

Communication-Effective Data Visualization: Design Principles, Widely Usable Graphic Examples, and Code for Visual Data Insights

LeRoy Bessler, Bessler Consulting and Research

ABSTRACT

Graphic software users are confronted with what I call Options Over-Choice, and with defaults that are designed to easily give you a result, but not necessarily the best result. This presentation and paper focus on guidelines for communication-effective data visualization. It demonstrates their practical implementation, using graphic examples likely to be adaptable to your own work. Code is provided for the examples. Audience members will receive the complete latest update of my tip sheet compendium of graphic design principles. The examples use SAS® tools (traditional SAS/GRAPH® or the newer ODS graphics procedures that are available with Base SAS®), but the design principles are really software independent. Come learn how to use data visualization to inform and influence, to reveal and persuade, using tips and techniques developed and refined over 34 years of working to get the best out of SAS® graphic software tools.

INTRODUCTION

The design principles recommended are software independent, but this paper uses both SAS/GRAPH and the ODS Statistical Graphics procedures for its examples. It will focus on data visualization applications for management reporting and presentations. In that context, typical graphic tasks are to: compare absolute magnitudes; compare shares of the whole; compare percent change; and monitor measurements of interest over time. Some examples were created for the first time with SAS Version 9.4, but were verified, in some cases with Version 9.3, with significant version-dependent functional differences noted. Some examples are, or are derived from, prior Version 9.3 work.

NOTE: To limit the length of this paper, code for Sections B to L (packaged as Appendices B to L) is available from the author via email, either as a PDF, or as a zip file of SAS programs. Code for Section M is in Reference 1.

COMMUNICATION-EFFECTIVE DATA VISUALIZATION: SOME OF THE KEYS

The stand-up presentation might amplify on some of the items below. Many need no explanation or justification.

1. NO 3D pie charts ever; "Safe" 3D horizontal bar charts, as designed here, are communication-effective.
2. Deliver both image and precise numbers:
Image = easy, quick inference and interpretation
Precise Numbers = reliable inference and interpretation
3. Show Them What's Important with ranking and/or subsetting.
4. Provide a headline for your graph whenever possible. State the conclusion, rather than force viewers to guess.
5. Assure text readability. This obvious natural expectation is NOT always achieved.
6. Use horizontal bar charts. Bar labels are always readable, and the length limit is huge (256 characters).
7. Whenever possible, size your web graphs so they will not require scrolling.
8. Facilitate seasonality discernment for time series.
9. Provide "data tips" and a web-linked spreadsheet for dense time series.
10. Use annotation, if feasible, or an on-chart table, for time series that are not web-enabled.
11. Use color to communicate, not to decorate.
12. Prevent ODS Graphics thinning of (i.e., omission of some) categorical values (e.g., of labels in bar charts).
13. Provide a usable legend, if a legend is needed. Some pre-Version 9.4 ODS Graphics legends are unusable.

The above are a subset of what I modestly like to call:

"Principia Graphika: Bessler's Principles for Communication-Effective Graphic Design",
subtitled "A Tip Sheet for Users of SAS Graphics Tools or Any Other Data Visualization Software".
Audience members at the stand-up presentation will receive the complete latest update.

The focus in much of what follows for time series plots or trend line charts is on Guidelines 2, 7, 8, 9, and 10. The focus of later topics on bar charts and pie charts is on Guidelines 1, 2, 3, 6, 12, and 13. Along the way, I am doing my best to adhere to Guidelines 5 and 11. Almost of all of these tutorial graphs are such that Guideline 4 is not relevant. In many cases, multiple title lines describe characteristics of the graph, with just one title line to describe what is being graphed.

DELIVER BOTH IMAGE AND PRECISE NUMBERS

Images facilitate inference and decisions, but precise numbers are required for a correct inference or decision.

For horizontal bar charts, as will be shown, it is always possible to provide precise values for the bar length, rather than requiring the graph viewer to run the eye from bar end to the horizontal axis. It is best to omit that useless axis.

For pie charts, it is always possible deliver precise numbers, but sometimes requires an extra effort (i.e., rather than relying on direct slice labels, which might not fit or might overlap each other, instead a fully informative legend with slice name, value, and percent of whole is provided).

It is trend lines, time series plots, and scatter plots that pose a challenge.

You will see how to best deal with a non-web-enabled trend line or time series plot. For a single line that is not very dense in plot points, you can hard annotate each plot point. For a multi-line plot that is not too dense, you can create a table of Y values stacked above each X-axis tick mark value, with a line identifier at the left of each row in the table.

But when a plot line becomes very dense, the only choice is to deliver it web-enabled so that resting the mouse on any plot point opens a pop-up box, which is called (in this paper and in ODS Graphics) a "data tip". This brings up a new problem due to a change added in the Internet Explorer 11, which is discussed in Appendix A.

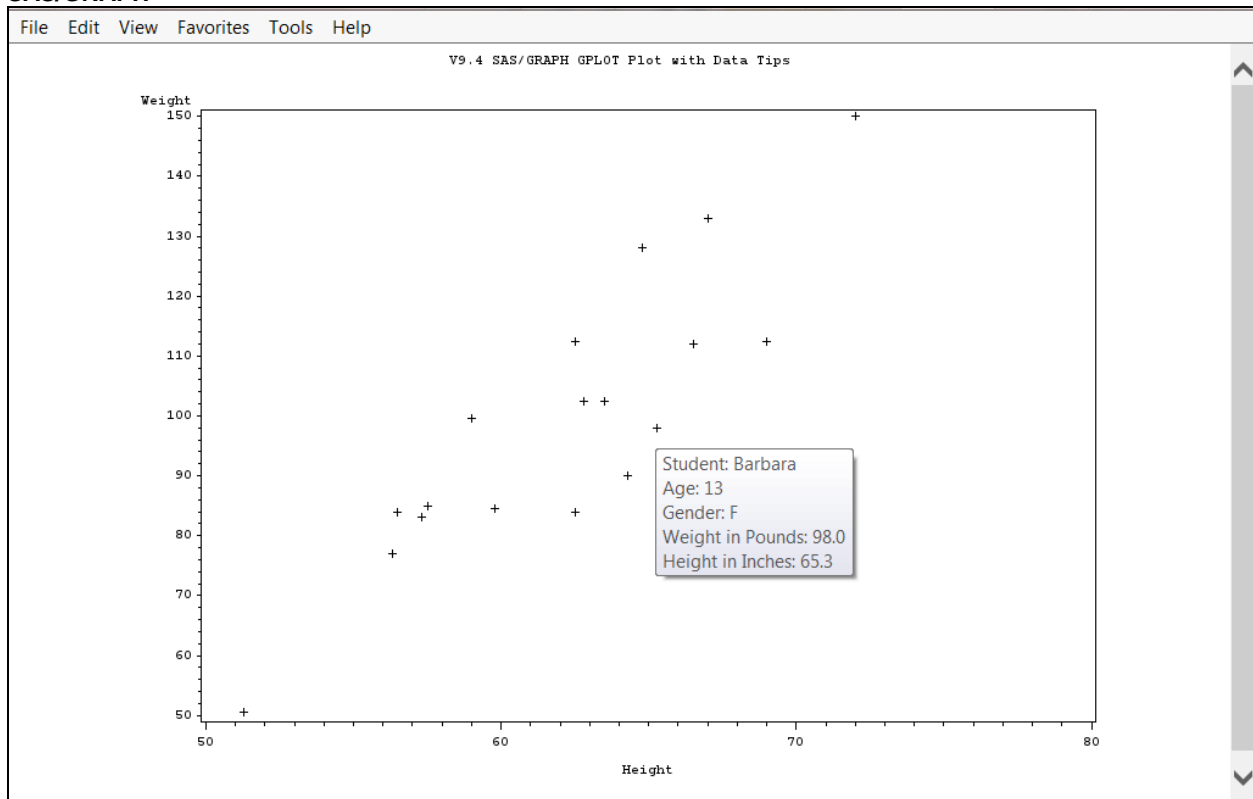
On the next page is a comparison of data tips producible using ODS Graphics (with custom coding in a preliminary DATA step) versus using SAS/GRAPH (with simple syntax in the graph creation step). In both cases, the graph formatting takes all of the defaults. They are not pretty. The default image size in ODS graphics (480 x pixels by 360 y pixels) is much smaller than that in SAS/GRAPH (800 x pixels by 600 y pixels). The code for these examples is in Appendix A.

In following sections, Cases B through I for handling time series plots are discussed and displayed, and the code used to create them is in Appendices B through I.

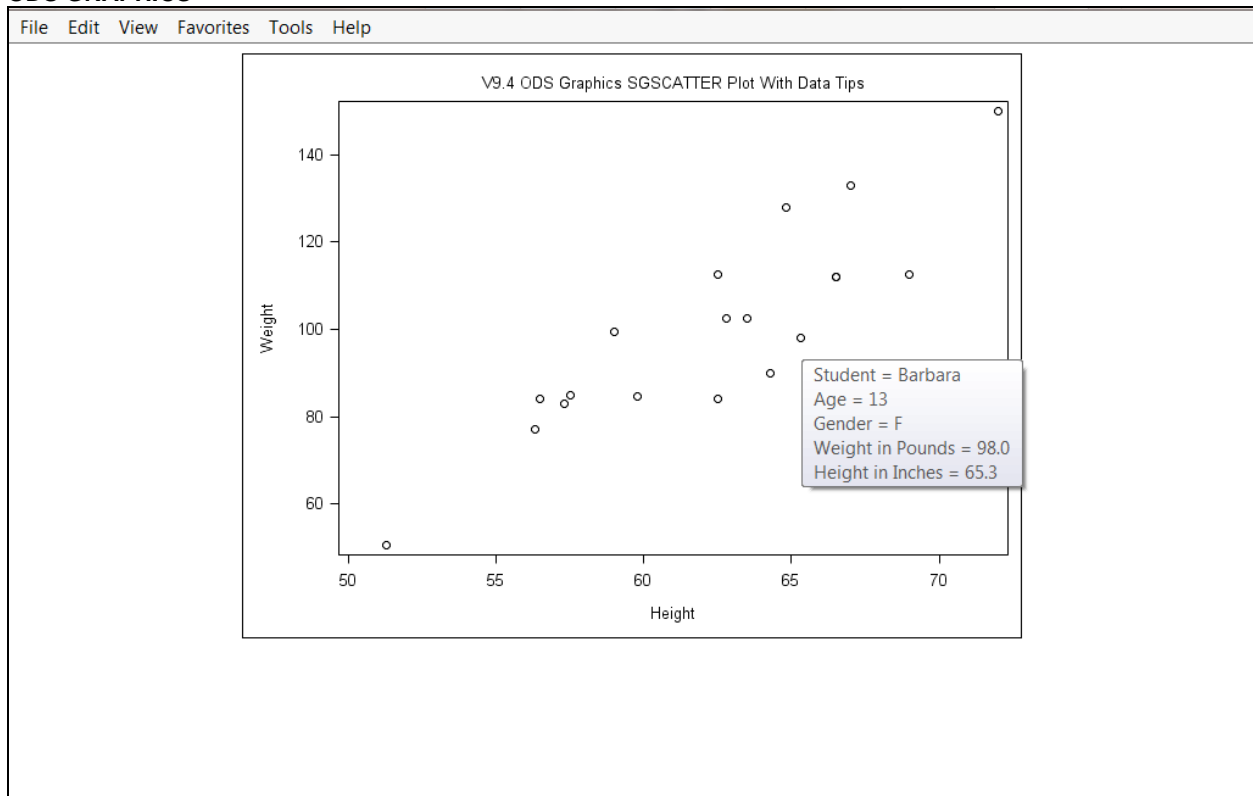
Then I will take up bar charts and pie charts in Cases J through M, and their code is in Appendices J through L, with the code for Case M being Reference 1.

A. WEB GRAPH DATA TIPS USING SAS/GRAPH VS. ODS GRAPHICS

SAS/GRAPH



ODS GRAPHICS

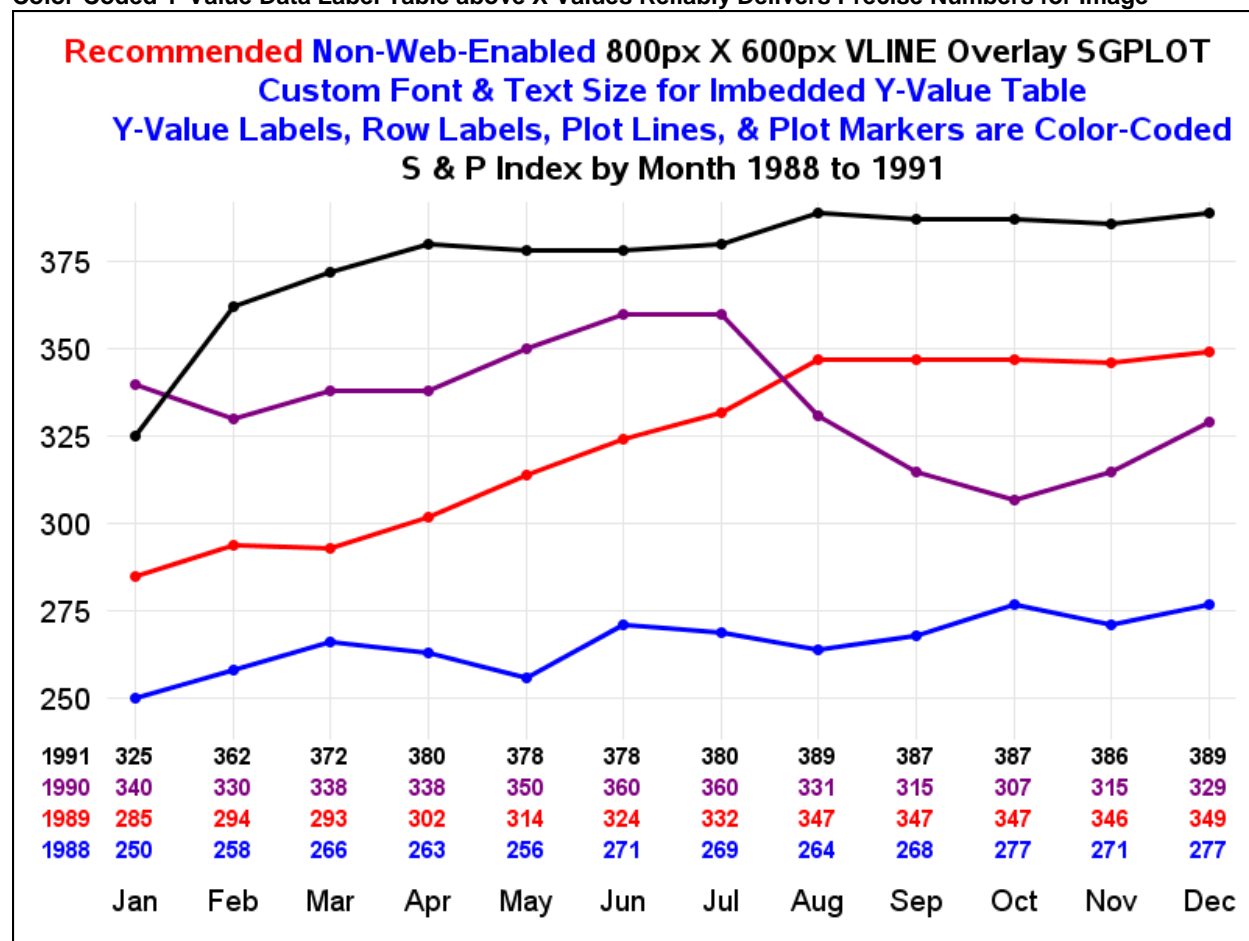


See the code for these graphs in Appendix A.

B. 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 ODS GRAPHICS DATALABEL option. You are able to control their text characteristics (font, size, color, etc.) with the DATALABELATTRS option. For a single-line plot, the labels can sometimes be safely annotated adjacent to the plot points. For a multi-line plot, the reliably successful alternative shown below removes the Y value labels out of the plot area and puts them in a table that is packaged within the graph immediately above the X-axis tick mark values. This feature mimics an analogous solution that I created several years ago using SAS/GRAPH. That code was complex, but this ODS Graphics built-in feature makes the task easy. **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.** When you have too many X values, you need to rely on a web-enabled plot with data tips, provided as pop-up boxes containing the Y value, the X value, and the line descriptor (here, the year). Also, a web-enabled plot can be hyperlinked to a spreadsheet of all of the input data, where the spreadsheet can be hyperlinked back to the plot. More about that later. For that solution, you would omit the imbedded data label table, since its entries would have to be too small to be readable.

Color-Coded Y-Value Data Label Table above X Values Reliably Delivers Precise Numbers for Image



See Appendix B for the code to create the graph above.

C. COMMUNICATION-EFFECTIVE MULTI-LINE PLOT WITH Y VALUES ANNOTATED: ANOTHER WAY TO SHOW IMAGE + DETAIL FOR RTF/Word, PDF, PowerPoint, or PRINT

The Y values for a plot are easily annotated along the plot line near their plot points with the ODS GRAPHICS DATALABEL option. You are able to control their text characteristics (font, size, color, etc.) with the DATALABELATTRS option. For a single-line plot that is not extremely dense in data points, the labels can usually be safely annotated adjacent to the points. (For a multi-line plot, the reliably successful alternative shown above removes Y value labels out of the plot area and puts them in a table immediately above the X-axis tick mark values.)

The problem with annotation is the possibility of “collisions” (i.e., overlap of label and line or overlap of the label for one point and the label for another point).

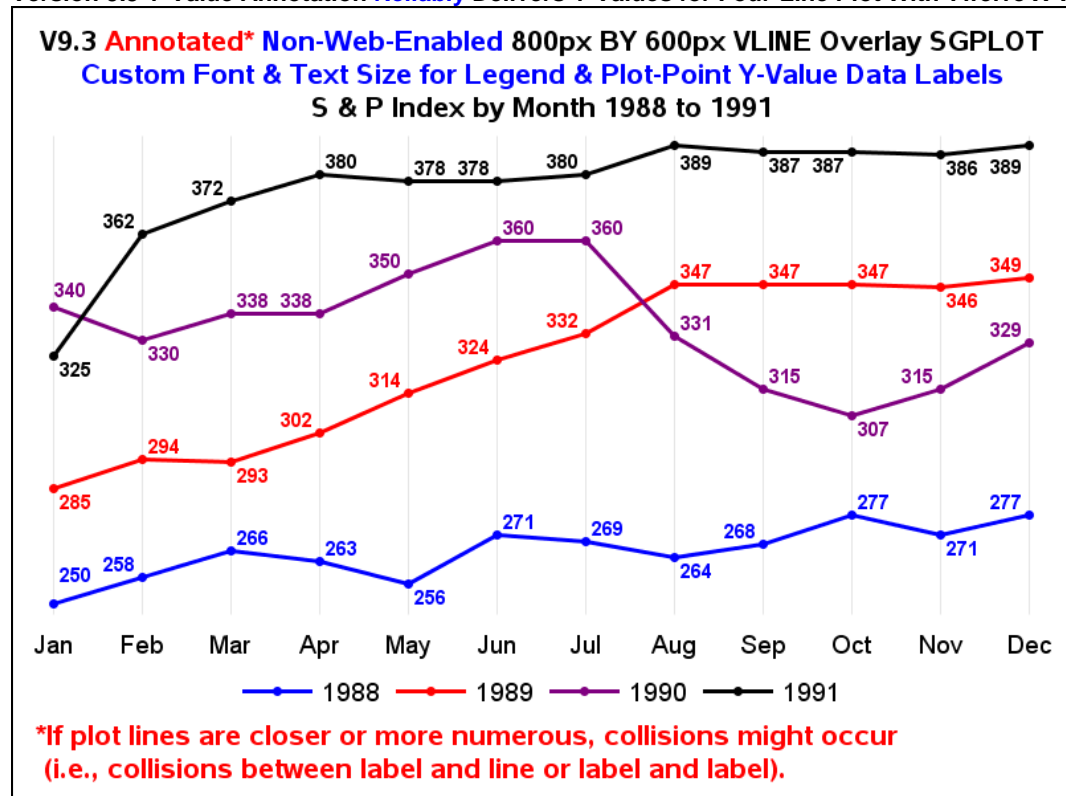
SAS/GRAPH supports plot point annotation with the POINTLABEL option on the SYMBOL statement(s) pertaining to the line(s) to be annotated. It allows annotation of Y values, X values, or both.

