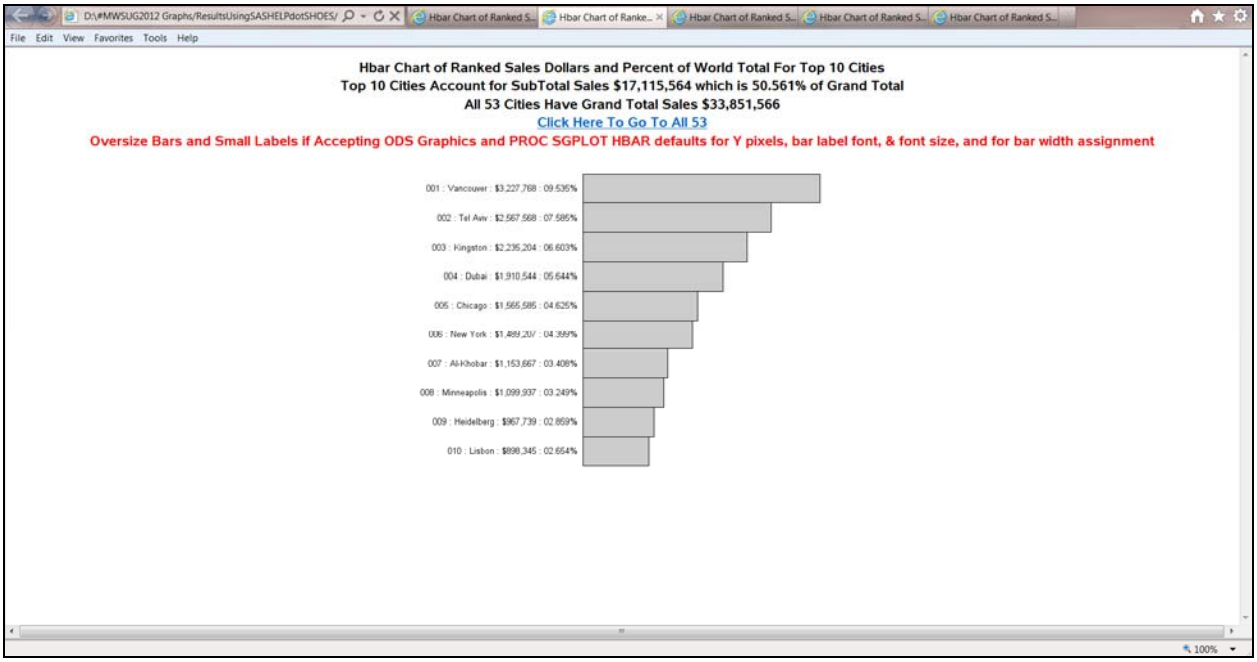
```
ODS NORESULTS;
ODS LISTING CLOSE;

ODS HTML PATH="&RptPath" (URL=NONE) STYLE=Styles.MinimalWithNoFrame
  NOGTITLE NOGFOOTNOTE
  BODY="&FileName..html" (TITLE="&Title1");

TITLE1 JUSTIFY=&TitlesJustify FONT=&TitleFont HEIGHT=&TitleFontPointSize PT "&Title1";
TITLE2 JUSTIFY=&TitlesJustify FONT=&TitleFont HEIGHT=&TitleFontPointSize PT "&Title2";
TITLE3 JUSTIFY=&TitlesJustify FONT=&TitleFont HEIGHT=&TitleFontPointSize PT "&Title3";
TITLE4 JUSTIFY=&TitlesJustify FONT=&TitleFont HEIGHT=&TitleFontPointSize PT
%if %eval(&CountOfAllBars GT &TopN)
%then %do;
  LINK="&FileNameLinkTo..html" "Click Here To Go To &Title4_LinkDescription";
%end;
%else %do;
  "You requested the Top &TopN &BarMeasureDescription., but there are only
&CountOfAllBars &Title1_Suffix";
%end;
%if %upcase(&UseTitle5ForDiagnostics) EQ YES
  or
    %length(&Title5) NE 0
  or
    %upcase(&RunDayDateTimeInTitle5) EQ YES
%then %do;
TITLE5 JUSTIFY=&TitlesJustify FONT=&TitleFont HEIGHT=&TitleFontPointSize PT
  %if %upcase(&UseTitle5ForDiagnostics) EQ YES %then %do;
  COLOR=red HEIGHT=14 PT /* make this long text string smaller */
  "&Title5ForDiagnostics "
  %end;
  %else
  %if %length(&Title5) NE 0 %then %do;
  COLOR=&Title5_Color
  "&Title5 "
  COLOR=black
  %end;
  %if %upcase(&RunDayDateTimeInTitle5) EQ YES
  %then %do;
  "| &RunDayDateTime"
  %end;
  ;
%end;

PROC SGPLOT DATA=&Which.toChart_WithBarLabels;
HBAR BarLabel / RESPONSE=&BarMeasureVar
  CATEGORYORDER=RESPDESC BARWIDTH=1 FILL FILLATTRS=(COLOR=&BarColor) OUTLINE;
YAXIS DISPLAY=(NOLABEL NOLINE NOTICKS) VALUEATTRS=(SIZE=&TextFontPointSize PT);
XAXIS DISPLAY=NONE;
RUN;

ODS HTML CLOSE;
ODS LISTING;

%mend  TopN_OR_All;

%macro RemoveCraphAreaFrameFromStyle
(ParentStyle=,StyleWithNoFrame=);

PROC TEMPLATE;
DEFINE STYLE &StyleWithNoFrame;
  PARENT=&ParentStyle;
  CLASS GRAPHWALLS / FRAMEBORDER=OFF;
  /* remove a useless box around the bars */
```

```
END; RUN;
%mend  RemoveCraphAreaFrameFromStyle;

%RemoveCraphAreaFrameFromStyle
(ParentStyle=Styles.Minimal
,StyleWithNoFrame=Styles.MinimalWithNoFrame);

%TopN_OR_All(Which=All);
%if %eval(&CountOfAllBars GT &TopN) %then %do;
  %TopN_OR_All(Which=TopN);
%end;

%MacExit:

%mend  SubsettedRankingHbarChartsToHTML;
```

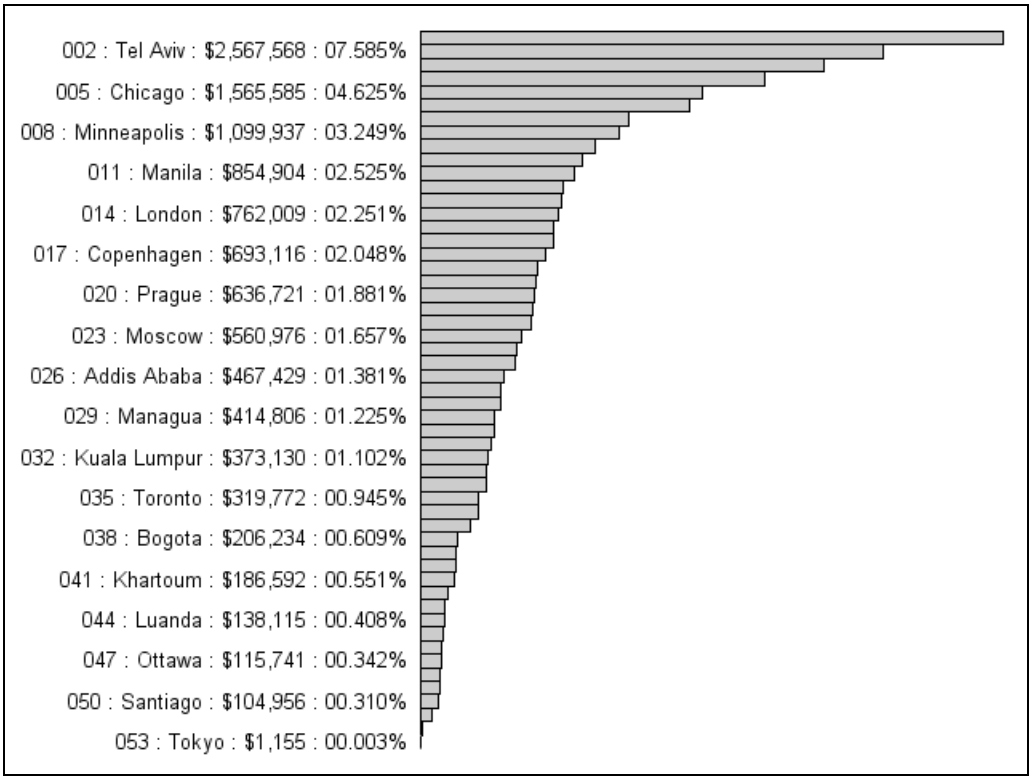
NOTE: The macro above might look rather daunting. If you like what it does As Is, just file it in a macro library, use **`OPTION SASAUTOS=`** to point your SAS program at the macro library, and invoke the macro in a manner shown in the sample invocations. Of course, you also can modify the macro if you wish.

On the following pages are a demonstration of what happens if you let ODS Graphics and PROC SGPLOT "do what comes naturally"—i.e., you create the same chart, but remove the built-in data-adaptive software intelligence from the macro.