In some prior work I developed a macro that is more collision-resistant than the POINTLABEL option. See “Show Them the Trend, Show Them the Numbers: The %TRNDANNO Macro for Collision-Free Annotation of Trend Plots”, published in the hardcopy-only *SUCHI Proceedings*, 13 November 2000. The macro was enhanced, renamed, and published (with co-author Francesca Pierri) as “%TREND: A Macro to Produce Maximally Informative Trend Charts with SAS/GRAPH, SAS, and ODS for the Web or Hardcopy” in *SUGI 2002 Proceedings* and is available at: <http://www2.sas.com/proceedings/sugi27/p093-27.pdf> Both versions of the macro support the options of Sparse Annotation, which is discussed later in Section D, but the enhanced version supports web-enabled trend lines, with added options of the trend line linked to a table in a separate web page or with a table below it in the same web page.

This paper instead uses ODS Graphics for annotated plots.

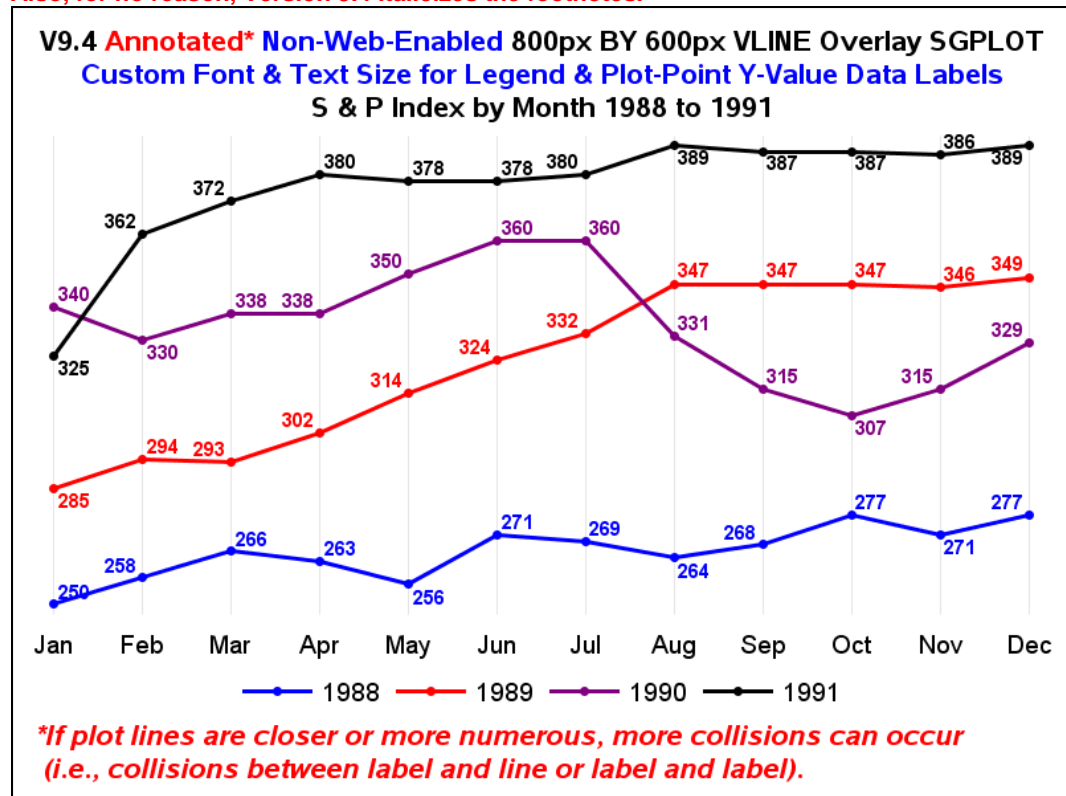
For the multi-line overlay plot (shown previously using a data label table rather than annotation) the lines happen to be well separated, and, as shown on the following page, the multiplicity of lines does not cause any data label collision problems in Version 9.3, but see what happens in Version 9.4 for January 1988. For these four-line plots, text sizes for annotated Y values and for X axis values were customized. There is no need for a Y axis.

See Appendix C for the code to create these graphs.



Version 9.4 Y-Value Annotation **Imperfectly** Delivers Y Values for Four-Line Plot With Twelve X Values
 See label-line collision for Jan 1988.

Also, for no reason, Version 9.4 italicizes the footnotes.



NOTE: Version 9.4 needlessly italicizes the footnotes, not at programmer request. Why?

D. EXPLORING THE LIMITS OF USING A PLOT WITH ANNOTATED Y VALUES IN IMAGES FOR RTF/Word, PDF, PowerPoint, or PRINT

Now we explore what happens to annotated single-line plots as we increase the number of plot points. Since the Y values are annotated, there is no need for a vertical axis and its tick mark values. For these examples, Version 9.3 is used, since Version 9.4 cannot be expected to perform better. You can verify that for yourself, if interested.

For the single-line plots shown on the following pages, text sizes for annotated Y values and for X axis values are left at the defaults. Note that the default annotate text size is rather small, presumably to reduce the probability of collisions between labels or between labels and the line.

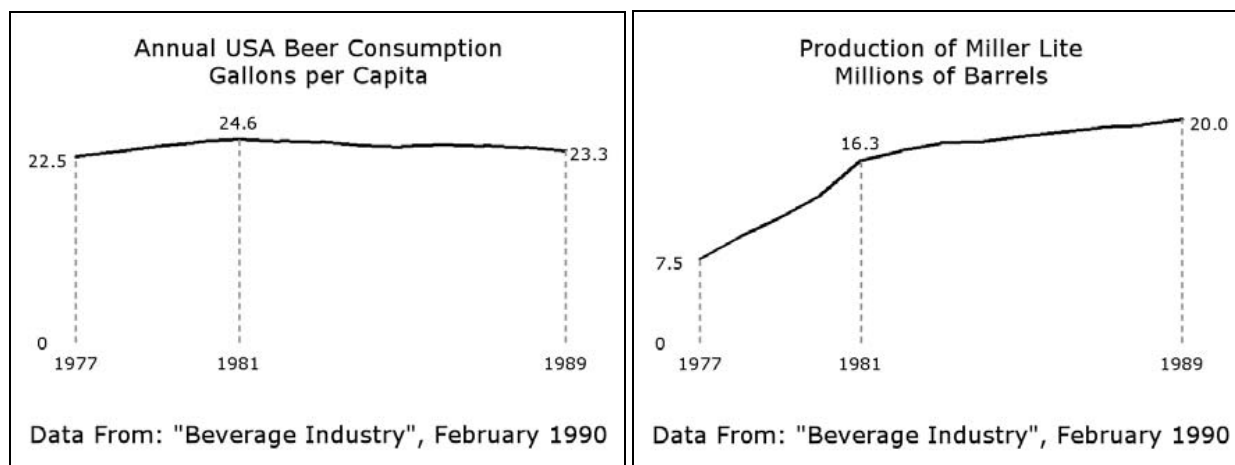
In the examples, presence of the optional subtitle of key statistics has no material effect on the placement of annotated values. This optional subtitle is also present in some other examples. It is switched on and off with a macro invocation parameter. You could apply the concept to a different selection of statistics. You could display them as a footnote instead of a subtitle. For a multi-line plot you could provide statistics for each line. Actually, for a multi-line plot that requires a legend, with some fancier coding you could append the statistics to the legend entry descriptions.

In order to retain readable X values without rotating the labels, for these examples, on the xaxis statement the option fitpolicy=thin is used, rather than what would be default for these graphs, namely, fitpolicy=rotate, which rotates the X axis values 45 degrees. Since the X axis values are of the form MMMYY, guessing the intermediate values is easy. (If the axis values were categorical character variables, which can happen for a bar chart, thinning would be unacceptable because there would be no way to guess the missing values.)

See Appendix D for the code to create the graphs on the following pages.

But let me digress and show you another “limit” on annotating a plot. In 1993, many years before the concept of spark lines (reputedly inspired by something seen in a notebook of Galileo) became fashionable, I advocated the **Sparsely Annotated Trend Chart**. It answers four common questions. How are we doing right now? What is the trend? What have been the best and worst numbers? When did the trend change?

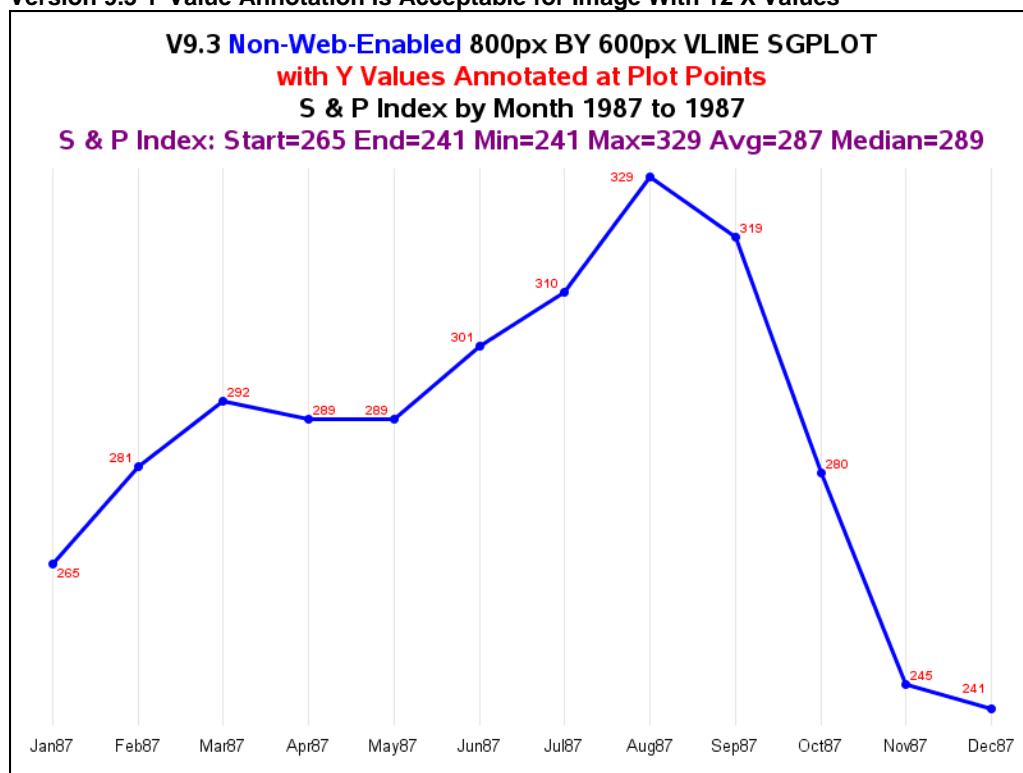
It's my contention that most of the time the only trend Y values of great interest are the Starting Value, the Ending Value, and any Maximum and/or Minimum along the way. The right-hand trend line below demonstrates that another situation of interest is when there is a point where the trend changes in a persistent fashion.



Note that the Y axis in both examples starts at zero. I recommend this choice of Y axis in order to prevent undue elation or undue distress about changes that are not necessarily significant, but get magnified by full use of the available vertical display space. Significance of an image needs also to be gauged by the actual precise numbers (i.e., absolute or relative magnitude of change). It is interesting that when *Beverage Industry* described the long term trend in beer consumption over the time period shown above in the left-hand graph, it was described as “flat”. Starting the Y axis at y=0 best shows the near flatness of the trend.

I first showed and extolled the benefits of sparse annotation in “Chart Smart: Designing Graphs and Tables for Effective Communication” published in the SUGI 1993 Conference Proceedings and available at <http://www.sascommunity.org/sugi/SUGI93/Sugi-93-123%20Bessler.pdf> The only place that I ever published the code, which uses SAS/GRAPH, for the above plots is the paper “Chart Smart: Design and Build SAS Graphs That Inform and Influence” published in the MWSUG 2007 Conference Proceedings and available at <http://www.lexjansen.com/mwsug/2007/dataviz/MWSUG-2007-D01.pdf>

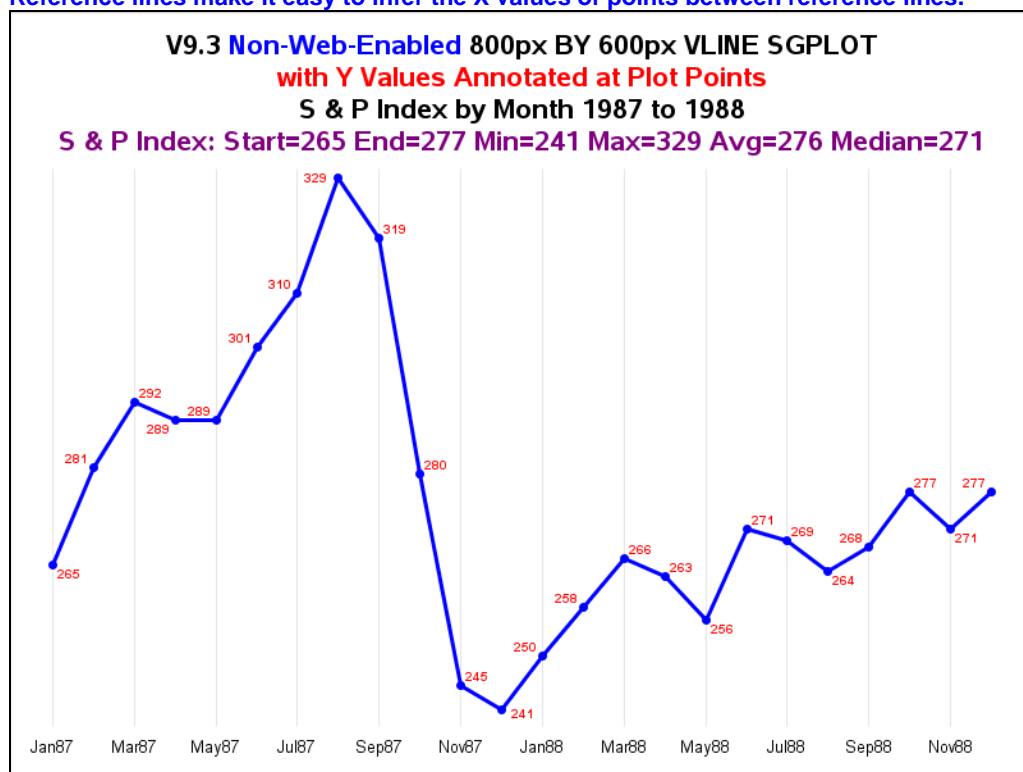
Version 9.3 Y-Value Annotation Is Acceptable for Image With 12 X Values



Version 9.3 Y-Value Annotation Is Acceptable for Image With 24 X Values

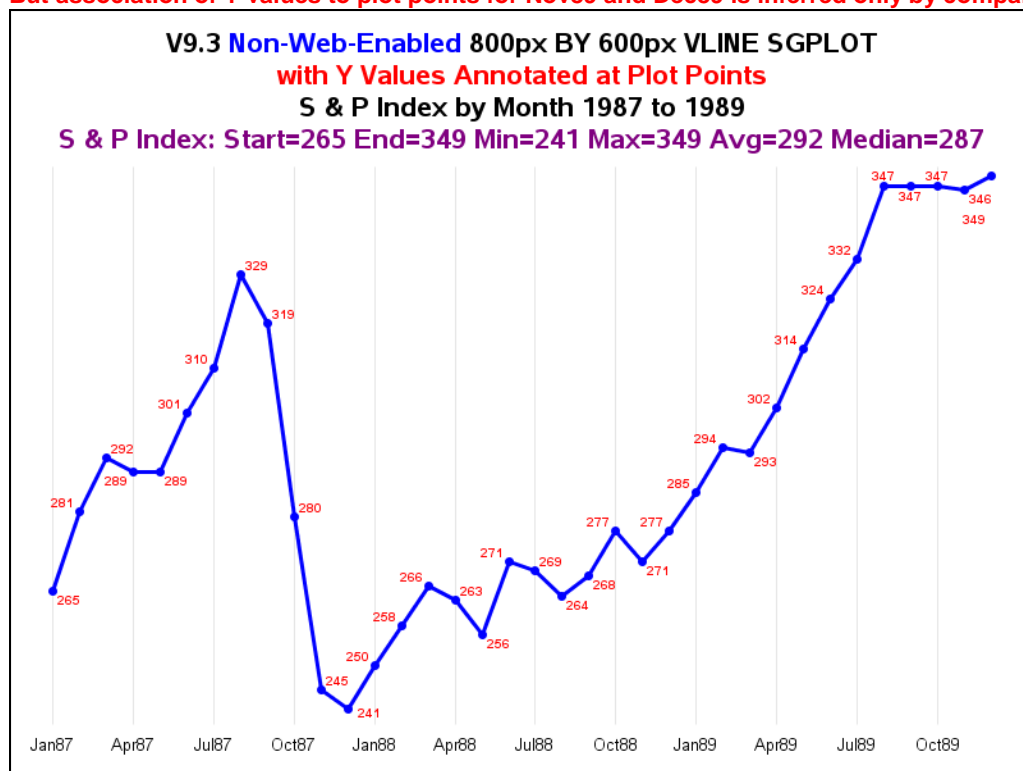
Thinning of X values begins. It is better than rotated text.

Reference lines make it easy to infer the X values of points between reference lines.



Version 9.3 Y-Value Annotation Is Acceptable for Image With 36 X Values

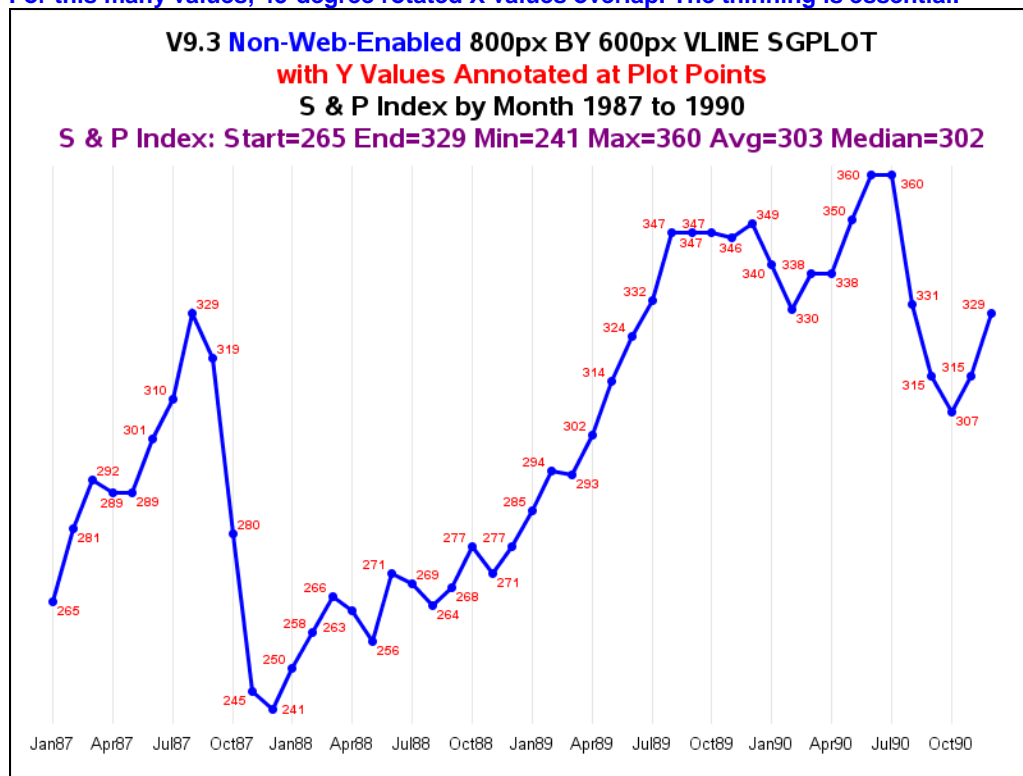
But association of Y values to plot points for Nov89 and Dec89 is inferred only by comparison of values.



Version 9.3 Y-Value Annotation Reliably Delivers Y Values for Image With 48 X Values

Y value for Oct89 is poorly placed above or below plot point for Sep89.

For this many values, 45-degree rotated X values overlap. The thinning is essential.