Below is the result for a Top 10 chart, NOT using the macro's built-incontrols for image height sizing and font sizings. It also accepts the default image width (640 pixels), not just the default image height (480 pixels).

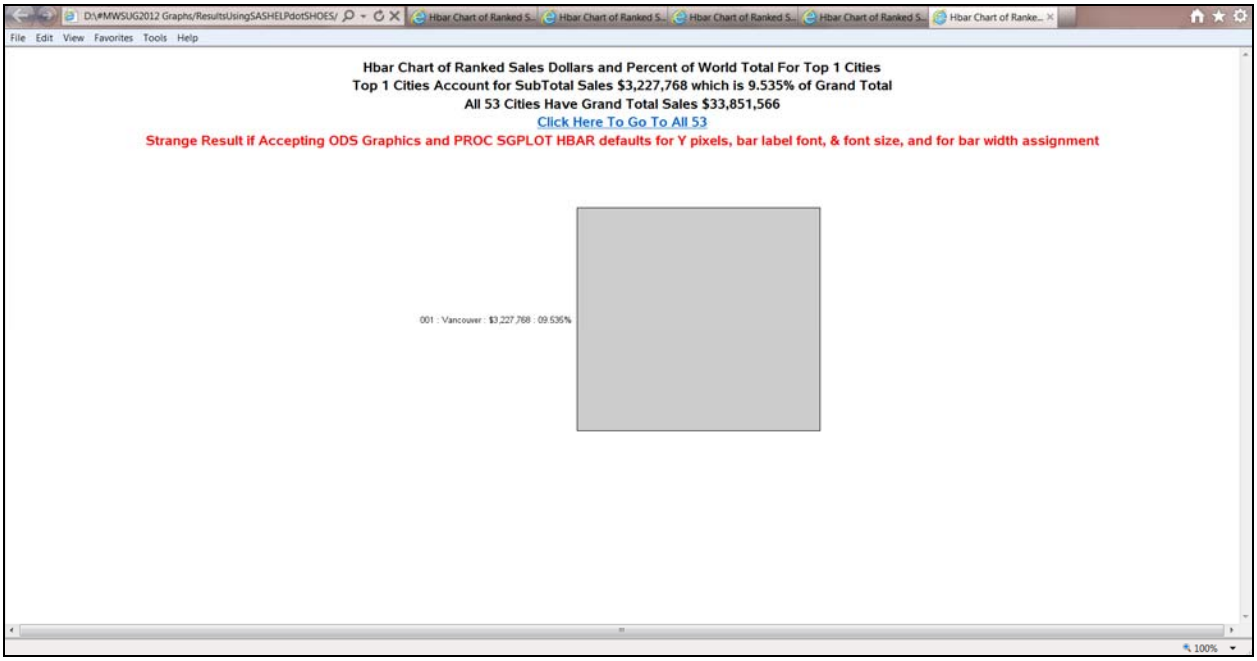


It is instructive to also see the unusable All 53 Cities chart which is the linked companion of the chart in the web page above. Inserted below is the image file (shrunk to 4 inches high from 5 inches in order to fit on this page), not a screen capture of the web page that imbeds it.



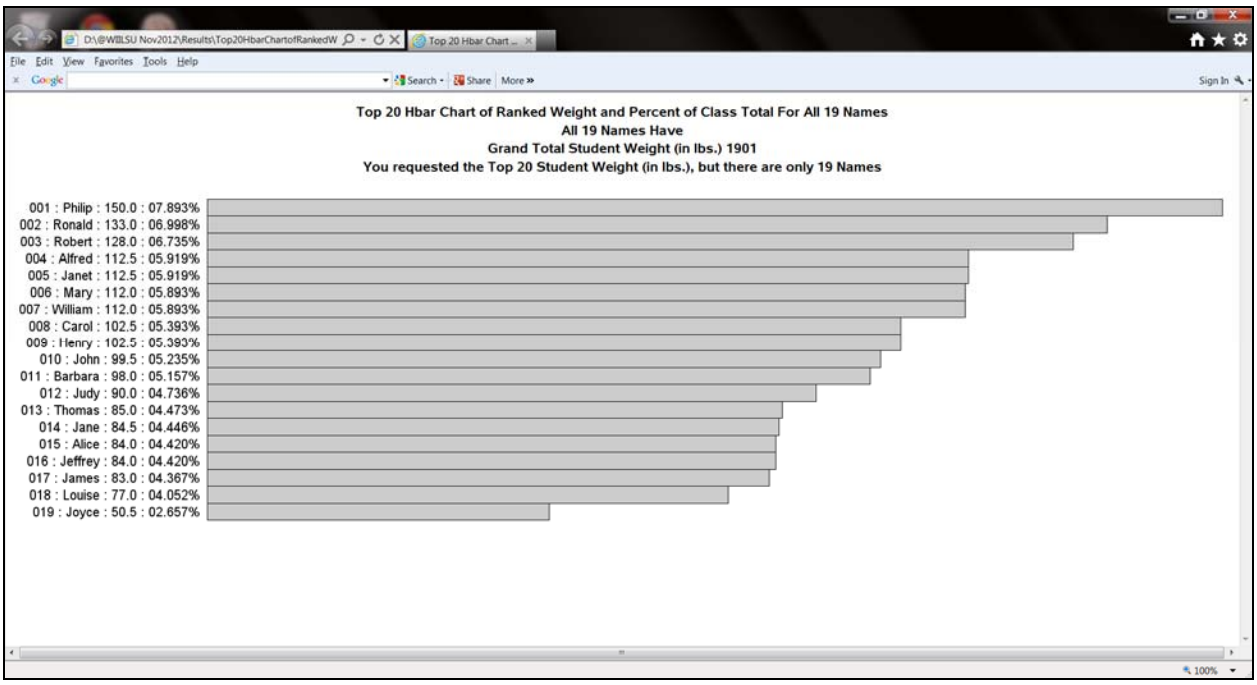
ODS Graphics PROC SGPLOT inflicts the above thinning of the bar labels with no mention in the SAS log.

To dramatize the problem further, below is the result for a Top 1 chart, NOT using the macro's built-in controls.



The undesirable results shown above from using defaults can be overcome by manually adding, changing, and tuning some additional ODS Graphics and/or PROC SGPLOT HBAR options. The purpose of a macro solution alternative is to allow your unchanging production job program, or your written-one-time program invoked by a real-time request, to simply identify the data set, the variables, a SAS format for the measurement variable, optionally a description for the measurement variable (to be used in titles instead of the measurement variable name), a title prefix, a title suffix to provide a title-appropriate label for the category variable, the value for TopN, and a value appropriate for the pixel width of the expected target monitor, and to *get the result right the first time every time*.

Before closing this section, below is a demonstration of what happens, IF USING the macro, when requesting a subset TopN that is equal to or exceeds the maximum number of possible bars. Note the message on title line 4. Rather than failing or rejecting the request, the macro is forgiving and informative.



Right-Sizing Subsetted Ranking Hbar Chart Macro For PowerPoint Slides Or MS Word Documents

The macro provided in this section is intended to provide all of the automatic data summarization, data management, layout formatting, and dynamic highly informative titling done by the macro presented in the prior section to produce interlinked web charts, but instead produces only one chart, the subsetted version, and allows you to easily control the dimensions. The examples below use pixel dimensions appropriate to fill a PowerPoint slide and to fill a portrait-oriented American Letter size page with one-inch margins.

In the case of PowerPoint slides, it might be a convenience to include all possible text related to the image inside the image itself, rather than having to provide the text with PowerPoint on the slide after (or before) inserting image. However, it is not difficult to build the slide in two parts, and keeping some text in the slide does permit one to easily change the text without rerunning the image creation. To leave some vertical space for on-slide-provided text simply requires adjusting the image height parameter when invoking the macro. The pixel count used in the examples here for full-height slides assumes that the physical height of the slide is 7.5 inches. Based on how much vertical space you want to be able to use for on-slide text, you need to simply reduce your Y pixel count proportionately.

In the case of a Word document, you usually might not want to fill the page unless the graph is very dense and will not communicate well in a smaller “window”. To deliver it in a smaller size for Word likewise simply means scaling down the example parameters used below. Also, if you do want to present it in a full-size landscape page rather than portrait, you only need to switch your use of the example Y pixel and X pixel counts.

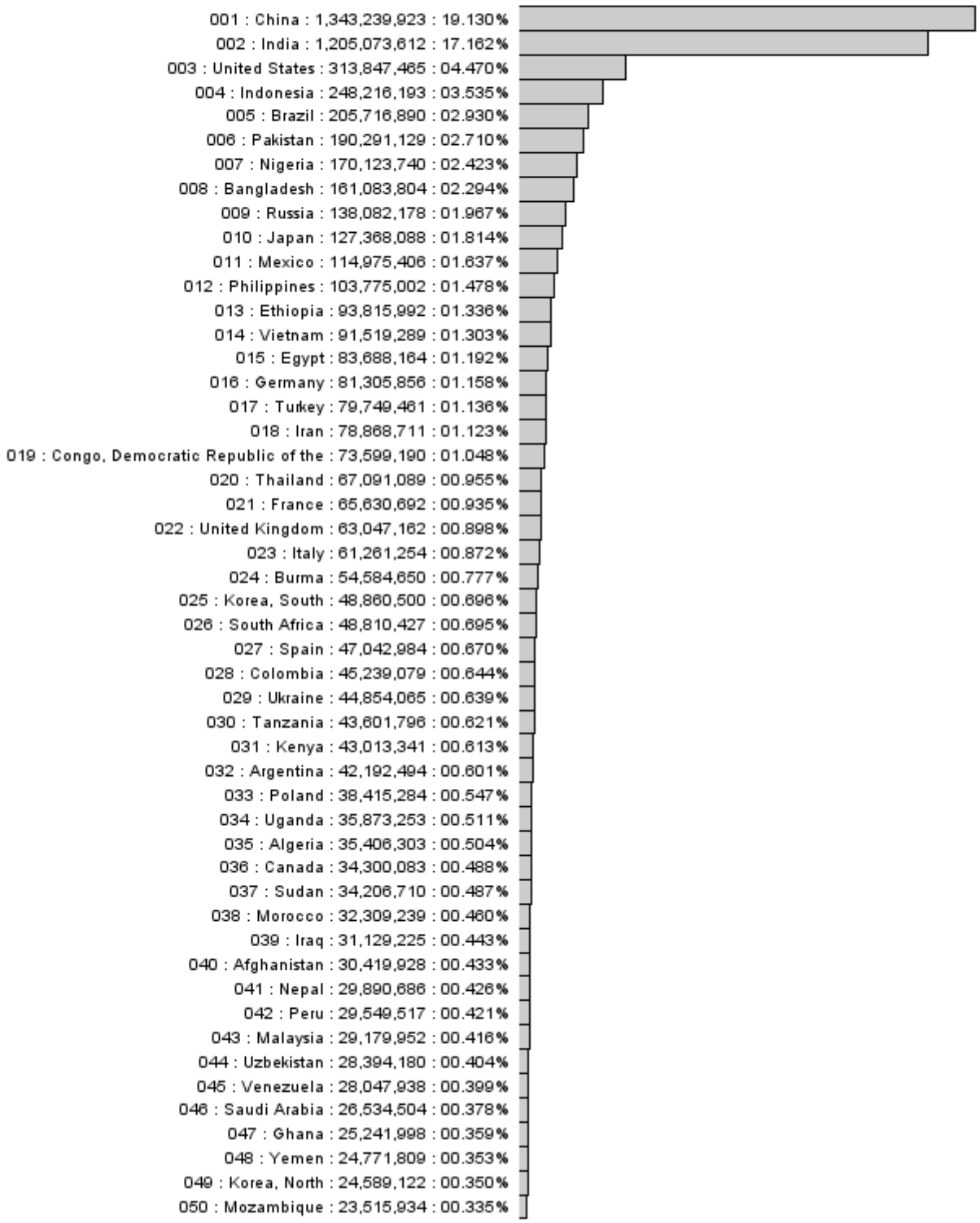
Below is the code to create a graph to fit on a full 8.5-inch X 11-inch page with 1-inch margins in a Microsoft Word document. The source document for this paper has such characteristics.

NOTE: If I had omitted the optional Title4 below,
I could have increased the TextFontPointSize to 8 without causing thinning of bar labels.

```
LIBNAME DataLib "D:\#MWSUG2012 Graphs\Data";

options MPRINT;

%SubsettedRankingHbarChartsToDisk(
data=DataLib.PopulationByCountryPerWFB2012
,TopN=50
,OutputDiskPath=D:\@WIILSU Nov2012\ResultsForManualInsertToMicrosoftOffice
,GraphFileName=ImageToInsertIntoAmicrosoftWordPage
,BorderOnOrOff=ON
,Title1_Prefix=MS Word Hbar Chart of Ranked Population and Percent of World Total For
,Title1_Suffix=Countries
,Title4=Increasing font point size would have caused thinning of labels
,Title4_Color=Red
,ShowRunDayDateTime=YES
,BarLabelVar=Country
,BarMeasureVar=Population
,BarMeasureFormat=comma13.
,TitleFontPointSize=9 /* adjusted to avoid TITLE line wrap
                         (no WARNING in SAS log when they occur) */
,TextFontPointSize=7  /* adjusted to avoid thinning of bar labels
                         (no WARNING in SAS log when it occurs) */
,Xpixels=624   /* image width  to fill 6.5 inches in Microsoft Word document */
,Ypixels=864); /* image height to fill 9   inches in Microsoft Word document */
```

On the next page, the image file was imbedded by using the Word Insert Picture function. I applied a Word border in addition to the chart’s own border. In order to get the bottom border to appear at its full thickness, it was necessary to use the Word Size function to reduce the imbedded image height to 8.99 inches.

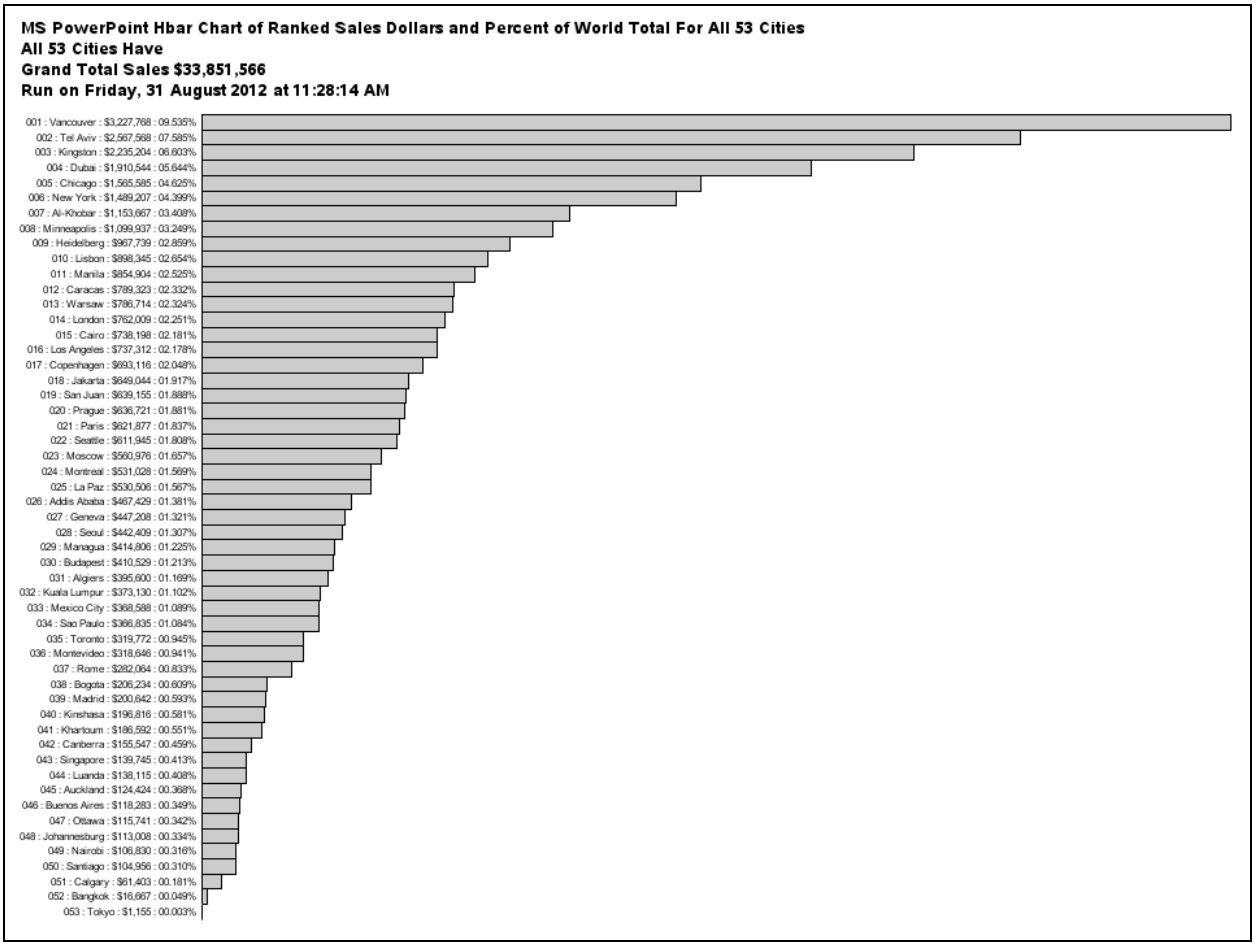
MS Word Hbar Chart of Ranked Population and Percent of World Total For Top 50 Countries
Top 50 Countries Account for SubTotal Population 6,121,300,000 which is 87.177% of Grand Total
All 238 Countries Have Grand Total Population 7,021,700,000
Increasing font point size would have caused thinning of labels
Run on Friday, 31 August 2012 at 11:28:13 AM