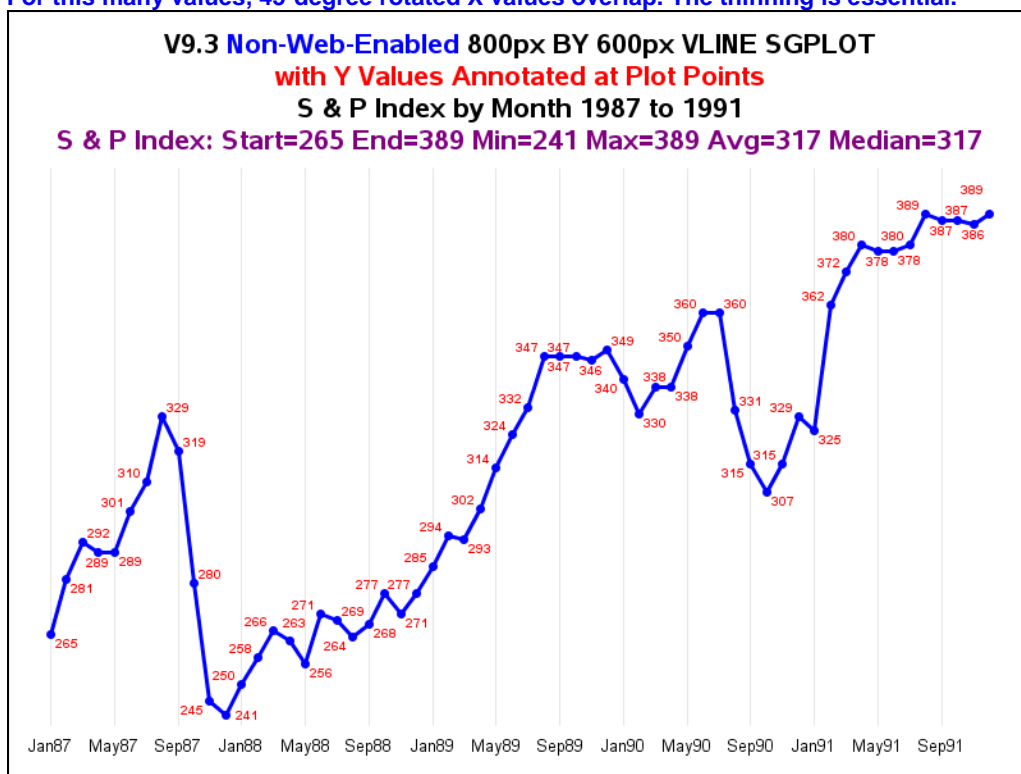


Version 9.3 Y-Value Annotation Reliably Delivers Y Values for Image With 60 X Values

Y value for Oct89 is poorly placed above or below plot point for Sep89.

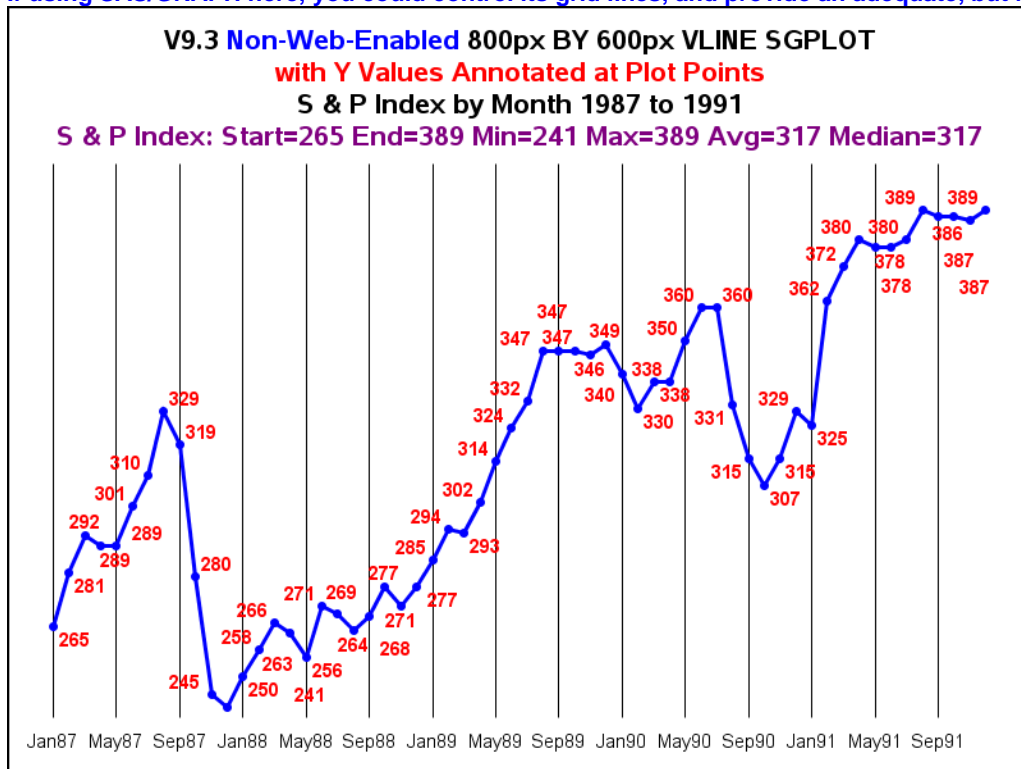
Correct association of Y value to plot points for Jun91 and Jul91 is inferred only by comparison of values.

For this many values, 45-degree rotated X values overlap. The thinning is essential.



Above, the reference grid for X values is not very prominent. Below, switching from ODS style HTMLblue to Minimal creates an overly prominent grid and, by default, larger data labels, but with worse placement.

If using SAS/GRAPH here, you could control its grid lines, and provide an adequate, but fainter, grid.



E. EXPLORING THE LIMITS OF USING THE PLOT DATA LABEL TABLE FOR Y VALUES IN IMAGES FOR RTF/Word, PDF, PowerPoint, or PRINT

As long as the text is small enough, yet readable, the best way to report the Y value associated with each X value for a plot, whether single-line or multi-line, is via the data label table sitting above the X axis.

In this example, the image was created in the size 800 X pixels by 600 Y pixels. The physical size delivered to, say, Microsoft Word, is 8.33 inches X 6.25 inches. If your target is American letter size paper, which is 8.5 inches wide and 11 inches tall, and you use 1 inch margins, the image when imbedded is squashed from 8.33 inches to 6.25 inches. The difference from what you might see on your computer screen will be non-trivial, but not enormous. If you insert the image into a PowerPoint slide, which is formatted with the traditional 4 to 3 ratio for width to height and with a 10 inch width and 7.5 inch height, the image ends up centered and with comfortable margins all the way around. Most, but not all, projection screens still have the traditional 4 to 3 ratio for width to height.

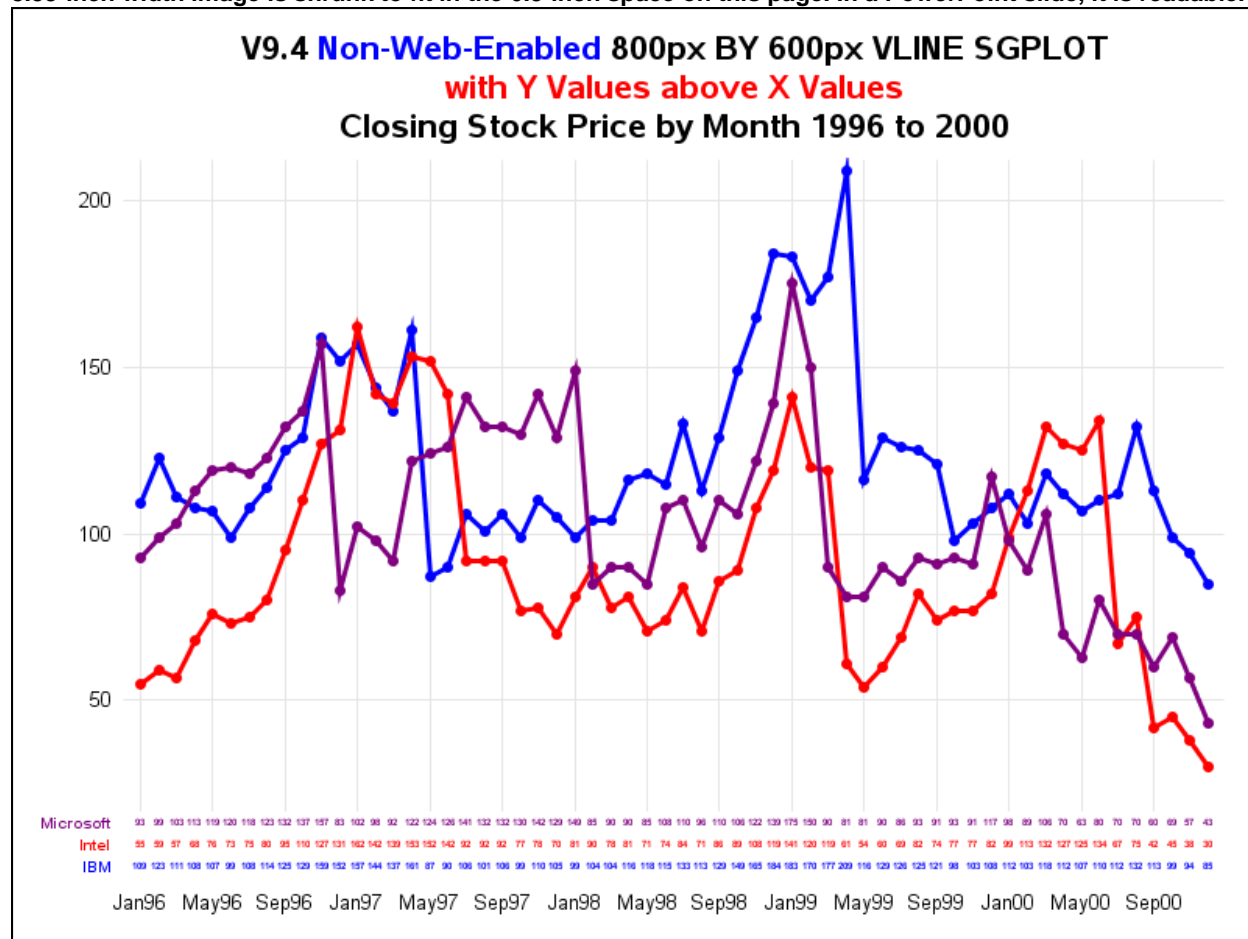
Of course, if you are formatting an image for display on a wide screen laptop, possibly with 1920 X 1080 resolution, obviously you can build a bigger image. A safe policy is to develop for the worst case, a skimpy target area instead.

The data label table avoids the possibility of line-label collisions and label-label collisions and as well as labels awkwardly placed relative to their relevant points.

Except for the titles, all of the text in this example takes the default sizes. The option of thinning the tick mark values for the X axis was turned on.

With some extra coding, it would have been possible to create three footnotes to display a line of statistics for each stock over the time period. **See Appendix E for the code to create this graph.**

Version 9.4 Y-Value Data Label Table Is, for my eyes, UNreadable for this image with 60 X Values, when the 8.33-inch width image is shrunk to fit in the 6.5-inch space on this page. In a PowerPoint slide, it is readable.



F. COMMUNICATION-EFFECTIVE PLOT WITH DENSE DATA POINTS: HOW TO PROVIDE IMAGE PLUS DETAIL WITH WEB ENABLEMENT AND A COMPANION SPREADSHEET

When you have too many X values it is infeasible to use the imbedded Y-value table above the X-axis tick mark values. Even when you have few enough X values, and have reduced their text size, so that there is an X value below every Y value (or stack of Y values), the table of data is not “live” and cannot be worked with, only viewed.

When you need to create a plot with dense data points, web enablement delivers two advantages: (a) the ability to display a tiny pop-up table of all of the point’s variable values (including line descriptor if it’s a multi-line plot) at each data point—providing a soft annotation of all relevant information AT the plot point, so that there is no association ambiguity; and (b) a link to a spreadsheet of all the plot data; The spreadsheet is a “live” resource that can be manipulated and reused as the viewer sees fit, using a tool that everyone has and knows how to use. The spreadsheet can be linked back to the plot. With SAS/GRAPH, the custom-coded pop-up data label can optionally carry any additional information that the programmer chooses to include. With ODS Graphics, additional information could be supplied via an extra variable added to each observation and included in the `tip=(var1 var2 . . .)` assignment on, e.g., the PLOT statement.

On the next two pages are three of four variations on the web graph, and a spreadsheet linked to one of them.

In the example with a subtitle line of statistics, one could include instead any statistics of her/his choosing. Also, the information could be moved to a footnote, if preferred.

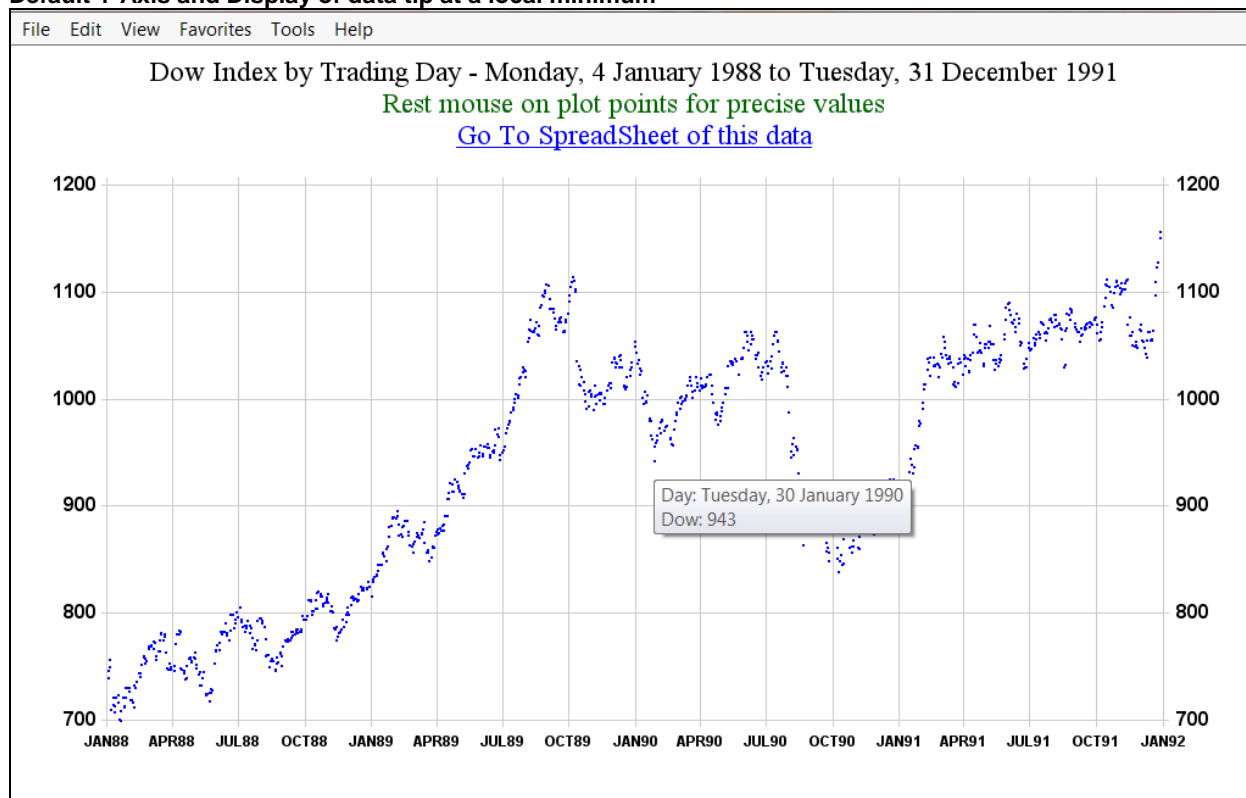
NOTE: In the case of the non-web-enabled overlay plots shown in the prior sections, one could, there also, include a footnote of statistics for each of the plot lines. The footnote body text would best be rendered in black, but each footnote would begin with the year value in its coded color.

There are two different ways of using the vertical plot space. One is to let SAS/GRAPH choose the range and the tick mark values. This makes maximum drawing use of the vertical display space available. However, instead starting the Y axis at zero eliminates the tendency of a graph viewer to over-react to increases or decreases that are really not as significant as they look due to the magnification caused by maximally using the vertical space.

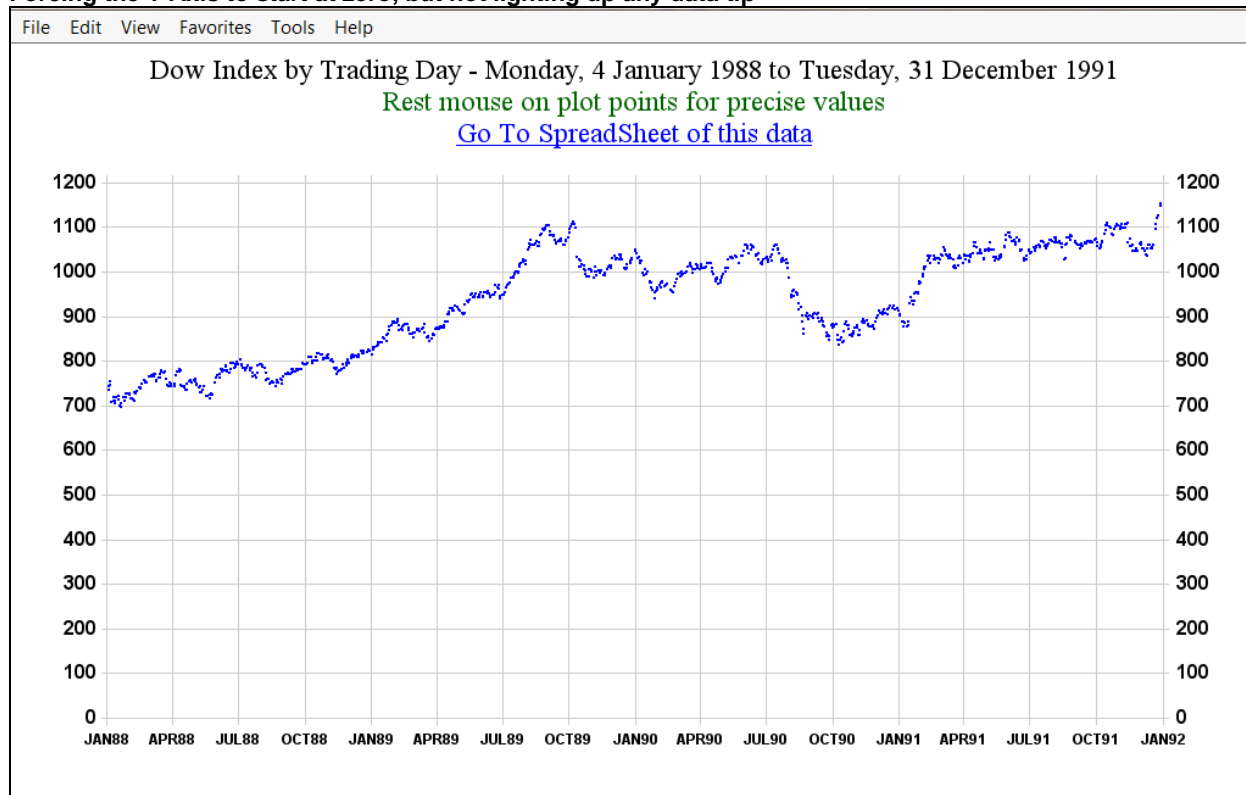
In these examples, the two different ways of using the vertical plot space are independent web graphs. However, one could modify the code to support the option of creating the two different views as hyperlinked to each other, so that the graph viewer would have the choice of how she/he wishes to view the graph.

See Appendix F for the code to create the graphs on the next two pages.

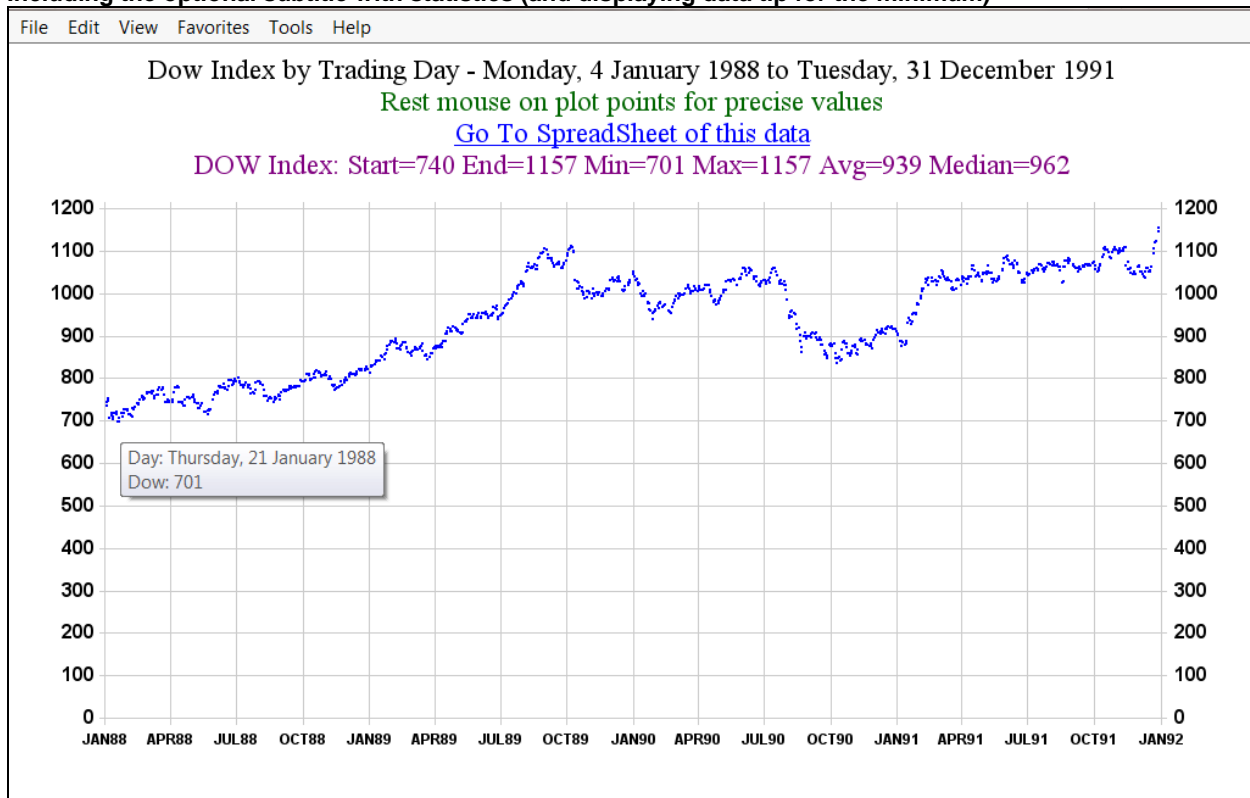
Default Y Axis and Display of data tip at a local minimum



Forcing the Y Axis to start at zero, but not lighting up any data tip



Including the optional subtitle with statistics (and displaying data tip for the minimum)



After linking to the spreadsheet of the input data and manually highlighting the row with minimum Dow value (Notice the link back to the graph in Row 2.)