Below is the code to create a graph to fit on the full 10-inch by 7.5-inch area of a Microsoft PowerPoint slide. Unlike the prior example for a graph imbedded in a Word document, I did not turn on the optional graph border, which cannot be visible on an image that fills the slide. Also, to save vertical space, this derivative macro does not include the option of Title5.

```
%SubsettedRankingHbarChartsToDisk(
data=sashelp.shoes
,TopN=999999999 /* I wanted a chart of ALL of the bars. */
                /* Any number greater than the total number of bars forces that. */
,OutputDiskPath=D:\@WIILSU Nov2012\ResultsForManualInsertToMicrosoftOffice
,GraphFileName=ImageToInsertIntoAmicrosoftPowerPointSlide
,Title1_Prefix=MS PowerPoint Hbar Chart of Ranked Sales Dollars and Percent of World
Total For
,Title1_Suffix=Cities
,BarLabelVar=Subsidiary
,BarMeasureVar=Sales
,BarMeasureFormat=dollar11.
,TitleFontPointSize=10 /* adjusted to avoid thinning of bar labels or TITLE line wrap
                          (no WARNING in SAS log when they occur) */
,TextFontPointSize=6   /* adjusted to avoid thinning of bar labels
                          (no WARNING in SAS log when it occurs) */
,Xpixels=960   /* image width  to fill 10  inches in Microsoft PowerPoint slide */
,Ypixels=720); /* image height to fill 7.5 inches in Microsoft PowerPoint slide */
```

Below is the image created by running the above code. Its readability here is impaired by the forced shrinkage to fit within the 6.5-inch width of the usable part of this page.



Here is the code for the SubsettedRankingHbarChartsToDisk macro:

```
%macro SubsettedRankingHbarChartsToDisk(
data=
,OutputDiskPath=
,ShowRunDayDateTime=NO /* It appears as Title5 if Title4 is used.
                          Otherwise, as Title4 */
,Title1_Prefix=
,Title1_Suffix=
,Title4= /* Title4 is optional */
,Title4_Color=Black
,TitlesJustify=Left /* Center is the SAS default.
                       Unlike web graphs to HTML,
                       for which titles can be pushed out to the HTML source
                       by using ODS HTML ... NOGTITLE,
                       for graphs to disk
                       the titles must be imbedded in the image file.
                       However, until it is changed,
                       SGPLOT centering of titles for HBAR charts is defective:
                       i.e., titles short enough to fit are centered over the bars,
                       while titles that are too long
                       are centered within the full width of the graph.
                       One would expect center justified titles
                       to always be centered within the full width of the graph.
                       For now, Left Justification is the better choice. */
,GraphFileName=    /* If GraphFileName is not assigned,
                      then the graph title is compressed and used as a default,
                      but that might have adverse consequences. */
,BorderOnOrOff=OFF /* OFF = my preferred choice for PowerPoint;
                       ON = my preferred choice for Word.
                      After manual Insert Picture in Word,
                      I also turn on the Word border */
,TopN=
,BarLabelVar=
,BarMeasureVar=
,BarMeasureFormat=
,BarMeasureDescription= /* Macro does not retrieve SAS var label even if one exists.
                           If BarMeasureDescription not specified,
                           then SAS var name (BarMeasureVar) is used instead. */
,BarColor=CXCCCCCC /* light grey */
,BarWidth=1 /* adjust to less than 1 if needed to reduce barwidth to bar label height
               which is controlled by TextFontPointSize */
,Yoffsetmax=
,TitleFont='Albany AMT/Bold'
,TitleFontPointSize=16 /* adjust to avoid TITLE line wrap,
                          or to reduce vertical space consumption
                          in order to help avoid thinning of bar labels
                          (no WARNING in SAS log when either phenomenon occurs) */
,TextFont='Albany AMT/Bold'
,TextFontPointSize=14 /* adjust to avoid thinning of bar labels
                         (no WARNING in SAS log when it occurs) */
,Xpixels=
,Ypixels= /* adjust to avoid thinning of bar labels
             (no WARNING in SAS log when it occurs) */
);

%macro RemoveCraphAreaFrameFromStyle
(ParentStyle=,StyleWithNoFrame=);

PROC TEMPLATE;
DEFINE STYLE &StyleWithNoFrame;
  PARENT=&ParentStyle;
  CLASS GRAPHWALLS / FRAMEBORDER=OFF; /* remove a useless box around the bars */
```

```
END; RUN;
%mend  RemoveCraphAreaFrameFromStyle;

%RemoveCraphAreaFrameFromStyle
(ParentStyle=Styles.Minimal
,StyleWithNoFrame=Styles.MinimalWithNoFrame);

%if %length(&BarMeasureDescription) EQ 0
%then %let BarMeasureDescription = &BarMeasureVar;

DATA _NULL_;
RunDayDateTimeText = 'Run on ' ||
                     TRIM(LEFT(PUT(DATE(),weekdatx37.))) ||
                     ' at ' ||
                     TRIM(LEFT(PUT(TIME(),timeampm11.)));
CALL SYMPUT('RunDayDateTime',TRIM(LEFT(RunDayDateTimeText)));
RUN;

PROC SUMMARY DATA=&data nway;
CLASS &BarLabelVar;
VAR &BarMeasureVar;
OUTPUT OUT=Summed(drop=_type_ _freq_) SUM=;
RUN;

PROC SQL NOPRINT;
SELECT COUNT(&BarLabelVar) , SUM(&BarMeasureVar)
  INTO :CountOfAllBars     , :BarMeasureGrandTotal
  FROM Summed;
QUIT;

PROC SORT DATA=Summed;
BY DESCENDING &BarMeasureVar;
RUN;

DATA All_WithBarLabels(drop=BarLabelSuffix);
LENGTH BarLabel $ 256 BarLabelSuffix $ 7;
SET Summed;
BarLabelSuffix =
  TRIM(LEFT(PUT(((&BarMeasureVar / &BarMeasureGrandTotal) * 100),6.3))) || '%';
IF LENGTH(BarLabelSuffix) EQ 6
THEN BarLabelSuffix = '0' || BarLabelSuffix;
BarLabel = TRIM(LEFT(PUT(_N_,z3.))) /* a bar count more than three digits
                                       will not fit on a PowerPoint slide or Word page */
           || ' : ' || TRIM(LEFT(&BarLabelVar))
           || ' : ' || TRIM(LEFT(PUT(&BarMeasureVar,&BarMeasureFormat)))
           || ' : ' || BarLabelSuffix;
RUN;

OPTIONS OBS=&TopN;

DATA Selected_WithBarLabels;
SET All_WithBarLabels;
RUN;

OPTIONS OBS=MAX;

DATA ToChart;
SET Selected_WithBarLabels;
RUN;

PROC SQL NOPRINT;
SELECT COUNT(&BarLabelVar)  , SUM(&BarMeasureVar)
  INTO :CountOfSelectedBars , :BarMeasureSubTotal
```

```
  FROM ToChart;
QUIT;

DATA _NULL_;
LENGTH ForSYMPUT ForSYMPUT1 ForSYMPUT2 8;
ForSYMPUT = &BarMeasureGrandTotal;
CALL SYMPUT('GrandTotal'     ,TRIM(LEFT(PUT(ForSYMPUT,&BarMeasureFormat))));
ForSYMPUT = &BarMeasureSubTotal;
CALL SYMPUT('SubTotal'       ,TRIM(LEFT(PUT(ForSYMPUT,&BarMeasureFormat))));
ForSYMPUT = &CountOfAllBars;
CALL SYMPUT('CountOfAll'     ,TRIM(LEFT(PUT(ForSYMPUT,comma32.))));
ForSYMPUT = &CountOfSelectedBars;
CALL SYMPUT('CountOfSelected',TRIM(LEFT(PUT(ForSYMPUT,3.))));
     /* Word page or PowerPoint slide will never fit more than a 3-digit bar count */
ForSYMPUT1 = &BarMeasureSubTotal;
ForSYMPUT2 = &BarMeasureGrandTotal;
CALL SYMPUT('SubTotalPercentOfGrandTotal',
  TRIM(LEFT(PUT(( ForSYMPUT1 / ForSYMPUT2 ) * 100 , 6.3))));
RUN;

%if %eval(&TopN GE &CountOfAllBars)
%then %do;
  %let Title1 = &Title1_Prefix.%str( All )&CountOfSelected &Title1_Suffix;
  %let Title2 = All &CountOfSelected &Title1_Suffix Have;
  %let Title3 = Grand Total &BarMeasureDescription &GrandTotal;
%end;
%else %do;
  %let Title1 = &Title1_Prefix.%str( Top )&CountOfSelected &Title1_Suffix;
  %let Title2 = Top &CountOfSelected &Title1_Suffix
    Account for SubTotal &BarMeasureDescription &SubTotal
    which is &SubTotalPercentOfGrandTotal% of Grand Total;
  %let Title2 = %sysfunc(compbl(%nrquote(&Title2)));
  %let Title3 =
    All &CountOfAll &Title1_Suffix Have Grand Total &BarMeasureDescription
&GrandTotal;
%end;

%if %length(&GraphFileName) EQ 0
%then %let GraphFileName = %sysfunc(COMPRESS(&Title1,' '));

ODS GRAPHICS ON / RESET=ALL BORDER=&BorderOnOrOff SCALE=OFF
    HEIGHT=&Ypixels.px WIDTH=&Xpixels.px
    IMAGENAME="&GraphFileName";

FOOTNOTE;

ODS NORESULTS;
ODS LISTING GPATH="&OutputDiskPath" STYLE=Styles.MinimalWithNoFrame;

TITLE1 JUSTIFY=&TitlesJustify FONT=&TitleFont HEIGHT=&TitleFontPointSize PT "&Title1";
TITLE2 JUSTIFY=&TitlesJustify FONT=&TitleFont HEIGHT=&TitleFontPointSize PT "&Title2";
TITLE3 JUSTIFY=&TitlesJustify FONT=&TitleFont HEIGHT=&TitleFontPointSize PT "&Title3";
%if %length(&Title4) NE 0 %then %do;
TITLE4 JUSTIFY=&TitlesJustify FONT=&TitleFont HEIGHT=&TitleFontPointSize PT
                                                COLOR=&Title4_Color "&Title4";
%end;
%if %upcase(&ShowRunDayDateTime) NE 0 %then %do;
  %if %length(&Title4) EQ 0
  %then %do;
TITLE4
  %end;
  %else %do;
TITLE5
```

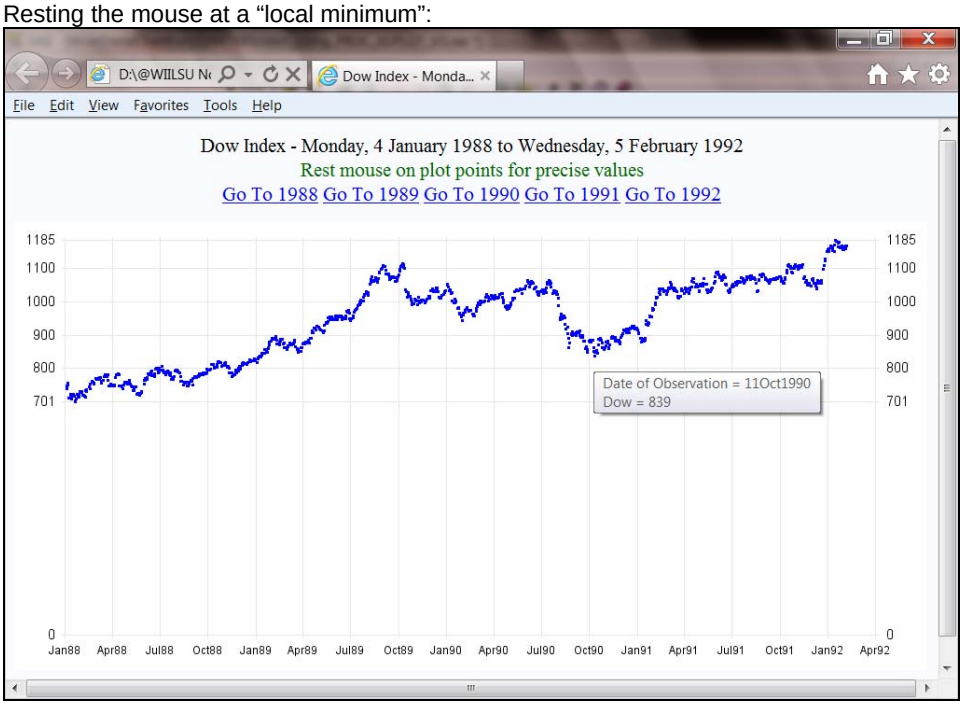
```
  %end;
       JUSTIFY=&TitlesJustify FONT=&TitleFont HEIGHT=&TitleFontPointSize PT
"&RunDayDateTime";
%end;

PROC SGPLOT DATA=ToChart;
HBAR BarLabel / RESPONSE=&BarMeasureVar
  CATEGORYORDER=RESPDESC BARWIDTH=&BarWidth FILL FILLATTRS=(COLOR=&BarColor) OUTLINE;
YAXIS DISPLAY=(NOLABEL NOLINE NOTICKS) VALUEATTRS=(SIZE=&TextFontPointSize)
%if %length(&Yoffsetmax) NE 0 %then %do;
      OFFSETMAX=&Yoffsetmax
%end;
      ;
XAXIS DISPLAY=NONE;
RUN;

ODS LISTING Style=Styles.Listing;

%mend  SubsettedRankingHbarChartsToDisk;
```

NOTE: The macro above is a bit simpler than the prior one that produces two interlinked web graphs. Nevertheless, it, too, might look rather daunting. If you like what it does As Is, just file it in a macro library, use **OPTION SASAUTOS=** to point your SAS program at the macro library, and invoke the macro in a manner shown in the sample invocations. Of course, you also can modify the macro if you wish.

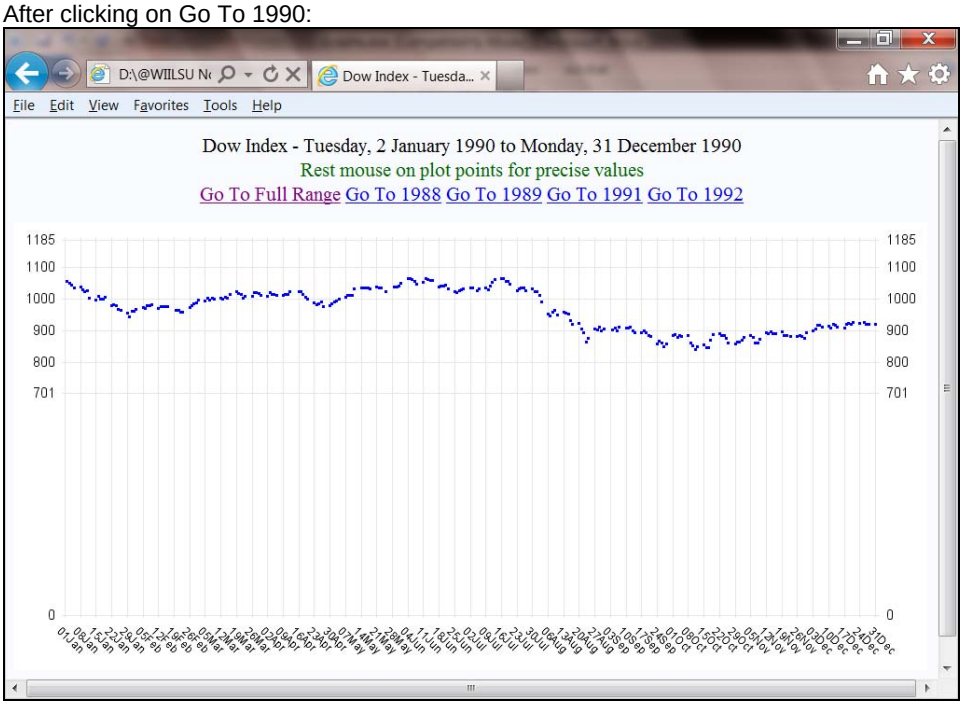
VERY LONG TREND CHARTS, Part 1: ODS GRAPHICS PROC SGPLOT Solution

One would like to see an overview of the total time period. However, since showing that time period requires a comparatively sparse set of horizontal axis tick mark values, it is helpful to be able to zoom in on sub-periods of the total period. Below is a solution that I like. Web deployment for trend data, with the availability of ALT text, is always preferable to static delivery. It not only supports navigation between different views, but also could be enhanced with the option to link to a spreadsheet of the data as demonstrated in an earlier section of this paper.

Resting the mouse at a “local minimum”:



After clicking on Go To 1990:



The graphs above start the y axis at zero to avoid magnifying minor fluctuations, and maintain the same y tick mark values and reference lines across all views for easy comparability. When moving the cursor around the web page, but not resting it on a plot point, there is displayed a default description of the graph provided by ODS Graphics. Unlike SAS/GRAPH, there is no way to customize it, and no way to suppress it.

The title lines for the graphs above must be delivered outside the image file (using ODS HTML . . . NOGTITLE) so that the hyperlinks will work. (This is opposite the situation for SAS/GRAPH PROC GPLOT.) With the ODS Graphics SG procedures there is no direct control of the GRID (reference lines). Unlike the case with the SAS/GRAPH GPLOT procedure, where reference lines can be specified at selected tick mark values OR turned on for all major tick marks (this latter alternative being analogous to GRID in ODS Graphics) and where line type and line color can be controlled, the ODS Graphics grid characteristics are controlled by whatever ODS Style has been selected. Here the HTMLBLUE style was used. If you look closely, you can see that the graph image file is surrounded by a light blue background, and above the image file it is overlaid with title lines (due to the NOGTITLE option mentioned above).