A18 701									
1	Dow Index by Trading Day - Monday, 4 January 1988 to Tuesday, 31 December 1991								
2	Go To Graph of This Data								
3									
4	Dow	Year	Date of Observation						
5	740	1988	Monday, 4 January 1988						
6	747	1988	Tuesday, 5 January 1988						
7	750	1988	Wednesday, 6 January 1988						
8	757	1988	Thursday, 7 January 1988						
9	711	1988	Friday, 8 January 1988						
10	722	1988	Monday, 11 January 1988						
11	715	1988	Tuesday, 12 January 1988						
12	714	1988	Wednesday, 13 January 1988						
13	708	1988	Thursday, 14 January 1988						
14	722	1988	Friday, 15 January 1988						
15	724	1988	Monday, 18 January 1988						
16	718	1988	Tuesday, 19 January 1988						
17	702	1988	Wednesday, 20 January 1988						
18	701	1988	Thursday, 21 January 1988						
19	709	1988	Friday, 22 January 1988						

Ready | Average: 1344.5 | Count: 3 | Sum: 2689 | 100%

G. COMMUNICATION-EFFECTIVE INSIGHTS INTO PLOTS OF BIG (I.E., WIDE) DATA: LINKING A BIG PICTURE OF A LONG TIME PERIOD AND PICTURES OF SUB-PERIODS

In the web graphs above, there was no way to look at individual years.

A not surprising solution is to link together a web graph for the full period and web graphs of the yearly periods, and with each of the yearly graphs and the full-period graph individually hyperlinked to a spreadsheet of its input data.

However, since the Y data range of the web graphs is likely to be different, you need to decide how best to handle that.

On the following pages are three sets of graphs:

- (a) Each web graph uses maximum vertical space suited to its specific Y variable data range.
- (b) All of the graphs are drawn with a Y axis range that matches that of the full time period, with tick mark values for only the minimum and maximum Y values during the full time period.
- (c) All of the graphs are drawn with a Y axis range from zero to the maximum Y value during the full time period, with tick mark values only for zero and the maximum Y value during the full time period.

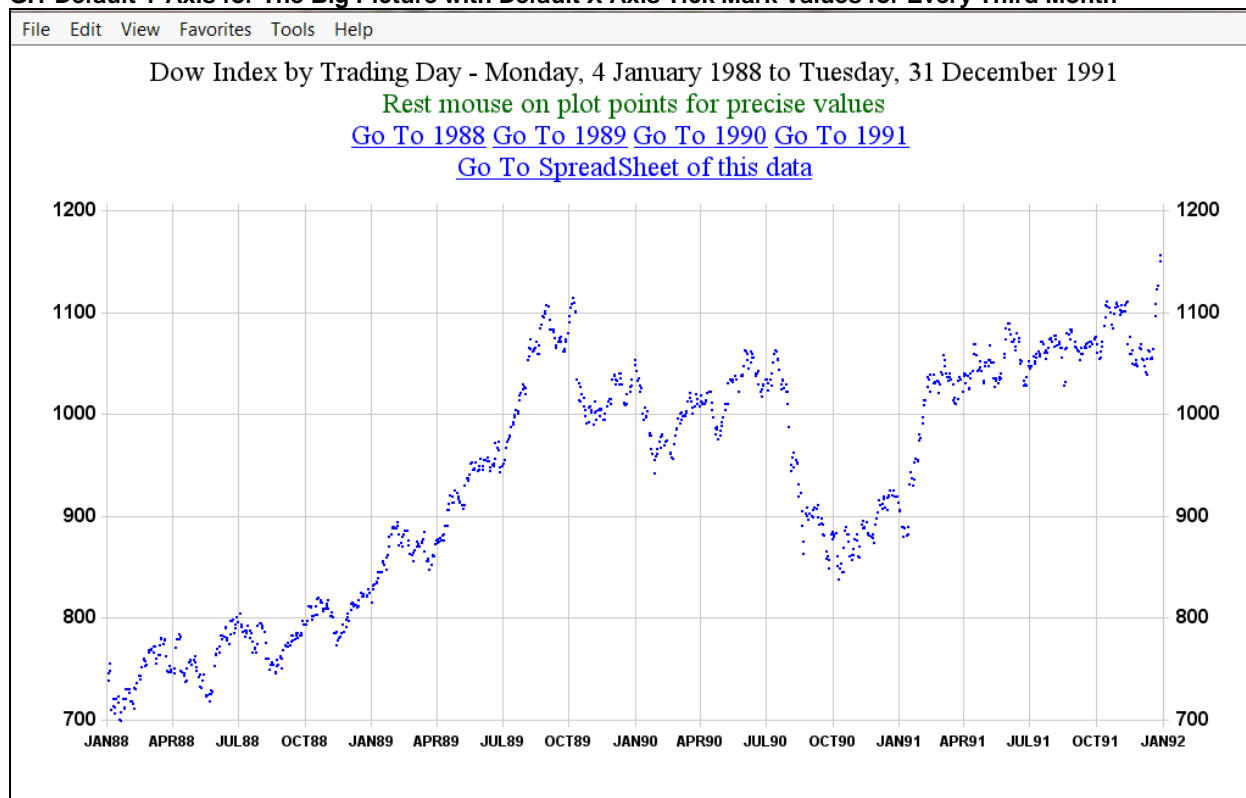
Just as in the prior web plots, data tips are provided for all of the data points.

Absence of any intermediate Y axis tick mark values for Cases (b) and (c) is not a cause for concern. The graphs are web-enabled, and the data tip pop-up box is the reliable way to get the exact Y (and X) value for any plot point, rather than running one's eye from the plot point to an axis and estimating the value relative to the tick mark values.

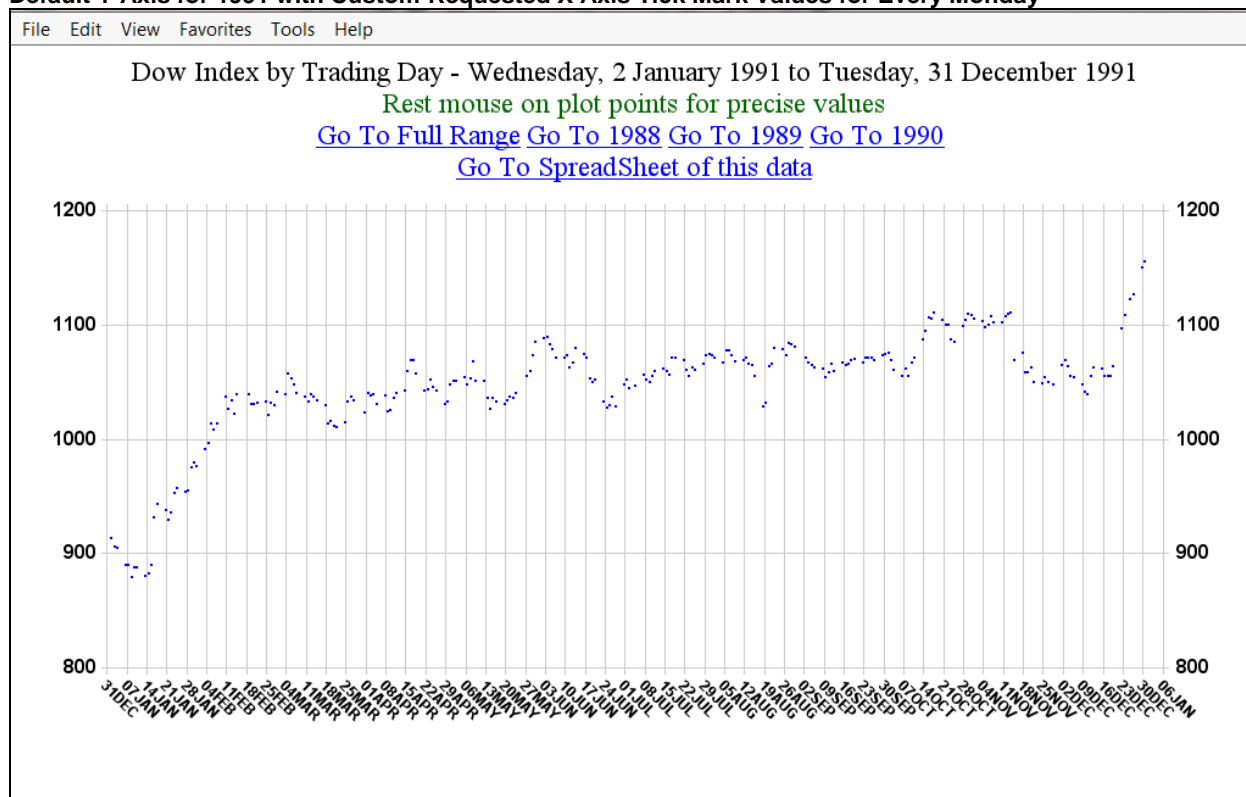
In principle, the three different ways to handle the vertical space could be hyperlinked to each other with clickable switches on each of the graphs. E.g., on any Case (a) web graph, the viewer could have the option to switch to Case (b) or Case (c).

NOT included in these examples is the option of a subtitle or footnote to list the Y variable statistics for the graph.

G.1 Default Y Axis for The Big Picture with Default X Axis Tick Mark Values for Every Third Month

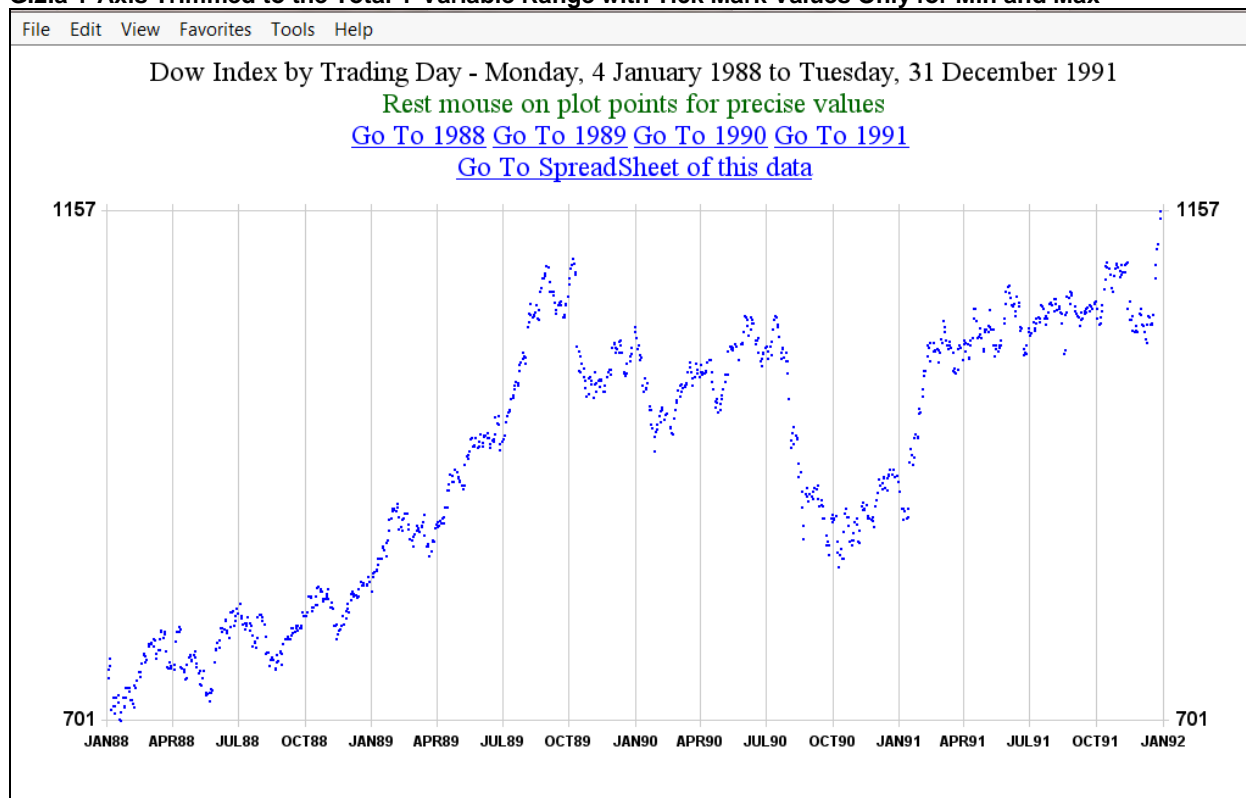


Default Y Axis for 1991 with Custom-Requested X Axis Tick Mark Values for Every Monday

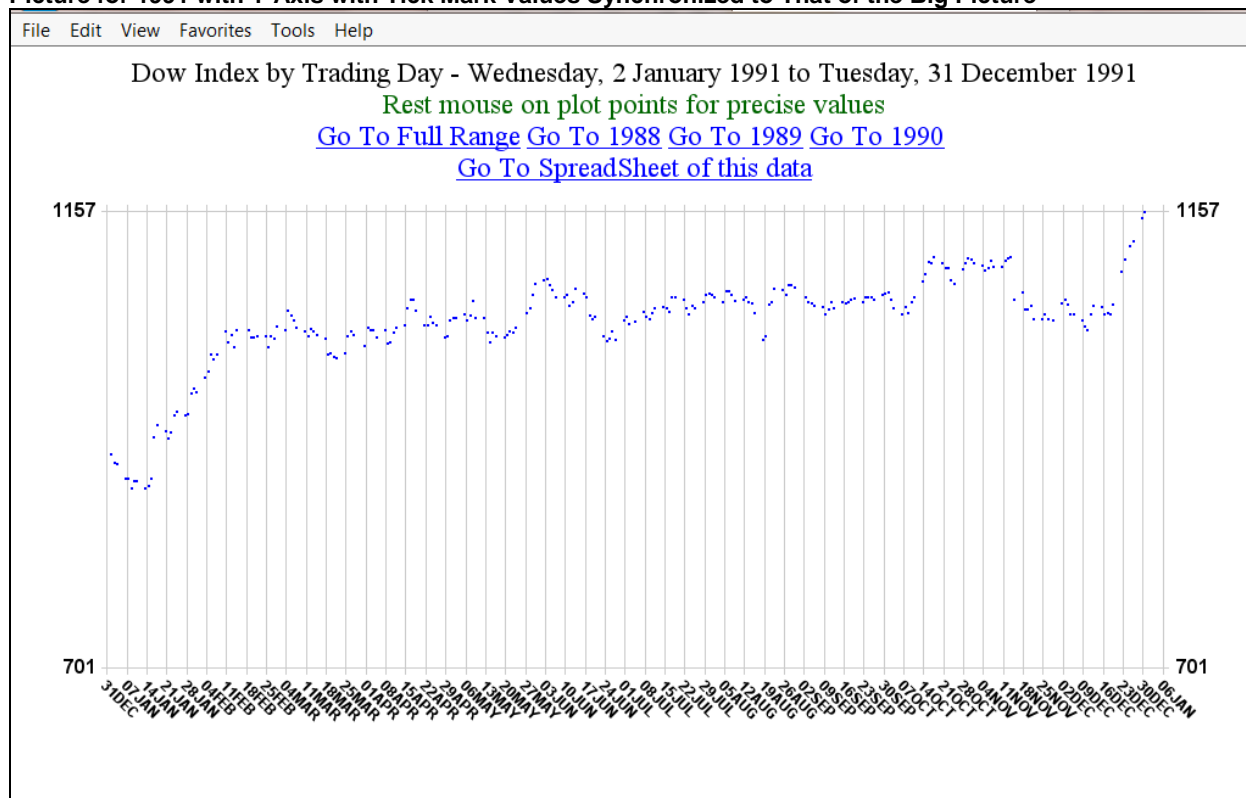


See Appendix G.1 for the code used to create the above graphs.

G.2.a Y Axis Trimmed to the Total Y Variable Range with Tick Mark Values Only for Min and Max

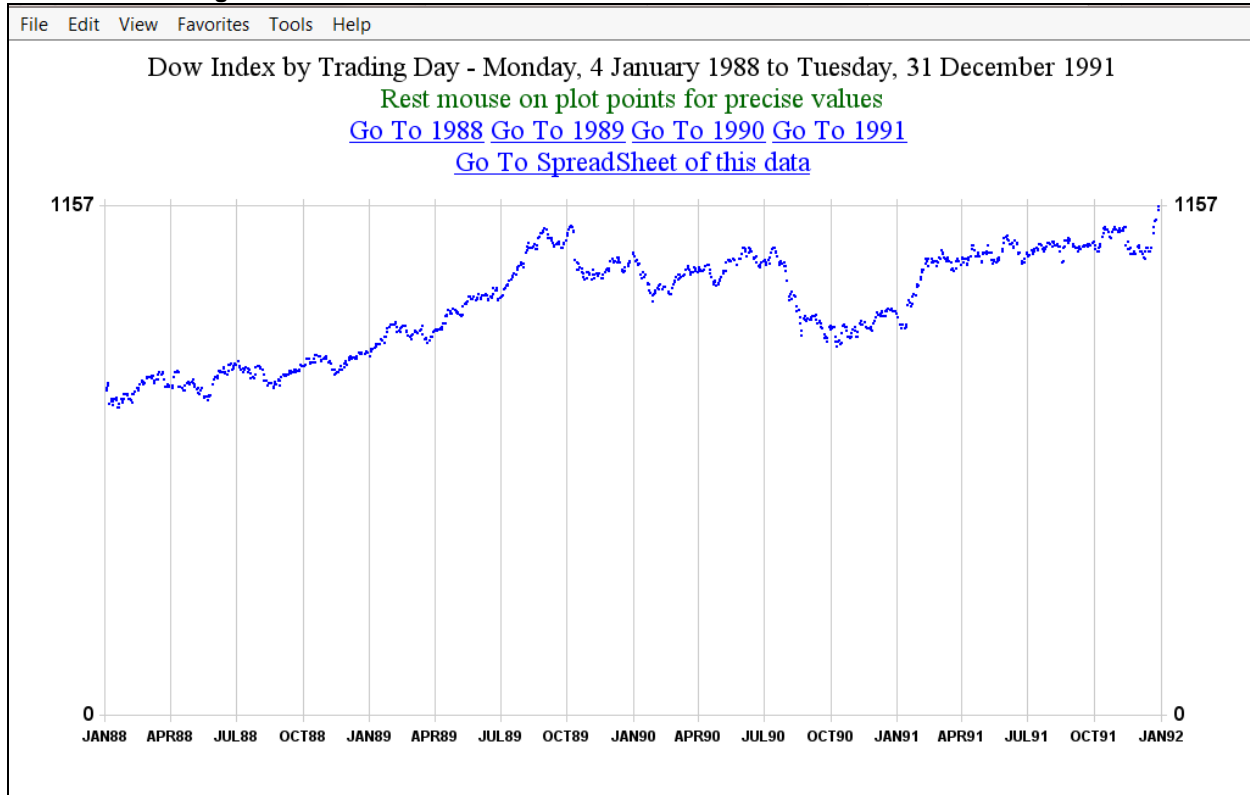


Picture for 1991 with Y Axis with Tick Mark Values Synchronized to That of the Big Picture