With the ODS Graphics PROC SPLOT SERIES statement the IMAGEMAP (HTML source code) that defines the hot spots for the cursor to display the ALT text is created in a defective manner. The boundary of each hot spot is defined three times for the marker being drawn for the left-hand vertical axis, and each of the three boundary definitions is different, although they are not extremely different. Furthermore, in order to get the tick mark values displayed on the right-hand vertical axis the markers must be drawn a second time, and the imagemap contains another three boundary definitions for each hot spot. The second set of three different boundary definitions DOES match the first set. The size of the HTML file for the full date range graph is 1058 KB, and for the one-year date range graphs varies, but is approximately 283 KB. The IMAGEMAP drawn for SAS/GRAPH PROC GPLOT is efficient and much smaller. Each hot spot is defined only once, even when both vertical axes have tick mark values. The HTML files created with SAS/GRAPH PROC GPLOT are 95 KB and 26 KB—i.e., less than one-tenth the size of those created with ODS Graphics PROC SGPLOT.

Here is the code used to create the above graphs with ODS Graphics and PROC SGPLOT:

```
%let Path = D:\@WIILSU Nov2012\Results\WidePlots\Using_PROC_SGPLOT;

 /* START of SetUp Code not unique to use of ODS Graphics */

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

DATA WORK.DowByDay;
LABEL snydjcm='Dow';
LENGTH Year $4;
SET CITIHELP.citiday(KEEP=date snydjcm WHERE=(snydjcm NE .));
snydjcm = ROUND(snydjcm,1);
Year = YEAR(date);
RUN;

PROC MEANS DATA=WORK.DowByDay MIN MAX NOPRINT;
VAR snydjcm date;
OUTPUT OUT=WORK.MinMax MIN=MinY StartDate MAX=MaxY EndDate;
RUN;

DATA _NULL_;
SET MinMax;
CALL SYMPUT('MinY',TRIM(LEFT(MinY)));
CALL SYMPUT('MaxY',TRIM(LEFT(MaxY)));
CALL SYMPUT('DateRange',
     TRIM(LEFT(PUT(StartDate,WEEKDATX32.))) ||
     ' to ' ||
     TRIM(LEFT(PUT(EndDate, WEEKDATX32.))));
RUN;

PROC SUMMARY DATA=WORK.DowByDay NWAY;
CLASS Year;
VAR date;
OUTPUT OUT=WORK.EachYearWithStartDateEndDate MIN=StartDate MAX=EndDate;
RUN;
```

```
DATA _NULL_;
SET WORK.EachYearWithStartDateEndDate END=LastYear;
CALL SYMPUT('DateRange'||TRIM(LEFT(_N_)),
     TRIM(LEFT(PUT(StartDate, WEEKDATX32.)))
     || ' to ' ||
     TRIM(LEFT(PUT(EndDate, WEEKDATX32.))));
CALL SYMPUT('Year'||TRIM(LEFT(_N_)),TRIM(LEFT(Year)));
IF LastYear;
CALL SYMPUT('NumberOfYears',TRIM(LEFT(_N_)));
RUN;

DATA _NULL_;
SET WORK.EachYearWithStartDateEndDate END=LastYear;
* Want to start on a Monday not later than the first day of data *;
StartDate = MDY(1,1,Year);
DayOfWeekStart = WEEKDAY(StartDate);
IF DayOfWeekStart GT 2
THEN StartDate = StartDate - (DayOfWeekStart - 2);
ELSE
IF DayOfWeekStart EQ 1
THEN StartDate = StartDate - 6;
CALL SYMPUT('StartDateTick'||TRIM(LEFT(_N_)),StartDate);
* Want to end on a Monday not earlier than the last day of data *;
EndDate = MDY(12,31,Year);
DayOfWeekEnd = WEEKDAY(EndDate);
IF DayOfWeekEnd GT 2
THEN EndDate = EndDate + (9 - DayOfWeekEnd);
ELSE
IF DayOfWeekEnd EQ 1
THEN EndDate = EndDate + 1;
CALL SYMPUT('EndDateTick'||TRIM(LEFT(_N_)),EndDate);
RUN;

 /* End of SetUp Code not unique to use of ODS Graphics */

%MACRO CreateYearLinks_SGPLOT;

%DO j = 1 %TO &NumberOfYears %BY 1;
  LINK="Year_&&Year&j...html" "Go To &&Year&j"
%END;

%MEND CreateYearLinks_SGPLOT;

PROC TEMPLATE;
DEFINE STYLE styles.htmlblueWithNoFrame; /* remove useless box around the plot area */
  PARENT=Styles.htmlblue;
  CLASS graphwalls / frameborder=off;
END; RUN;

ODS NORESULTS; /* do not open in SAS session */
ODS LISTING CLOSE;

  /* START SetUp that applies to all graphs */

GOPTIONS RESET=ALL;
ODS GRAPHICS ON / RESET=ALL BORDER=OFF IMAGEMAP TIPMAX=2500 ANTIALIASMAX=2500
    WIDTH=980px HEIGHT=480px; /* override default size 480px X 360px */

  /* END SetUp that applies to all graphs */

  /* HEIGHT for TITLEs rendered with ODS NOGTITLE cannot use PCT as the Unit */
TITLE1 FONT='Times New Roman' HEIGHT=16pt COLOR=black
       "Dow Index - &DateRange" ;
```

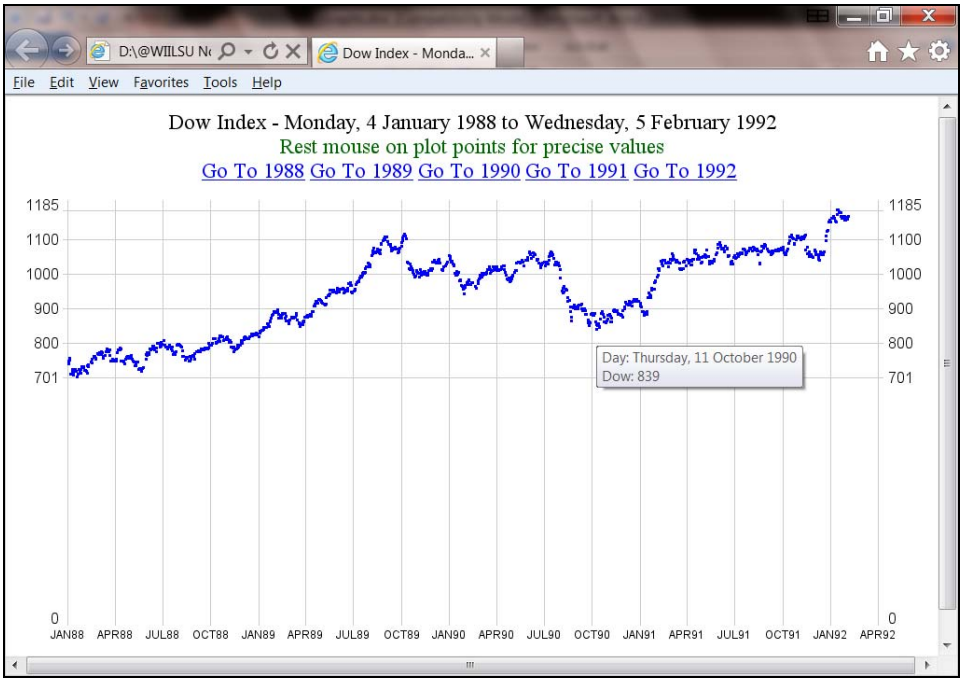
```
TITLE2 FONT='Times New Roman' HEIGHT=16pt COLOR=CX006600
       "Rest mouse on plot points for precise values";
TITLE3 FONT='Times New Roman' HEIGHT=16pt
       %CreateYearLinks_SGPLOT
       ;

ODS HTML PATH="&Path"  (URL=NONE) STYLE=Styles.htmlblueWithNoFrame NOGTITLE GFOOTNOTE
    BODY="FullRange_QuarterlyHticks.html" (TITLE="Dow Index - &DateRange");

ODS GRAPHICS ON / MAXLEGENDAREA=0 /* suppress legend caused by extra Y2AXIS */
                  IMAGENAME="atZERO";

PROC SGPLOT DATA=WORK.DowByDay;
SERIES Y=snydjcm X=date /
       MARKERS MARKERATTRS=(SIZE=3 SYMBOL=circlefilled COLOR=blue)
       LINEATTRS=(COLOR=white);
SERIES Y=snydjcm X=date / Y2AXIS /* create a right-hand y axis */
       MARKERS MARKERATTRS=(SIZE=3 SYMBOL=circlefilled COLOR=blue)
       LINEATTRS=(COLOR=white);
YAXIS DISPLAY=(NOLABEL NOLINE NOTICKS) GRID
      VALUEATTRS=(SIZE=2PCT FAMILY='Arial')
      MIN=0 MAX=&MaxY VALUESHINT VALUES=(0 &MinY 800 900 1000 1100 &MaxY);
Y2AXIS /* define the right-hand y axis */
      DISPLAY=(NOLABEL NOLINE NOTICKS) GRID
      VALUEATTRS=(SIZE=2PCT FAMILY='Arial')
      MIN=0 MAX=&MaxY VALUESHINT VALUES=(0 &MinY 800 900 1000 1100 &MaxY);
XAXIS INTERVAL=quarter FITPOLICY=ROTATE
      DISPLAY=(NOLABEL NOLINE NOTICKS) GRID
      VALUEATTRS=(SIZE=1.5PCT FAMILY='Arial') TICKVALUEFORMAT=MONYY5.;
RUN;

ODS HTML CLOSE;

%MACRO LinkedYearlyTrendCharts_SGPLOT;

%DO i = 1 %TO &NumberOfYears %BY 1;

  /* HEIGHT for TITLEs rendered with ODS NOGTITLE cannot use PCT as the Unit */
TITLE1 FONT='Times New Roman' HEIGHT=16pt COLOR=black "Dow Index - &&DateRange&i" ;
TITLE2 FONT='Times New Roman' HEIGHT=16pt COLOR=CX006600 "Rest mouse on plot points
for precise values";
TITLE3 FONT='Times New Roman' HEIGHT=16pt
  LINK="FullRange_QuarterlyHticks.html" "Go To Full Range"
  %DO j = 1 %TO &NumberOfYears %BY 1;
    %IF &&Year&j NE &&Year&i
    %THEN %DO;
  LINK="Year_&&Year&j...html" "Go To &&Year&j"
    %END;
  %END;
  ;

ODS HTML PATH="&Path"  (URL=NONE) STYLE=Styles.htmlblueWithNoFrame NOGTITLE GFOOTNOTE
    BODY="Year_&&Year&i...html" (TITLE="Dow Index - &&DateRange&i");

ODS GRAPHICS ON / IMAGENAME="atZERO&i" ;

PROC SGPLOT DATA=WORK.DowByDay(WHERE=(Year EQ "&&Year&i" ));
SERIES Y=snydjcm X=date /
       LINEATTRS=(COLOR=white)
       MARKERS MARKERATTRS=(SIZE=3 SYMBOL=circlefilled COLOR=blue);
    /* The default unit for SIZE is px (pixels). At a size of only three pixels
       the symbol which should be a circular can be drawn only as a square.
       Because there are so many plot points, if they are to be distinguishable,
```