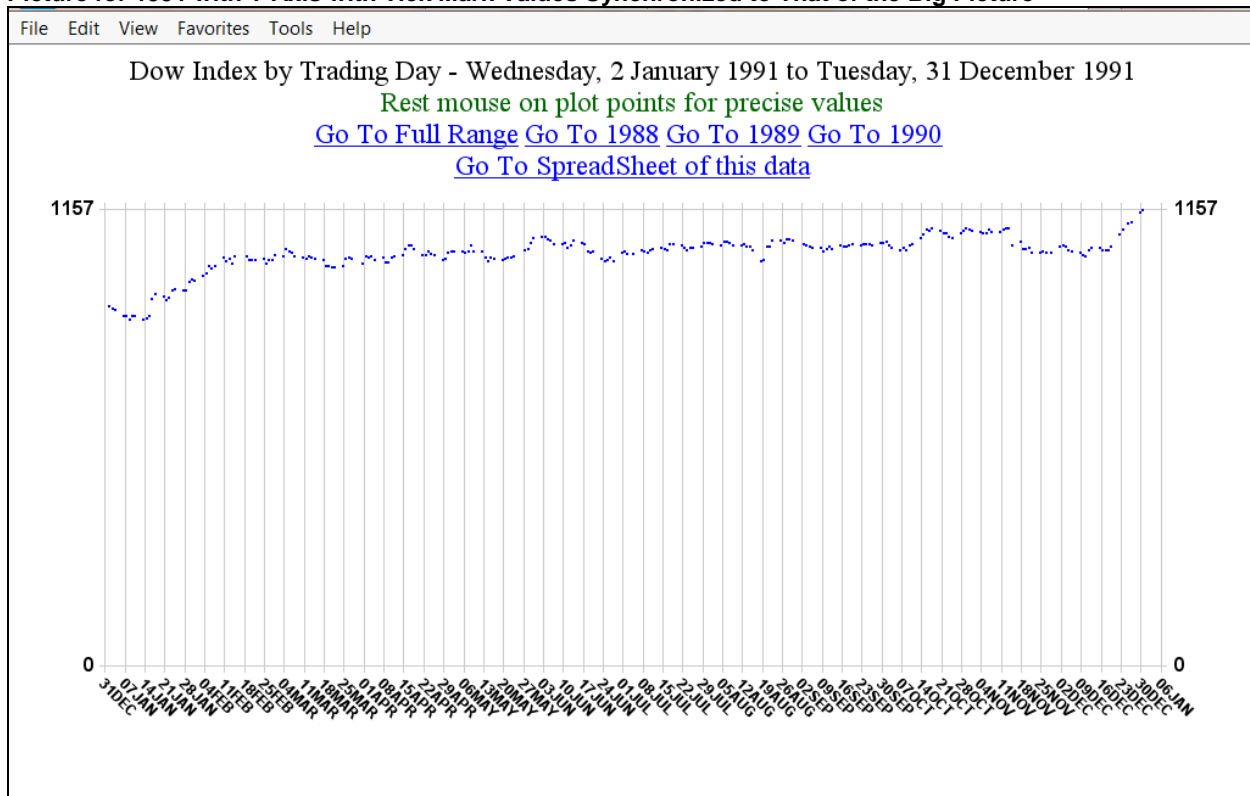


See Appendix G.2 for the code used to create the above graphs.

G.2.b Y Axis Range from Zero to the Full-Period Maximum Y Value



Picture for 1991 with Y Axis with Tick Mark Values Synchronized to That of the Big Picture



See Appendix G.2 for the code used to create the above graphs.

H & I. COMMUNICATION-EFFECTIVE INSIGHTS INTO PLOTS OF BIG (I.E., WIDE) DATA: LINKING A BIG PICTURE OF A LONG TIME PERIOD AND A PANEL OF SUB-PERIODS

In the web graphs above, there was no way to look at all of the individual years in the same window. Any opportunity to observe possible seasonality was denied.

SAS/GRAPH with the HTMLPANEL tagset solves those problems.

On the following pages are two solutions to handle the four years:

- (1) A 2 X 2 panel.
- (2) A one column panel which allows the viewer to detect seasonality effects across the years.

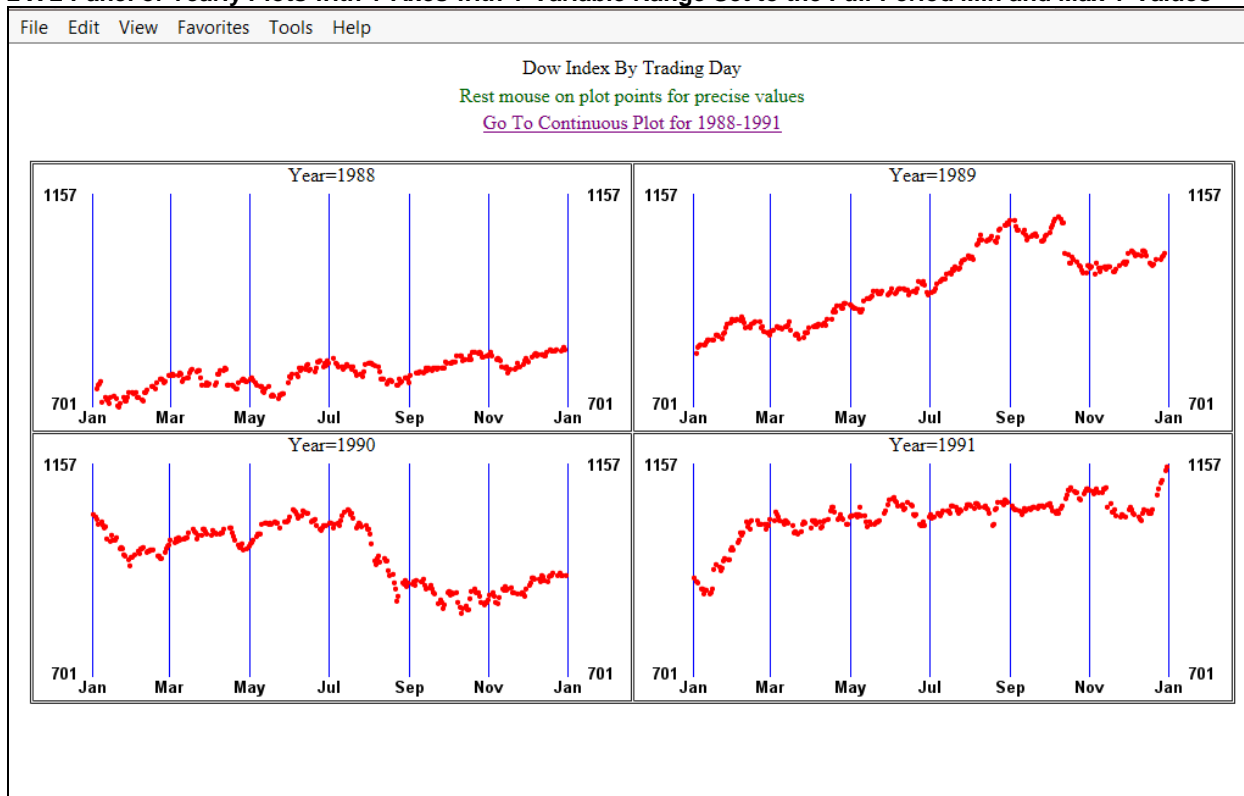
Just as in the prior web plots, data tips are provided for all of the plot points.

Absence of any intermediate Y axis tick mark values is not a cause for concern. The graphs are web-enabled, and the mouse-triggered data tips are the reliable way to get the exact Y (and X) value for any plot point, rather than running one's eye from a plot point to an axis and estimating where the value lies relative to the tick mark values.

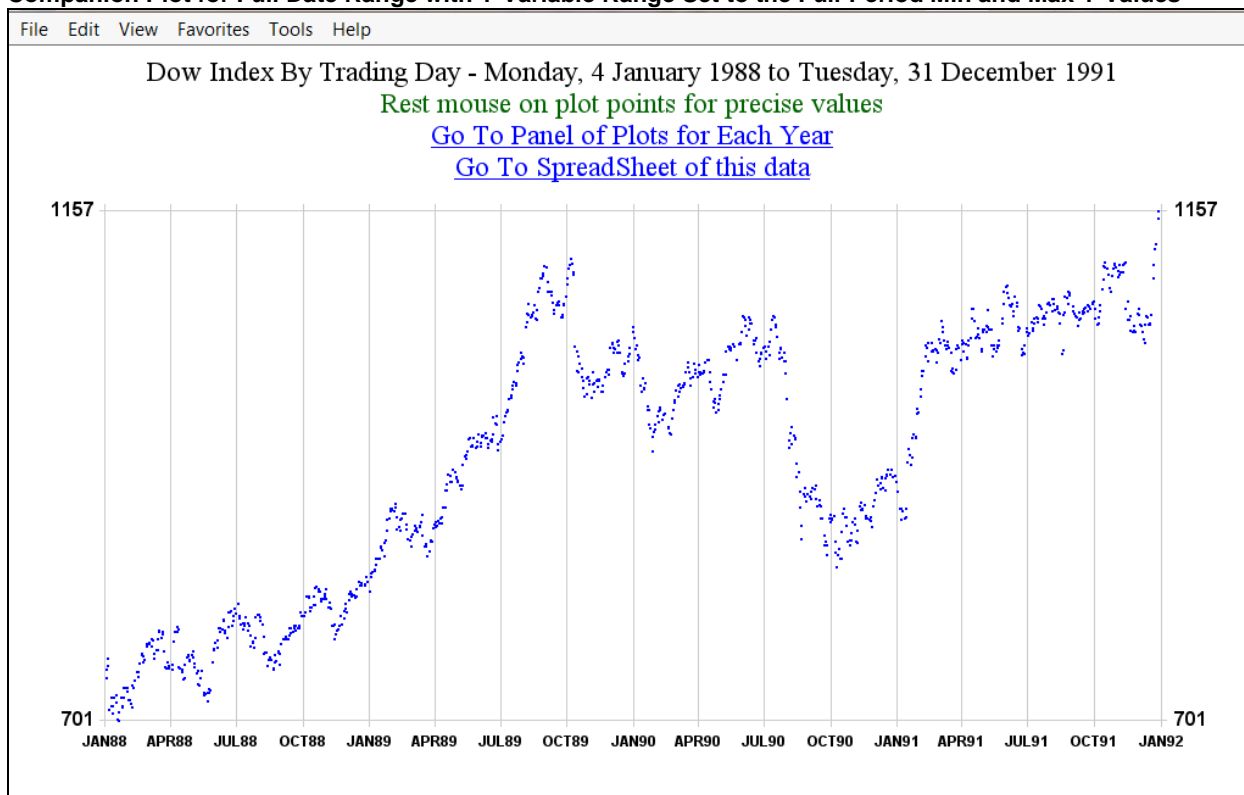
NOT feasible in these examples is the option of a subtitle or footnote to list the Y variable statistics for each graph.

It is impossible to provide a link from the plot of a single year to a spreadsheet of input data for that year.

2 X 2 Panel of Yearly Plots with Y Axes with Y Variable Range Set to the Full-Period Min and Max Y Values

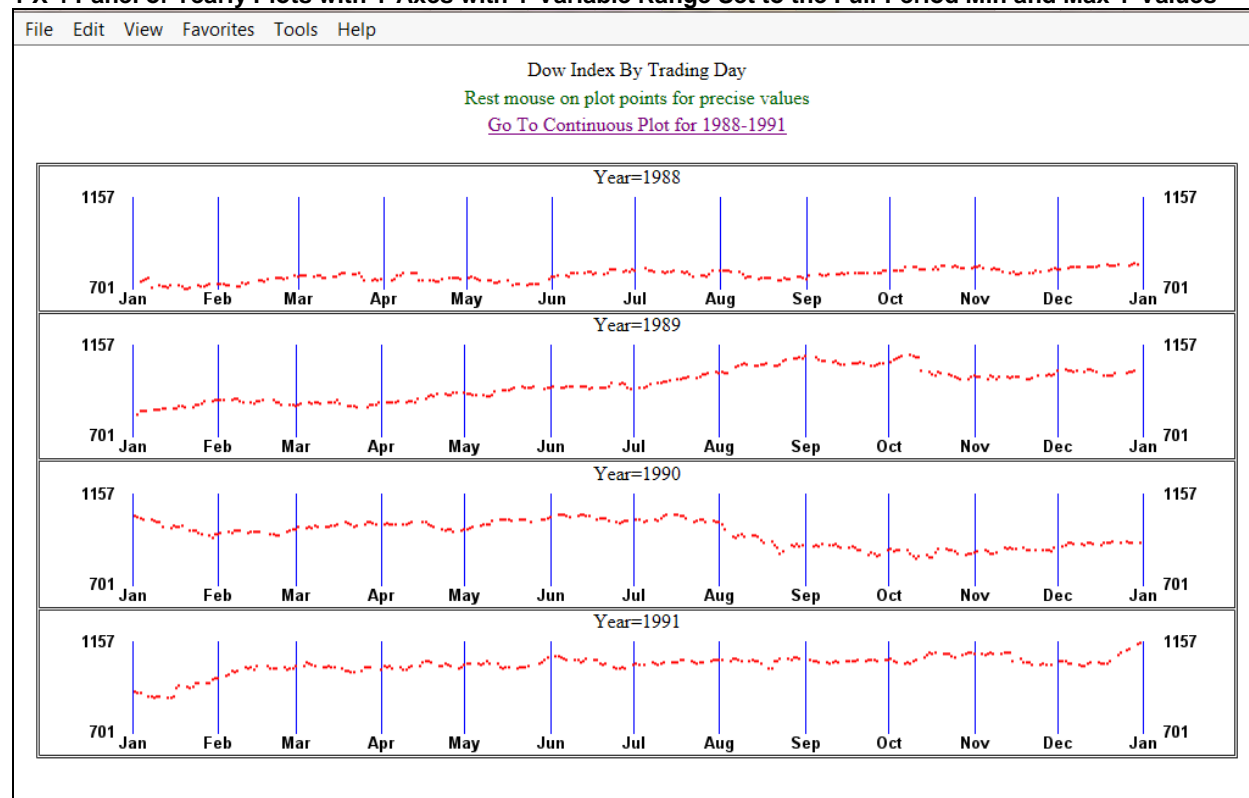


Companion Plot for Full Date Range with Y Variable Range Set to the Full-Period Min and Max Y Values



See Appendix H for the code used to create the above graphs.

1 X 4 Panel of Yearly Plots with Y Axes with Y Variable Range Set to the Full-Period Min and Max Y Values



See Appendix I for the code used to create the above graph.

The Companion Plot for Full Date Range is identical to the companion plot on the previous page for the 2 X 2 Panel of Yearly Plots.

With the stack of yearly plots above, it would be easy to visually discern seasonality effects, if there were any.

NOTE: The macros used to create the two column by two row plots and the one column plot do support the option of forcing all of the plots (by year and continuous) to start the Y axis at zero. Especially in the case of the one column panel, starting the Y axis at zero tends to flatten the Y variability in the plot lines because of the reduced vertical drawing space. With more than four years, the effect could be even more severe.

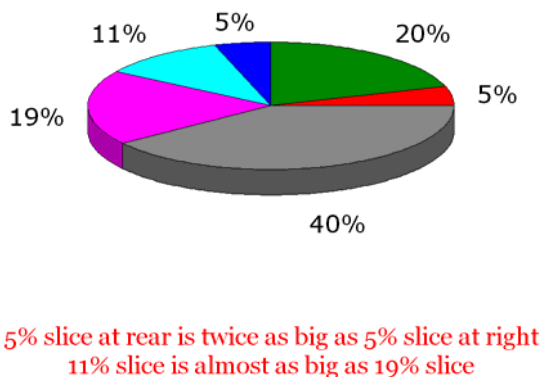
J. 3D IS NEEDED ONLY FOR THREE VARIABLES, AND THE QUEST FOR THE BEST PIE CHARTS

The regrettably popular 3D pie chart offers no communication advantage, but distorts the truth.

Below I have pasted in two previously published images. Please ignore the Figure numbers. These charts were created with SAS/GRAPH. The code is simple and not provided in this paper. I uncovered this phenomenon back in 1995, when to make a 3D pie chart, without some SAS/GRAPH ANNOTATE jiggery-pokery, I had to use an IBM graphics tool. Eventually, SAS Institute provided pie3d, so that users could finally have their way, and be able to unwittingly do damage to data presentation. The problem is inherent to 3D pie charts, not the choice of graphics tool.

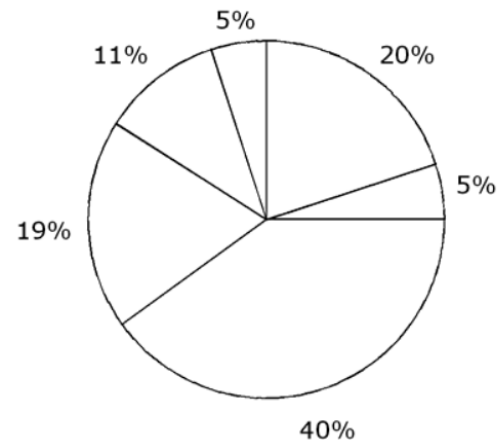
Problem

Figure 1: 3D Pie Chart.
Always Distorts Apparent Relative Size of Slices.



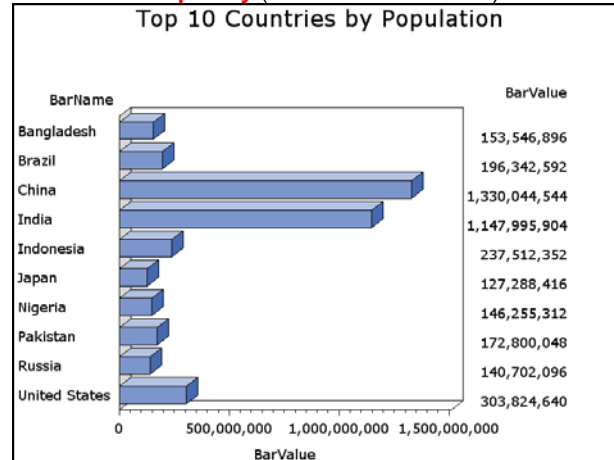
Solution

Figure 2: 2D Pie Chart. Simpler is Better.
Always Presents Accurate Relative Size of Slices.

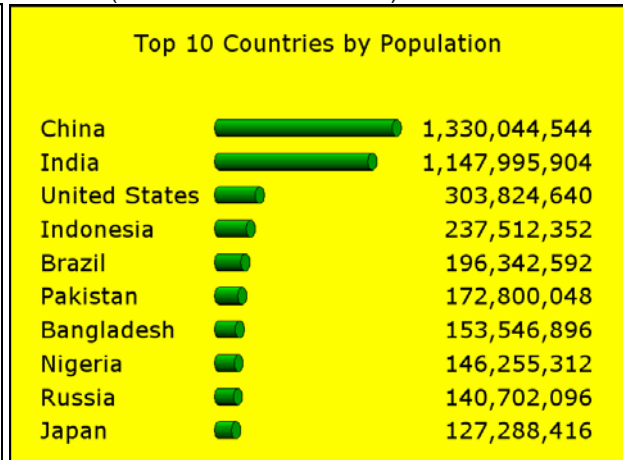


Having made the case against 3D, let me look at possibilities for bar charts. Several years ago, when I was responsible to create a proof of concept for presentation to executive management, I decided to see what I could do to pacify the alleged universal enthusiasm of executives for glitzy graphics, including 3D. I amazed myself by finding that though the standard SAS/GRAPH 3D bar chart is needlessly complicated and can be hard to interpret, Safe 3D is possible for a SAS/GRAPH bar chart. Here, again, I am pasting in previously published images. **See Appendix J for code to create a Safe 3D Horizontal Bar Chart.**

Needless Complexity (default 3D hbar chart)



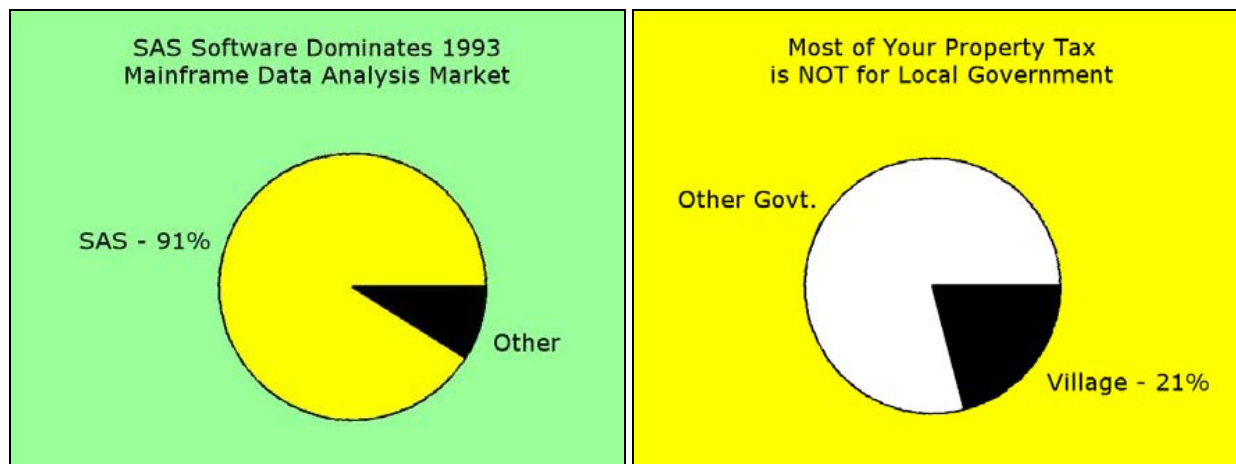
Safe 3D (customized 3D hbar chart)



In the next section, more value is added to the communication-effective Safe 3D SAS/GRAPH horizontal bar chart, and I show how to create one with "simulated 3D" using ODS Graphics. They can serve as pie chart alternatives.

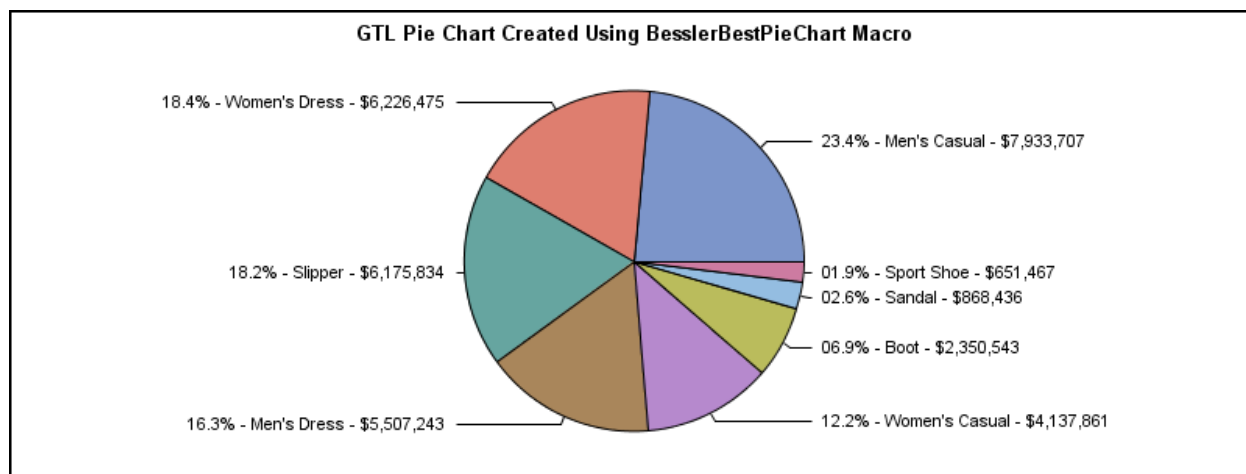
Now, I want to share two of my favorite 2D pie charts. I call this design “The Pac-Man Pie Chart”. These were done with SAS/GRAPH, but you should be able to adapt my pie chart code in Reference 1 to do one with ODS Graphics, which does not support pie charts directly (at this time), but requires you to go through the needlessly more cumbersome route of using Graphics Template Language.

Normally, I recommend NOT using an “Other” category in pie charts or bar charts. Using “Other” just leaves you open to the question of “What is IN ‘Other’?” I call this design “The Extremes of Other”. In the left-hand pie chart, in my opinion, what is in Other is too small to be of any significance. The right-hand pie chart, developed during my time as a local elected official, was intended to direct the concern of people complaining about their property tax bill toward other public bodies which were also getting a share (in some cases, a much bigger share) of taxes that had to be collected by our Village Treasurer.



For code to create the above charts, see “Easy, Elegant, and Effective SAS Graphs: Inform and Influence with Your Data” available at <http://www.lexjansen.com/phuse/2005/ps/ps05.pdf>

Below is a more typical pie chart, done with ODS Graphics and GTL in Reference 1. Please find the code (using a macro) in that paper. This was my “Best” ODS Graphics pie chart as of 27 June 2012. It is a GTL pie chart because there is no Statistical Graphic procedure in ODS Graphics. ODS Graphics Template Language must be used.



A fundamental problem with pie charts is the possibility that labels can collide, especially if you have several small slices and/or you want to use a larger font. Normally, I advocate against lumping small slices into “Other”. So that is not a solution. Several years ago, I mastered the task of creating a SAS/GRAPH pie chart with a more usable legend than the software supports. (I.e., my legend entries include slice description, value, and percent of whole, but the SAS/GRAPH legend omits percent of whole.) Developed fresh for this conference is a new macro to create a pie chart with a legend using ODS Graphics, which, as I noted above, is less friendly to pie charts. There is a false claim by some statisticians that pie charts are hard to interpret because comparing angular measures is more difficult than comparing linear measures. The Bessler style of slice label above or legend below removes the ambiguity, and, as I have emphasized from the outset, you need to provide both image and precise numbers to make reliable inferences.

Depending on the dimensions of your target image file and the font sizes used for chart title and the legend text, ODS Graphics might or might not draw the legend. When the legend cannot fit, you will see this message in the SAS log:

NOTE: Some graph legends have been dropped due to size constraints. Try adjusting the MAXLEGENDAREA=, WIDTH= and HEIGHT= options in the ODS GRAPHICS statement.

Above WIDTH and HEIGHT are the dimensions of the image file, for which the defaults are 640px and 480px. While developing my macro, I learned several things.

Tuning WIDTH and HEIGHT to get the legend to fit is an iterative process, especially since I prefer to NOT make the image unnecessarily large—after picking some dimensions to get the legend to appear, I then progressively decreased the dimensions to the minimum.

Since a graph developer is likely to have a preferred image size, either to fit in the expected web browser live area (that which you view without scrolling), or to be a size nicely compatible with a Word document or a PowerPoint slide and not have to be shrunk by the Microsoft Office Insert Picture process, it is best to keep the image dimensions constant, and tune the legend size with respect to the available space while selecting font sizes to one's satisfaction. This requires use of the MaxLegendArea parameter, which is defaulted to 20 percent of the image area. The value assigned omits use of any percent sign or abbreviation like PCT, which is used for other percent-based parameters.

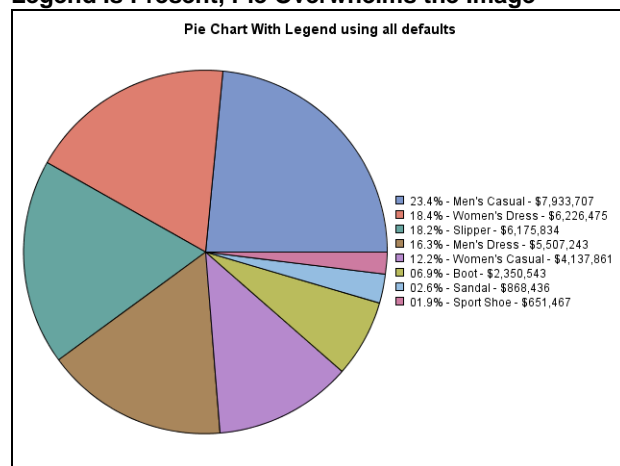
There is no direct control of the pie size. ODS Graphics tends to make it the maximum diameter that will fit within the image area left after drawing the chart title. The legend is drawn last, if it fits. (Over the course of testing, there were situations where extra white space WAS left around the pie, but the conditions to trigger that are unpredictable.) The direct and reliable way to get some extra white space between the title and the pie is to use the PAD option on the GTL statement used to specify the chart title. You can also use a PAD option on the LAYOUT statement to get some extra white space between the pie and the bottom margin of the image.

There are several ways to specify text font size. My testing found use of pt (point sizing) to cause anomalous behavior with respect to whether or not the legend would be drawn depending on the image size. Also, the drawn size of the text for a fixed point size appeared to sometimes be dependent on the image file dimensions. 16pt should be 16pt, irrespective of the dimensions of the image. Other choices include inches, millimeters, centimeters, and pixels. The default unit for font sizing is pixels. PCT is the unit that proved consistently satisfactory (even if the drawn percent of image height might not look like the percent specified). By “satisfactory” I mean that no “legend appear, then disappear, then reappear” phenomena occurred as I progressively increased image HEIGHT.

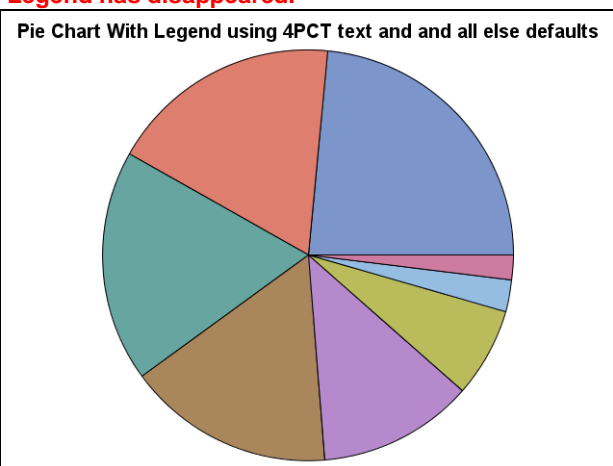
Below are pictures of pie charts (shrunk to fit them side by side between the margins of this Word document), first using the default ODS Graphics image size, the default GTL title size, and the default GTL legend text size, and then testing with 4PCT text size and various image dimensions and vertical spacing adjustments. One objective is to get a pie with a diameter comparable to the height of the legend block. Another option explored is pushing the pie and legend up the image area. After insert to Word or PowerPoint, the useless white space at the bottom can be trimmed.

NOTE: The ODS default image width of 640px, when inserted in a Word document with 1- inch margins for American Letter Size Paper (8.5 inches by 11.0 inches), is shrunk only to 97% of actual size.

Legend Is Present, Pie Overwhelms the Image



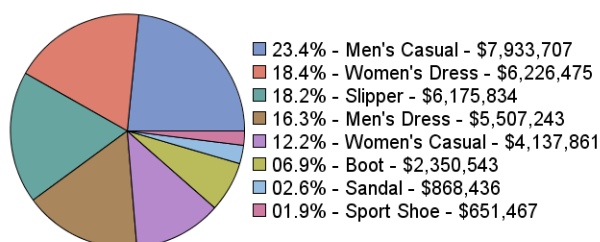
Legend has disappeared.



The macro provides five parameters to be adjusted: ImageWidth, ImageHeight, MaxLegendArea, ExtraSpaceAbovePie (between it and the title), and ExtraSpaceBelowPie (between it and bottom of the image). The macro defaults for image sizing are the same as ODS Graphics defaults of 640px and 480px. The ODS Graphics default for MaxImageArea is 20 (percent). Extra vertical space defaults are inherently 0. I have not tried to investigate whether there is any way to adjust horizontal spacing. On this page are various scenarios using the default image dimensions. On the next page are scenarios using the minimum image dimensions that will fit the legend.

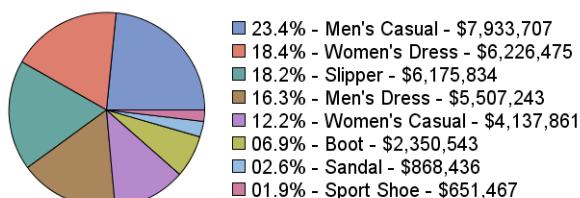
Desired Font Size, Default Image Size, Legend OK

Pie Chart With Legend using 4PCT text, ImgWdth=640px, ImgHgt=480px, MaxLegendArea=24



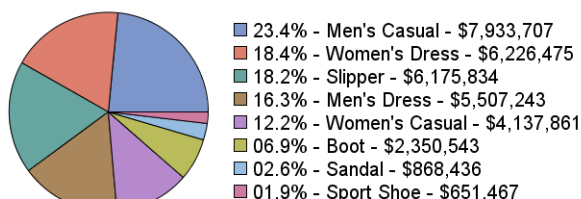
Pie Diameter Approximates Height of Legend Block

Right-Sized Pie - 4PCT text, default image height & width With 100px extra white space above and below pie



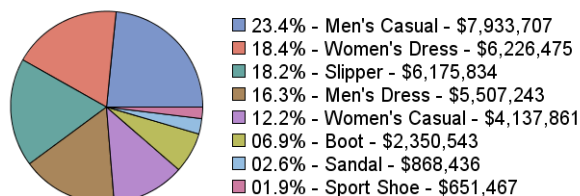
Shorten title & increase space above & below the pie

Right-Sized Pie - 4PCT text, default image height & width



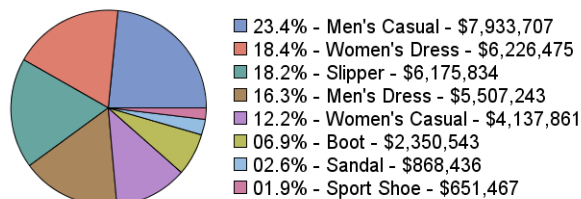
Adjust spacing to move the pie up

Pie Chart To Trim - 4PCT text, ImgWdth=640px, ImgHgt=480px, MaxLegendArea=24, ExtraSpaceAbovePie=10px, ExtraSpaceBelowPie=170px



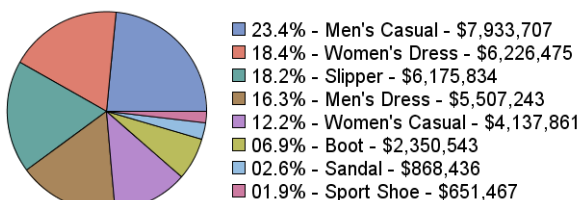
Shorten title & increase space below the pie

Pie Chart To Trim - 4PCT text, default image height & width With 20px more white space below pie

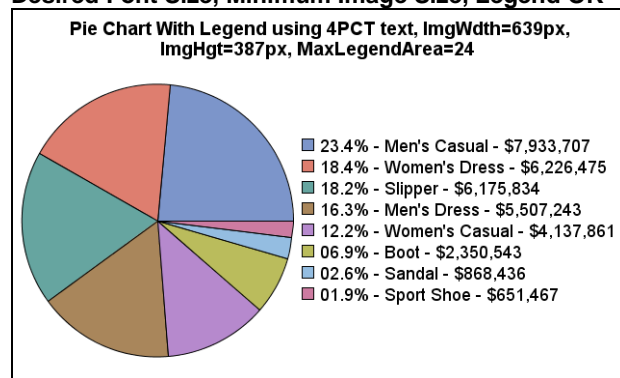


Shorten title & adjust space above & below the pie

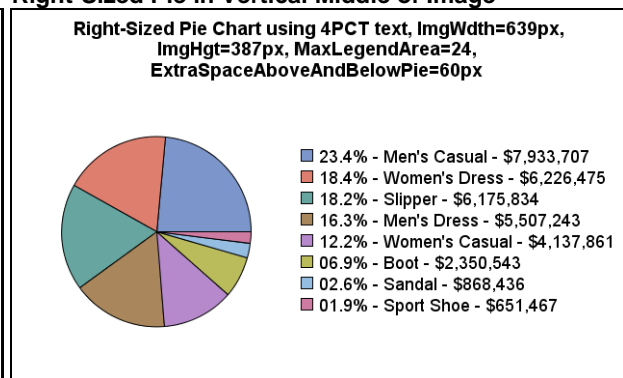
Pie Chart To Trim - 4PCT text, default image height & width



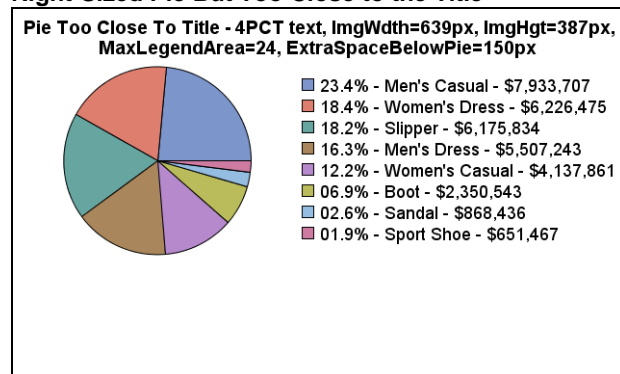
Desired Font Size, Minimum Image Size, Legend OK



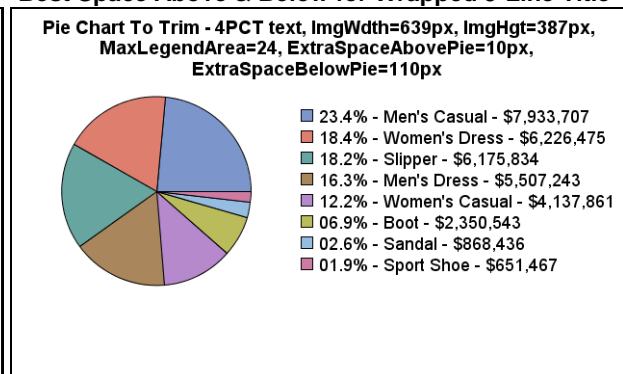
Right-Sized Pie in Vertical Middle of Image



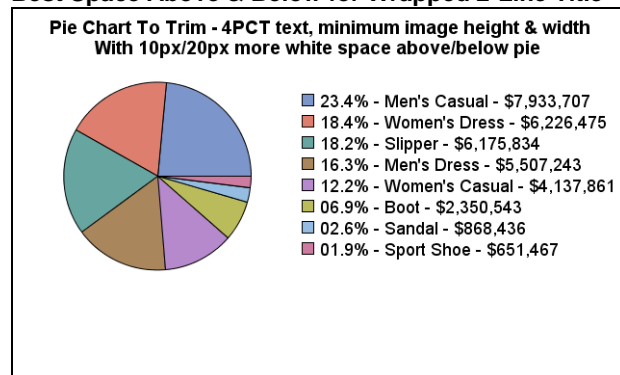
Right-Sized Pie But Too Close to the Title



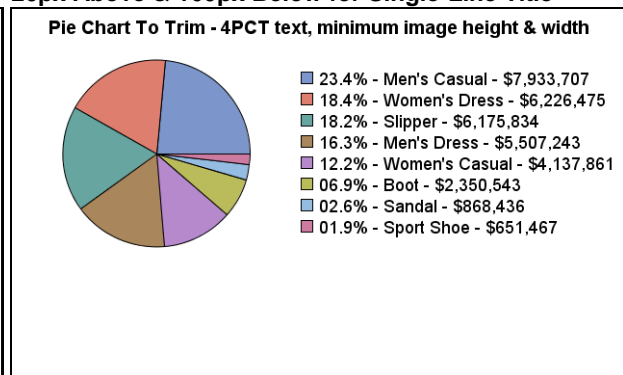
Best Space Above & Below for Wrapped 3-Line Title



Best Space Above & Below for Wrapped 2-Line Title



20px Above & 160px Below for Single-Line Title



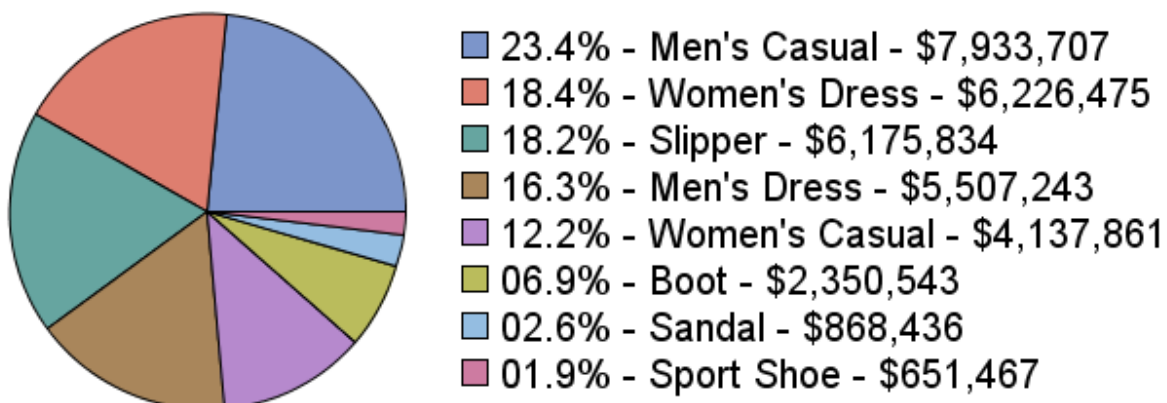
In the twelve examples above, you see the process of seeking an optimum design, first with default image size, and then with minimum size for which GTL will deliver a legend at my preferred text size. Not everyone wants a chart that is to be trimmed at the bottom after insert in Word or PowerPoint. A vertically centered pie of a size comparable to the legend might be good enough. For some users an oversized pie might be OK. Even getting to the minimum value of MaxLegendArea that will deliver the legend might not be important to some users. If the ODS default value delivers the legend, the graph creator is unlikely to even think about optimizing MaxLegendArea. The macro used above allows you to do as much or as little tuning as you desire or as the intended user of the pie chart demands.