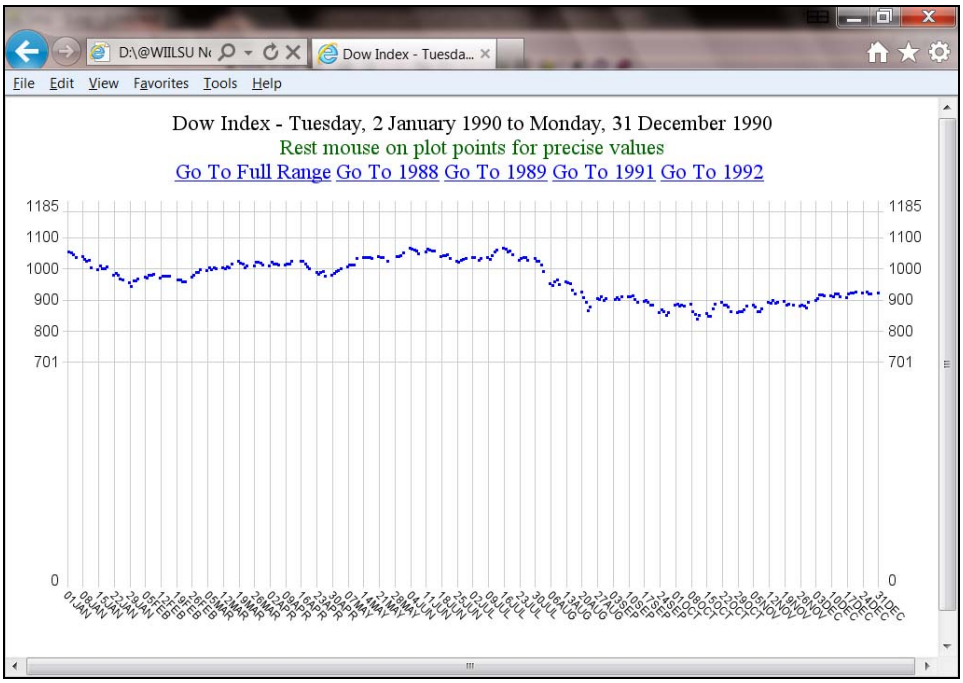
```
       it is necessary to render them at a small size. */
SERIES Y=snydjcm X=date / Y2AXIS /* create a right-hand y axis */
       LINEATTRS=(COLOR=white) /* there is no way to suppress the unneeded line */
       MARKERS MARKERATTRS=(SIZE=3 SYMBOL=circlefilled COLOR=blue);
      /* Using the NOMARKERS option to suppress the second drawing of the markers,
         and thus use Y2AXIS only to get the tick mark labels at the right-hand side
         would have the undesirable effect of having the white plot line for the
         right-hand y axis drawn over the blue markers that are first written over
the
         white plot line for the left-hand y axis. */
YAXIS DISPLAY=(NOLABEL NOLINE NOTICKS) GRID
      VALUEATTRS=(SIZE=2PCT FAMILY='Arial')
      MIN=0 MAX=&MaxY VALUESHINT VALUES=(0 &MinY 800 900 1000 1100 &MaxY);
Y2AXIS /* define the right-hand y axis */
      DISPLAY=(NOLABEL NOLINE NOTICKS) GRID
      VALUEATTRS=(SIZE=2PCT FAMILY='Arial')
      MIN=0 MAX=&MaxY VALUESHINT VALUES=(0 &MinY 800 900 1000 1100 &MaxY);
XAXIS DISPLAY=(NOLABEL NOLINE NOTICKS) GRID
      VALUEATTRS=(SIZE=1.5PCT FAMILY='Arial') TICKVALUEFORMAT=DATE5.
      VALUES=(&&StartDateTick&i to &&EndDateTick&i by 7)
      FITPOLICY=ROTATE;
RUN;
ODS HTML CLOSE;

%END;

%MEND LinkedYearlyTrendCharts_SGPLOT;

OPTIONS MPRINT;

%LinkedYearlyTrendCharts_SGPLOT;

ODS LISTING;
```

The entire concatenation of lengthy code above COULD Be converted to a macro. The setup code is specific to the input data set, but the mechanics of supporting a multi-year time span interlinked with yearly subset time spans could be macro-packaged. For now, it was enough work to get this far.

VERY LONG TREND CHARTS, Part 2: SAS/GRAPH PROC GPLOT Solution

Here is the same pair of plots created with SAS/GRAPH PROC GPLOT. **Note the custom ALT text.**





Because the links above are in a title line that overlays the image file (due to ODS HTML . . . GTITLE), they do not change color after having been visited.

With the SAS/GRAPH PROC GPLOT the IMAGEMAP (HTML source code) that defines the hot spots for the cursor to display the ALT text is created very efficiently. The sizes of the HTML file for the full date range graph is 95 KB and for the one-year date range graphs is 26 KB. The sizes of the corresponding HTML files created with ODS Graphics PROC SGPLOT are 1058 KB and 283 KB—i.e., over ten times the size of those created with SAS/GRAPH GPLOT.

There is setup code for the above graphs that is the same as that used for the ODS Graphics PROC SGPLOT solution. It is clearly labeled there. Below is the unique code used to create the above graphs with SAS/GRAPH and PROC GPLOT.