NOTE: In the shrunk images above, the legend color samples are difficult to distinguish and match with their slices. You can guess the association based the ordering (of slices and legend entries) and slice size. If the images were inserted here and left to fill the page width available, color distinguishability would not be a problem. On the next page are nearly full-size pie chart examples, which support that assertion. However, prior to Version 9.4 (used to create all of the pie chart with legend examples in this section), the color samples are generated so small as to create a distinguishability problem. The new AUTOITEMSIZE option, which matches the height of the legend color sample to the height of its text description, solves the color distinguishability problem, IF the legend text is of adequate size.

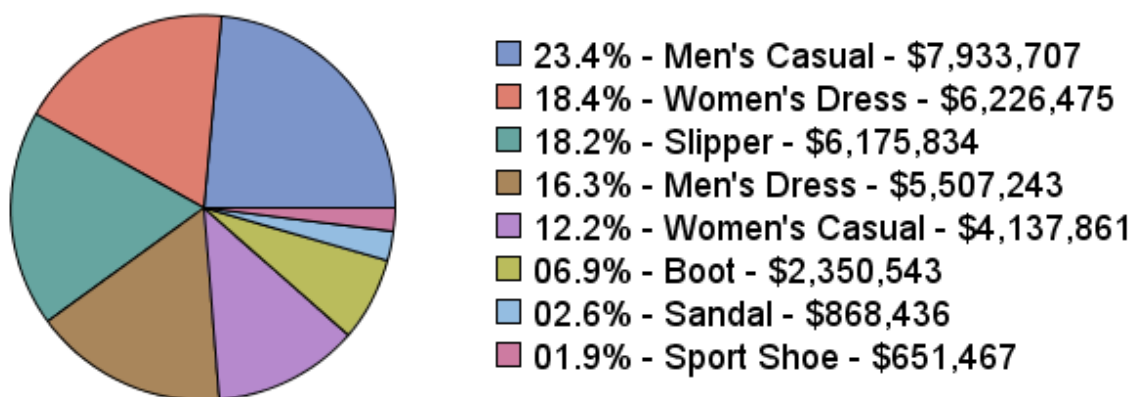
The code used to create the pie charts on the preceding pages (and the next page) is in Appendix J.

The two image files inserted on this page were shrunk, respectively, only to 97% and 98% of actual size. The legend color samples are easily distinguishable.

**Right-Sized Pie using 4PCT text, default image height & width
With 100px extra white space above and below pie**



**Right-Sized Pie Chart using 4PCT text, ImgWdth=639px,
ImgHgt=387px, MaxLegendArea=24
With 60px extra white space above and below pie**

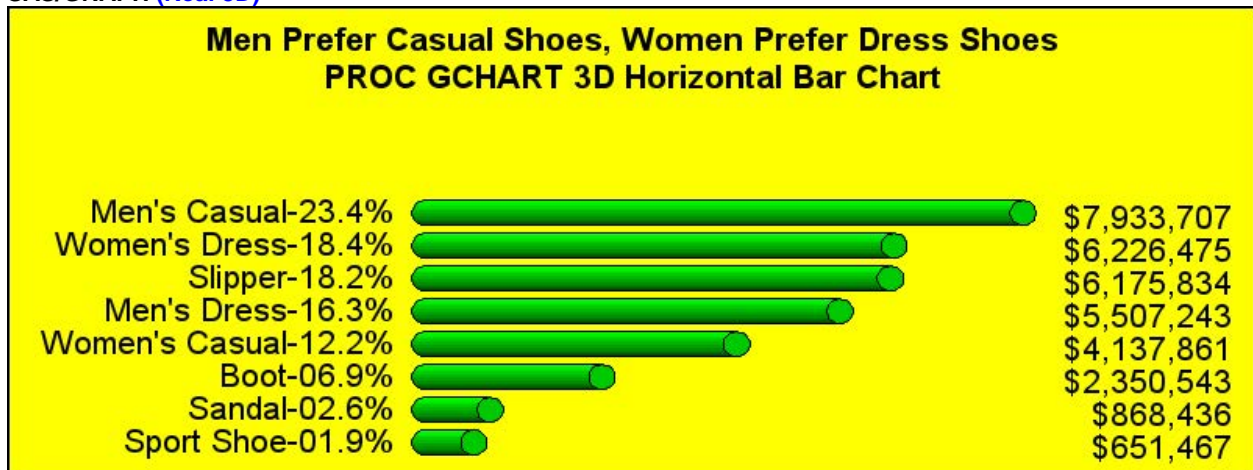


K. WHEN A PIE CHART WILL NOT WORK, BUT YOU WANT CATEGORY, VALUE, & PERCENT OF WHOLE

If you have a pie chart which has problems fitting in slice name, value, and percent of whole, or you don't want to create one with a legend, either of the bar charts below is an adequate substitute, but, regrettably, they cannot visually show the comparative percent of whole, which is the unique power of the pie chart.

Both examples pre-process the original input to add percent of grand total to the bar label.

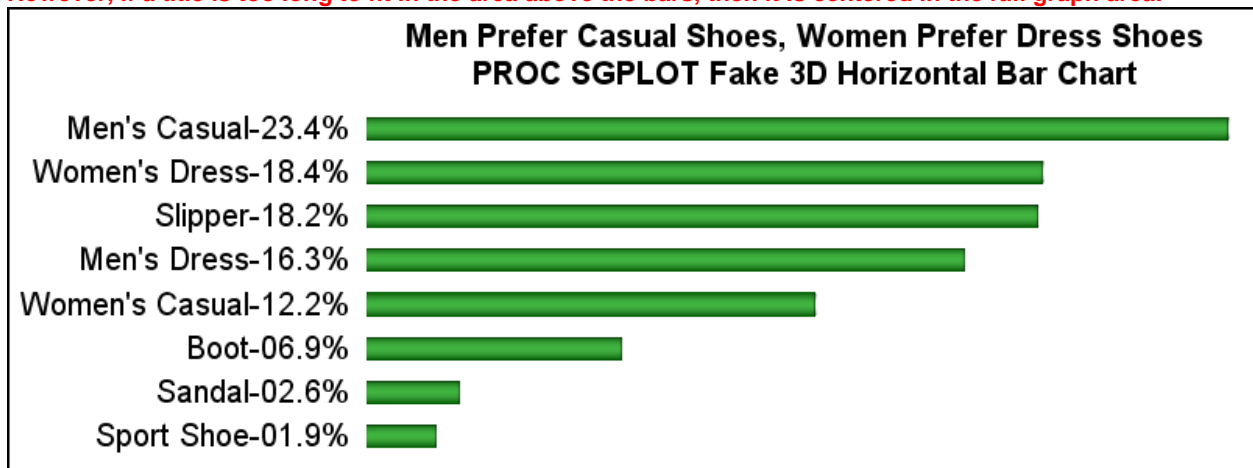
SAS/GRAPH (Real 3D)



ODS Graphics ("dataskin=pressed" option on the hbar statement to simulate part of 3D effect)

Note that the titles are centered over the bars, rather than in the full graph display area.

However, if a title is too long to fit in the area above the bars, then it is centered in the full graph area.

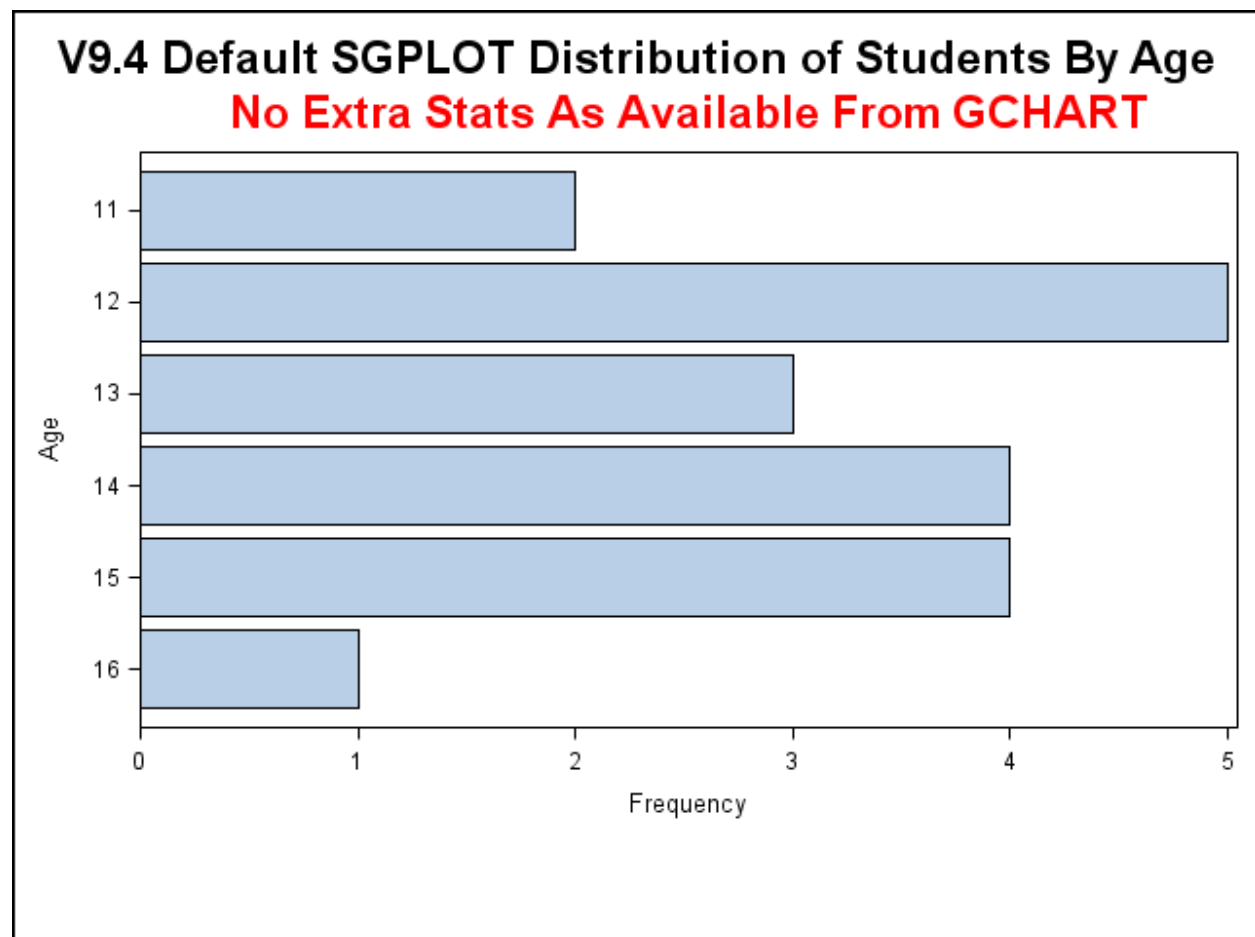
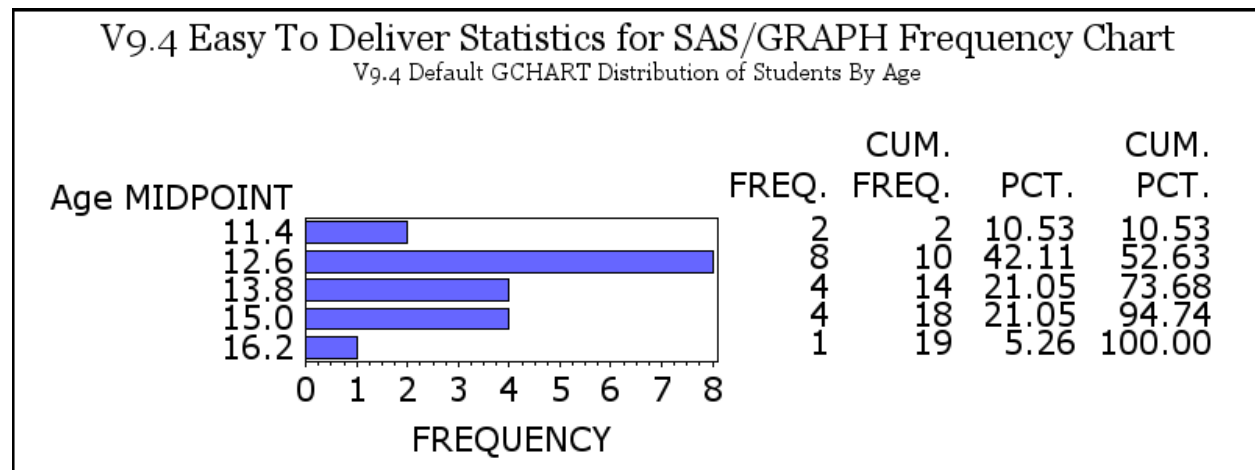


See Appendix K for the code used to create the above graphs.

L. HORIZONTAL BAR CHARTS FOR FREQUENCY DISTRIBUTIONS: ODS GRAPHICS VS. SAS/GRAPH

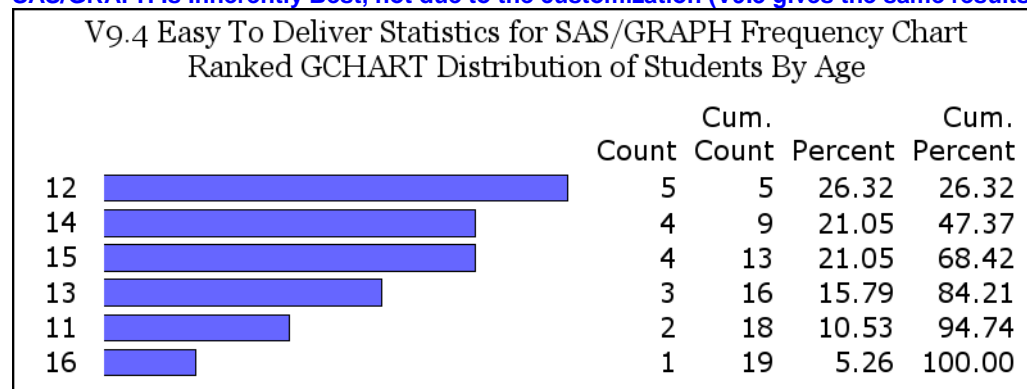
There are significant differences in ODS Graphics and SAS/GRAPH when used to graph frequency distributions.

First, let's compare what you get with no special coding effort. The height of the SAS/GRAPH image was reduced from the default Y pixels. Both of these results are inelegant, but the point is that SAS/GRAPH delivers MORE, even when accepting the defaults.

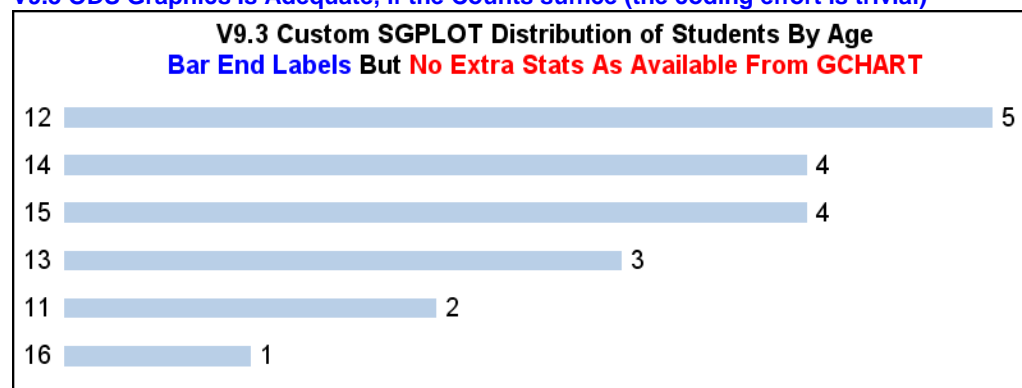


On the next page is a comparison of customized results. Not much can be easily added with ODS Graphics.
Please see Appendix L for the code used to create all of the graphs in this section.

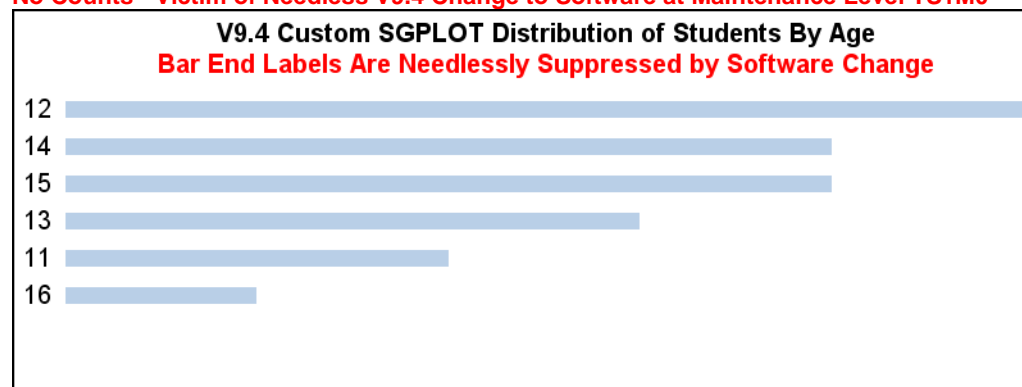
SAS/GRAPH Is Inherently Best, not due to the customization (V9.3 gives the same results)



V9.3 ODS Graphics Is Adequate, if the Counts suffice (the coding effort is trivial)

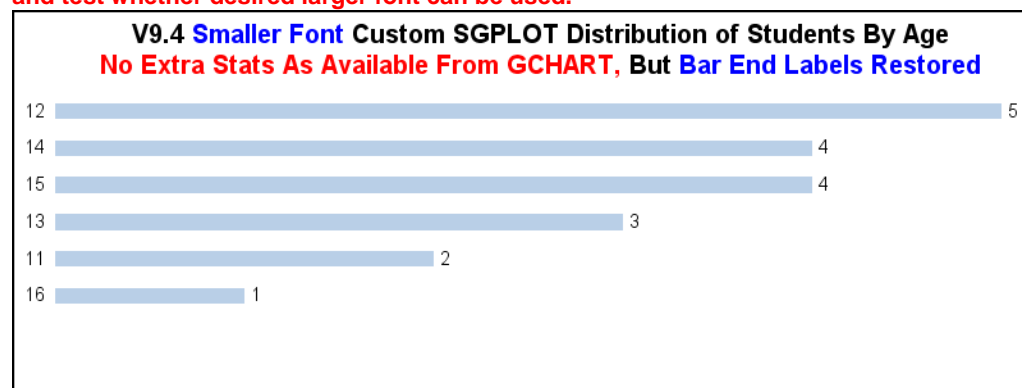


No Counts - Victim of Needless V9.4 Change to Software at Maintenance Level TS1M0



Circumvent Needless Software Change with Smaller Font

Alternative: upgrade to TS1M1, use new datalabelfitpolicy=none option on HBAR statement, and test whether desired larger font can be used.



M. IMAGE PLUS PRECISE NUMBERS, EVERYTHING TO COMPARE CATEGORICAL DATA, WITH RANKING AND SUBSETTING TO SHOW THEM WHAT'S IMPORTANT

Unless you are creating a comprehensive lookup tool, e.g., to report some measures for every province in a nation, you do not necessarily need to show ALL of the data. In a lookup display, you will probably order character keys in alphabetical order or numerical ID's in numeric order.

But even for the lookup tool, your users might want an alternative view that ranks the categories by some measure of interest. That enables you to "Show Them What's Important". Furthermore, your users might not want to see EVERY category. Subsetting the input enables you to "Let Part Stand for the Whole"—a communication concept, I believe, first mentioned by Jim White.

There are three ways to subset data:

- (a) cutoff: all values above or below some specified value
- (b) Top N, be it Top 3, Top 10, Top 50, or whatever value of N suits your preference
- (c) enough of Top ranked categories to account for P percent of the grand total, be it Top 80%, Top 90%, etc.

I have been working this concept since 1987, when I first applied it to tabular reporting. For tabular reporting examples and a macro to provide all three varieties of subsetted ranking reports in one of my earliest papers where I shared them, see "Effective and Efficient Information Delivery for Executive Management" published in the SUGI 1992 Proceedings and available at: <http://www.sascommunity.org/sugi/SUGI92/Sugi-92-20%20Bessler.pdf> In this same paper, I emphasized the value of horizontal bar charts in showing both magnitude and percent of whole.