NOTE: When running the code below, the steps that create the one-year-at-a-time plots generate a misleading WARNING message that says (wrongly): **The bottom horizontal axis labeled Date of Observation could not be fit as specified. The axis values will overwrite.**

```
%let Path = D:\@WIILSU Nov2012\Results\WidePlots\Using_PROC_GPLOT;

 /* Same SetUp Code as used for ODS Graphics would be here */

%MACRO CreateYearLinks_GPLOT;
%DO j = 1 %TO &NumberOfYears %BY 1;
  LINK="Year_&&Year&j...html" UNDERLINE=1 "Go To &&Year&j" UNDERLINE=0 " "
%END;
%MEND  CreateYearLinks_GPLOT;

DATA WORK.DowByDay;
LENGTH HTMLvar $ 512;
SET  WORK.DowByDay;
HTMLvar = "alt='"
  || "Day: " || TRIM(LEFT(PUT(Date, WEEKDATX32.)))
  || '0D'X /* force a line break in the ALT text display */
  || "Dow: " || TRIM(LEFT(PUT(snydjcm,4.))) || "'";
RUN;

PROC CATALOG CAT=WORK.gseg KILL; RUN; QUIT;

ODS NORESULTS; /* do not open in SAS session */
ODS LISTING CLOSE;

 /* START SetUp that applies to all graphs */

ODS GRAPHICS OFF;
GOPTIONS RESET=ALL;
GOPTIONS FTEXT="Arial"; /* for axis values */
GOPTIONS XPIXELS=980 YPIXELS=576;

SYMBOL1 COLOR=blue HEIGHT=0.5 FONT=ZAPF VALUE='E2'X;
SYMBOL2 COLOR=blue HEIGHT=0.5 FONT=ZAPF VALUE='E2'X;
                    /* Use of FONT=ZAPF VALUE='E2'X results in a smaller HTML file
  because many other SAS/GRAPH symbol choices cause the AREA map, used to provide
  ALT text, to be generated in a very inefficient manner. When multiple browse
  windows are open, not only those browser windows, but also other windows that
  the viewing user might have open, can become inoperative. That problem of huge
  area maps might have been observed only with very large numbers of plot points
  on multi-line plots. Multiple lines multiply the numbers of plots points.
  Instead of FONT=ZAPF VALUE='E2'X, an unwise symbol choice for a very dense plot
  would be, e.g., VALUE=DOT. */

AXIS1 LABEL=NONE MAJOR=NONE MINOR=NONE STYLE=0 /*  left-hand y axis */
      ORDER = 0 TO 1200 BY 100
      VALUE=(HEIGHT=3PCT
      '0' '' '' '' '' '' '' "&MinY" '800' '900' '1000' '1100' "&MaxY");

AXIS2 LABEL=NONE MAJOR=NONE MINOR=NONE STYLE=0 /* right-hand y axis */
      ORDER = 0 TO 1200 BY 100
      VALUE=( HEIGHT=3PCT JUSTIFY=LEFT
      '0' '' '' '' '' '' '' "&MinY" '800' '900' '1000' '1100' "&MaxY");

 /* END SetUp that applies to all graphs */
```

```
AXIS3 LABEL=NONE MAJOR=NONE MINOR=NONE STYLE=0 VALUE=(HEIGHT=2PCT);

TITLE1 FONT='Times New Roman' HEIGHT=4PCT COLOR=black
       "Dow Index - &DateRange";
TITLE2 FONT='Times New Roman' HEIGHT=4PCT COLOR=CX006600
       "Rest mouse on plot points for precise values";
TITLE3 HEIGHT=4PCT FONT='Times New Roman'
       COLOR=blue %CreateYearLinks_GPLOT
       ;

ODS HTML PATH="&Path" (URL=NONE) STYLE=Styles.Minimal GTITLE GFOOTNOTE
    BODY="FullRange_DefaultHticks.html" (TITLE="Dow Index - &DateRange");

PROC GPLOT DATA=WORK.DowByDay;
PLOT snydjcm * date / HTML=HTMLvar NAME="atZERO" DESCRIPTION=' '
     VAXIS=AXIS1 CVREF=CXCCCCCC LVREF=1 VREF=(&MinY 800 900 1000 1100 &MaxY)
     HAXIS=AXIS3 AUTOHREF CHREF=CXCCCCCC LHREF=1;
PLOT2 snydjcm*date / VAXIS=AXIS2;
FORMAT date MONYY5.;
RUN; QUIT;

ODS HTML CLOSE;

%MACRO LinkedYearlyTrendCharts_GPLOT;

%DO i = 1 %TO &NumberOfYears %BY 1;

AXIS3 LABEL=NONE MAJOR=NONE MINOR=NONE STYLE=0 VALUE=(HEIGHT=2PCT ANGLE=-45)
      ORDER = &&StartDateTick&i to &&EndDateTick&i by 7;
TITLE1 FONT='Times New Roman' HEIGHT=4PCT COLOR=black
       "Dow Index - &&DateRange&i";
TITLE2 FONT='Times New Roman' HEIGHT=4PCT COLOR=CX006600
       "Rest mouse on plot points for precise values";
TITLE3 HEIGHT=4PCT FONT='Times New Roman' COLOR=blue
  LINK="FullRange_DefaultHticks.html" UNDERLINE=1 "Go To Full Range" UNDERLINE=0 " "
  %DO j = 1 %TO &NumberOfYears %BY 1;
    %IF &&Year&j NE &&Year&i
    %THEN %DO;
  LINK="Year_&&Year&j...html" UNDERLINE=1 "Go To &&Year&j" UNDERLINE=0 " "
    %END;
  %END;
  ;

ODS HTML PATH="&Path" (URL=NONE) STYLE=Minimal GTITLE GFOOTNOTE
    BODY="Year_&&Year&i...html" (TITLE="Dow Index - &&DateRange&i");
PROC GPLOT DATA=WORK.DowByDay(WHERE=(Year EQ "&&Year&i"));
PLOT snydjcm * date / HTML=HTMLvar NAME="atZERO&i" DESCRIPTION=' '
     VAXIS=AXIS1 CVREF=CXCCCCCC LVREF=1 VREF=(&MinY 800 900 1000 1100 &MaxY)
     HAXIS=AXIS3 AUTOHREF CHREF=CXCCCCCC LHREF=1;
PLOT2 snydjcm * date / VAXIS=AXIS2;
FORMAT date DATE5.;
RUN; QUIT;
ODS HTML CLOSE;

%END;

%MEND LinkedYearlyTrendCharts_GPLOT;

OPTIONS MPRINT;

%LinkedYearlyTrendCharts_GPLOT;

ODS LISTING;
```

CONCLUSION

There are many, many useful features in SAS/GRAPH, which has been progressively improved over 33 years, and some of those features are regrettably missing in the new technology of ODS Graphics and SG procedures. The first macro provided here simplifies the task of overcoming a current limitation of the new technology. The third and fourth macros, to create subsetted ranked horizontal bar charts, implement a powerful and useful, though simple, concept and graphic method, and these macros work better when built with the new technology.

The remaining macros, provided both for ODS Graphics and SAS/GRAPH, are not general-purpose tools, but show you how to deliver both an overview and by-segment views of long trends, with links between all of the views, and with ALT text so that the viewer can get at the precise y value and exact date for any plot point, rather than try to guess. Their code could be enhanced to provide links to spreadsheets of the plotted data from and back to any of the web views.

REFERENCES

1. 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
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. “Communication-Effective Use of Color for Web Pages, Graphs, Tables, Maps, Text, and Print”, *Proceedings of the Twenty-Ninth Annual SAS Users Group International Conference*, Cary, NC, USA: SAS Institute Inc., 2004. http://www2.sas.com/proceedings/sugi29/176-29.pdf
4. Bessler, LeRoy. “Communication-Effective Pie Charts”, *Proceedings of the SAS Global Forum 2007 Conference*. Cary, NC: SAS Institute Inc. 2007. http://www2.sas.com/proceedings/forum2007/134-2007.pdf
5. 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
6. 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
7. 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
8. Bessler, LeRoy. “%TRENDANO Macro for Collision-Free Automatic Annotation of Trend Plots”, *Wisconsin Illinois SAS Users Conference Proceedings*. Mequon, WI: Software User Services. 2001. For the latest report on this work, see http://www2.sas.com/proceedings/sugi27/p093-27.pdf for which Dr. Francesca Pierri was co-author.
9. Holland, Philip R. “Why Should You Be Using the New SG (Statistical Graphics) Procedures in SAS 9.2?”, *Proceedings of the SAS Global Forum 2011 Conference*. Cary, NC: SAS Institute Inc. 2011. http://support.sas.com/resources/papers/proceedings11/427-2011.pdf

AUTHOR INFORMATION

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

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

Dr. LeRoy Bessler has presented at software user conferences in the US, Canada, and Europe, on topics such as effective visual communication (using graphs, tables, web pages, or color), highly formatted Excel reporting from SAS, custom-developed tools to assist SAS server administrators, users, and managers, and Software-Intelligent Application Development methods to maximize Reliability, Reusability, Maintainability, Extendability, and Flexibility. His SAS experience includes application development and supporting users, servers, software, and data.

APPENDIX 1. Set-Up Code for SGPLOT Time Series Graphs Using CITIDAY Data Set

```
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";

 /* There are five cyclic data sets of financial and economic data,
    citiday, citiwk, citimon, citiqtr, and citiyr,
    that are shipped when your site licenses SAS/ETS.
    If you do not have SAS/ETS, SAS Technical Support can tell you
    how to download them. */

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;
```

APPENDIX 2. Set-Up Code for SGPLOT Time Series Graphs Using CITIMON Data Set

```
 /* Style-related code below has nothing to do with use of CITIMON data set per se */

%macro RemoveCraphAreaFrameFromStyle
(ParentStyle=,StyleWithNoFrame=);
PROC TEMPLATE;
DEFINE STYLE &StyleWithNoFrame;
  PARENT=&ParentStyle;
  CLASS GRAPHWALLS / FRAMEBORDER=OFF;
  /* remove a useless box around the bars */
END; RUN;
%mend  RemoveCraphAreaFrameFromStyle;

%RemoveCraphAreaFrameFromStyle
(ParentStyle=Styles.htmlblue
,StyleWithNoFrame=Styles.htmlblueWithNoFrame);

 /* Style-related code above has nothing to do with use of CITIMON data set per se */

proc format library=work;
  value MonthAbbrev
  1='Jan' 2='Feb' 3='Mar'  4='Apr'  5='May'  6='Jun'
  7='Jul' 8='Aug' 9='Sep' 10='Oct' 11='Nov' 12='Dec';
run; quit;

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

 /* There are five cyclic data sets of financial and economic data,
    citiday, citiwk, citimon, citiqtr, and citiyr,
    that are shipped when your site licenses SAS/ETS.
    If you do not have SAS/ETS, SAS Technical Support can tell you
    how to download them. */

data work.SandPindexByMon1988to1991;
keep Year Month SandPindex date;
format SandPindex 5.;
set CITIHELP.citimon(keep=date FSPCOM where=(FSPCOM ne .));
Year = year(date);
if 1988 LE Year LE 1991;
Month = month(date);
SandPindex=round(FSPCOM,1);
run;
```