The ultimate implementation of this concept was reported most recently at PharmaSUG 2013, where I delivered a web-enabled horizontal bar chart macro that interlinks all three varieties of subsetting and a bar chart of ALL of the categories. Furthermore, the macro does rightsizing so all subsets/views of the data are drawn with bars of the same width, and only the "ShowAll" view, described below, might require vertical scrolling to see all of the bars. (The natural disposition of the ODS Graphics hbar chart software is to FILL the allocated image height with as many bars as are available. To see how strange that looks, see illustrations on Page 18 of Reference 1.)

I also created a variation on that macro which will create a bar chart that, with as many bars as can fit, fills the between-margins display area of a Word document or fills a PowerPoint slide, for another form of rightsizing.

For this paper, I am borrowing from Reference 1, and ask you to please refer to that resource for a more comprehensive exposition of the topic and for all of the supporting code. Here, I will show you some example outputs from what I called "TOOL 5. THE UpTo4WaysRankingHbarChartsToHTML MACRO", which provides up to four ways to create a Subsetted and Ranked Horizontal Bar Chart (and deliver supporting detail for individual bars) with just one macro invocation. The macro is able to support various screen resolutions.

It creates up to four interlinked web graphs, depending on how you invoke it. You can specify any of the following options alone or in any combination:

TopN – for the Top N categories

TopPercent – for the minimum number of the Top categories to account for at least the specified percent of the grand total of the measure of interest

Min – all of the categories for which the measure of interest is greater than or equal to the specified minimum

ShowAll – list all of the categories in rank order

It, optionally, provides hyperlinked spreadsheets: (a) a link is added on the web page to a table of the final input used to create the chart (and that spreadsheet contains a link back to the web page); and (b) each bar is linked to a detail table for all of the variables in the data set and those rows that were summarized to provide input for that bar on the chart. If the data set was only one row per category value, then the raw detail tables for the bars are one-row tables.

Data tips, to present the bar label information (which contains rank, description, value, and percent of whole) in a pop-up box when the mouse is rested on any bar, are turned on if spreadsheets are requested, or if the web graphs are being created on a high resolution screen. On a high resolution screen, more bars can be accommodated, but, for large enough bar counts, a very small font size is used for the bar labels and they might be hard to read.

The macro sizes the web graphs (except for ShowAll) so that they fit in the browser window and need no scrolling.

The images can be sized for either of two alternatives: (a) the current maximum resolution of 1920 X 1080 (wide screen laptop monitors, designed, I think, for viewing movies); and (b) the most common older resolution of 1024 X 768 (suited to old style monitors with screen width-to-height proportions of 4 to 3, which still are the proportions of most conference room projection screens).

Though the Top N concept of ranking has a long tradition, the more rational ways to subset categories, in my judgement, are with the Top Percent or a minimum threshold.

Being able to “Show Them What’s Important” in three different ways, plus provide the option to see the whole picture, are a maximally effective way to visually communicate the significance of the data.

On the following pages are screen images from various scenarios.

When you examine the graphs, pay close attention to the dynamically (i.e., from dynamic code that reacts to the data, not from manual hard coding) created title lines which are designed to provide as much information as possible about the subset of the data, all of the data, and how much of the total that the subset accounts for.

Each graph is linked forwards and backwards to a spreadsheet of all of its input data, and, when the values for the bars are derived by summarization from more detailed data, each bar is linked to a spreadsheet of that data.

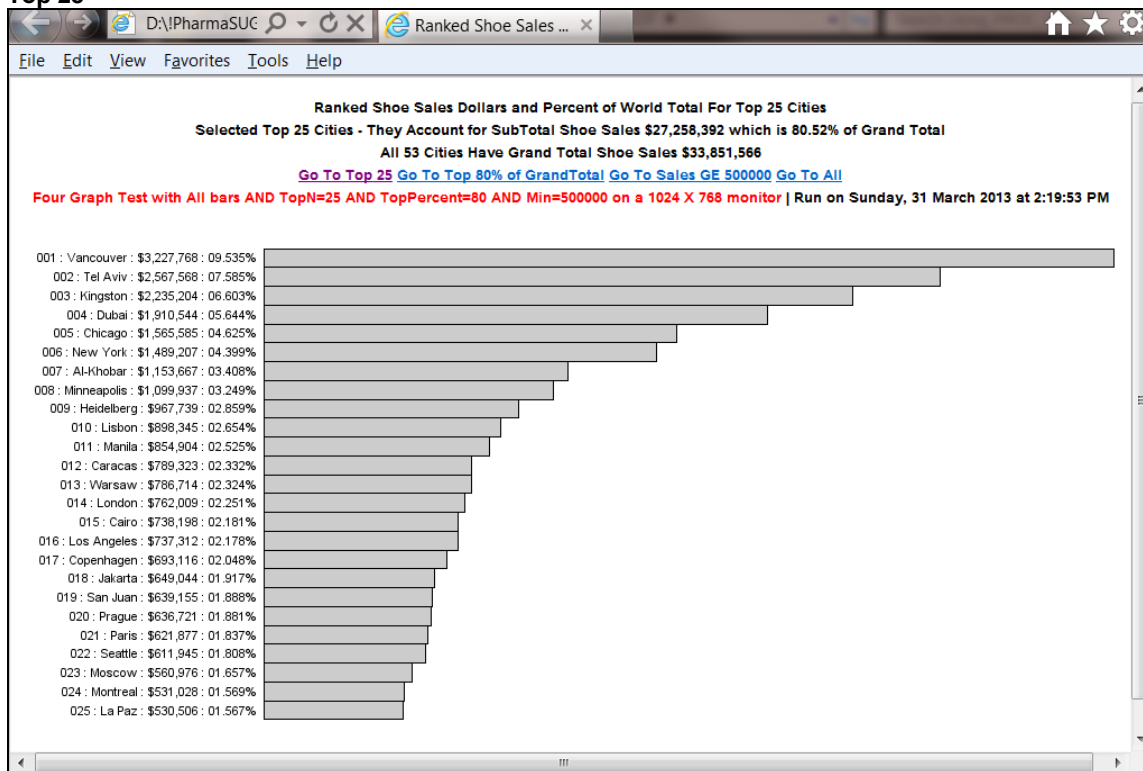
When a graph contains so many bars that its label is hard to read, then data tips are provided for the bars.

The bar labels are maximally informative: rank, description/category, value, percent of total of all of the bars present. In this way, this horizontal bar chart design really delivers a table, with an image to the right of every table entry to provide a visual comparison with all of the other table entries present.

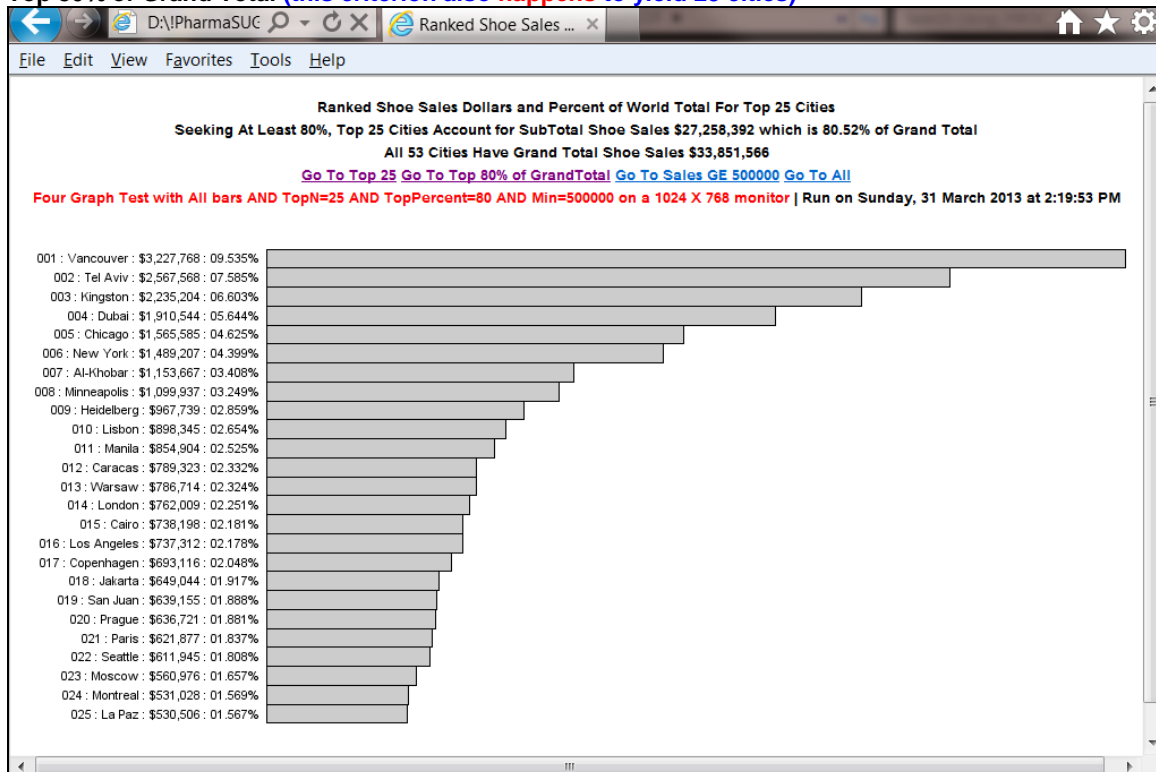
I cannot think of a more informative or more easily interpreted way to compare categorical measurement data.

NOTE: The macro requires at least SAS Version 9.3, and relies on use of SAS ODS Graphics. The code for the macro and invocations of it, including those used for graphs on the following pages, is in Reference 1. The code has not been re-tested with Version 9.4, and there might be some differences in results.

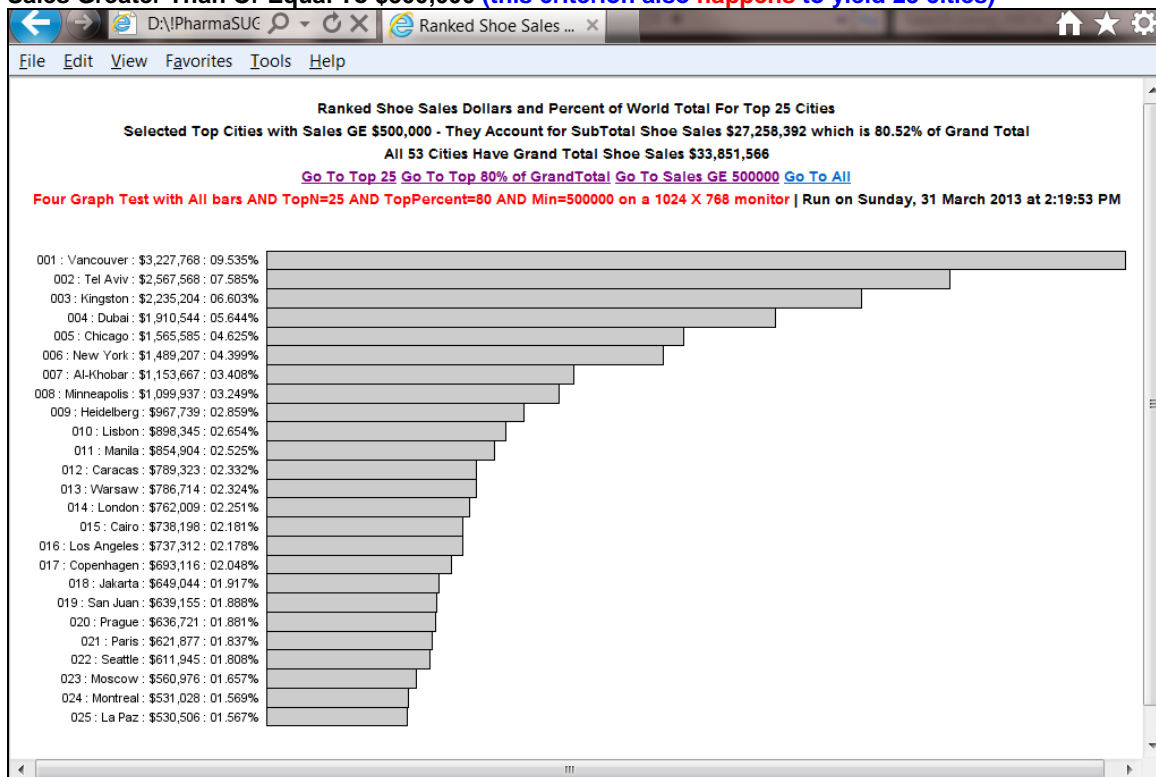
Top 25



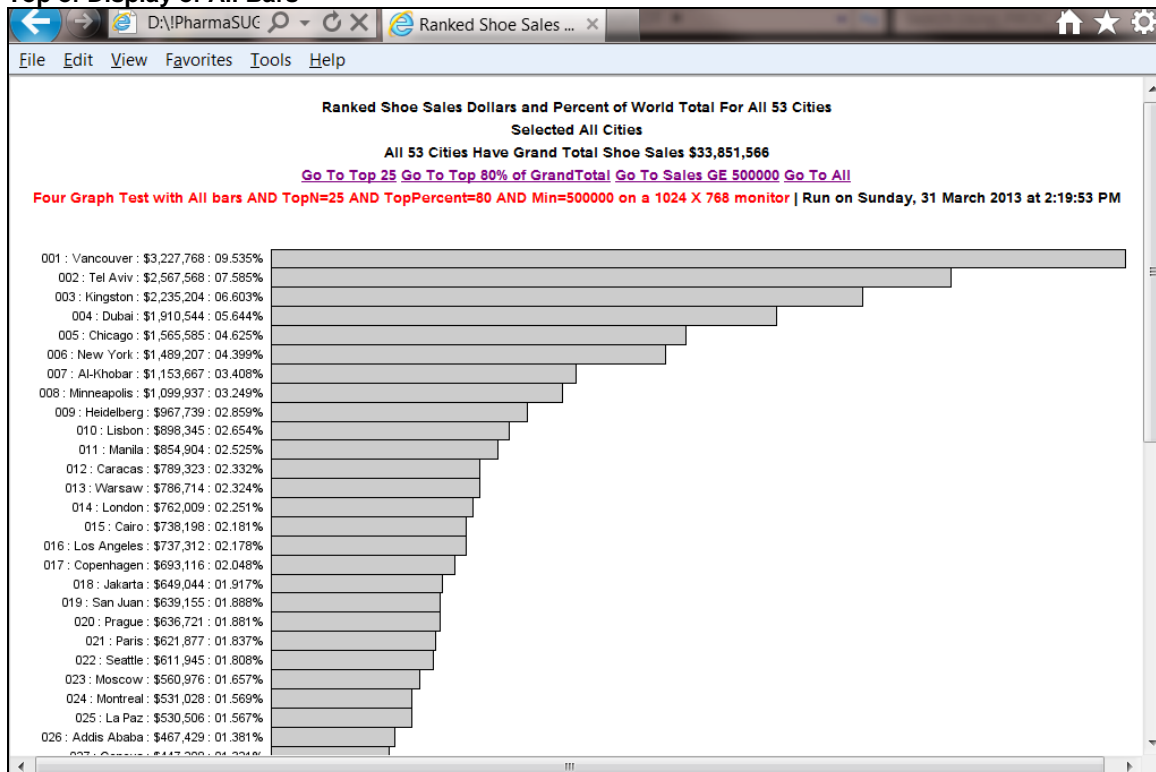
Top 80% of Grand Total (this criterion also happens to yield 25 cities)



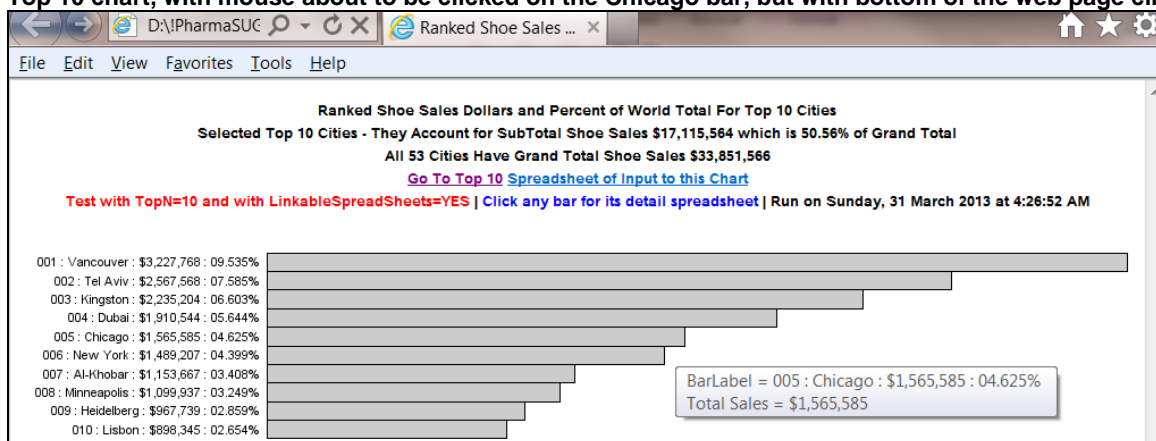
Sales Greater Than Or Equal To \$500,000 (this criterion also happens to yield 25 cities)



Top of Display of All Bars



Top 10 chart, with mouse about to be clicked on the Chicago bar, but with bottom of the web page clipped off



After clicking on the Chicago bar

IDX	Subsidiary	Region	Product	Stores	Sales	Inventory	Returns
1	Chicago	United States	Men's Casual	26	\$408,978	\$831,672	\$17,907
2	Chicago	United States	Slipper	13	\$329,235	\$1,049,530	\$10,654
3	Chicago	United States	Women's Dress	37	\$293,313	\$754,157	\$10,194
4	Chicago	United States	Men's Dress	28	\$261,607	\$824,485	\$8,915
5	Chicago	United States	Women's Casual	11	\$172,021	\$474,524	\$5,399
6	Chicago	United States	Boot	16	\$82,483	\$305,061	\$3,735
7	Chicago	United States	Sport Shoe	17	\$17,347	\$74,605	\$684
8	Chicago	United States	Sandal	2	\$601	\$17,121	\$35
9	Chicago	United States			\$1,565,585		

After clicking on the hyperlink in the subtitle of the web bar chart, here is the spreadsheet of its input data:

Subsidiary	BarLabel	Total Sales
Vancouver	001 : Vancouver : \$3,227,768 : 09.535%	\$3,227,768
Tel Aviv	002 : Tel Aviv : \$2,567,568 : 07.585%	\$2,567,568
Kingston	003 : Kingston : \$2,235,204 : 06.603%	\$2,235,204
Dubai	004 : Dubai : \$1,910,544 : 05.644%	\$1,910,544
Chicago	005 : Chicago : \$1,565,585 : 04.625%	\$1,565,585
New York	006 : New York : \$1,489,207 : 04.399%	\$1,489,207
Al-Khobar	007 : Al-Khobar : \$1,153,667 : 03.408%	\$1,153,667
Minneapolis	008 : Minneapolis : \$1,099,937 : 03.249%	\$1,099,937
Heidelberg	009 : Heidelberg : \$967,739 : 02.859%	\$967,739
Lisbon	010 : Lisbon : \$898,345 : 02.654%	\$898,345
		\$17,115,564

CONCLUSION

Graphs are intended to permit valid inferences from data and/or to guide decisions. Visuals accelerate the process, but reliable inferences and decisions still need to see the precise numbers. This paper embodies a collection of communication-effective design principles, but the main thrust of the examples is to show you various ways to deliver both image and precise numbers. All of the supporting code is either (mainly) in the Appendices or in references cited. **Appendix A contains an important discussion of recent developments that affect accessibility to data tips.** Appendices B through L are available via email, either as a PDF, or as a zip file of SAS programs.

REFERENCES

1. Bessler, LeRoy. 2013. "Data Visualization Power Tools: Expedite the Easy, Implement the Difficult, or Handle Big Data" in PharmaSUG 2013 Conference Proceedings. Chicago, Illinois: Pharmaceutical SAS Users Group. Available at: <http://www.pharmasug.org/proceedings/2013/DG/PharmaSUG-2013-DG11.pdf>

ACKNOWLEDGEMENT

My thanks to Laura MacBride for her review of an early draft of this paper, and to Lelia McConnell, and others too numerous to mention, in SAS Technical Support who have helped me over the years in my quest to create the exact graphic output that I sought.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Also, code for Sections B to L (packaged as Appendices B to L) is available from the author via email, either as a PDF, or as a zip file of SAS programs. (Recall that code for Section M is in Reference 1.)

Contact the author at:

LeRoy Bessler PhD
Bessler Consulting and Research
Mequon, Milwaukee, Wisconsin, USA
Le_Roy_Bessler@wi.rr.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

APPENDIX A. Data Tips (MouseOver Text) with Internet Explorer Version 11 or Google Chrome

The web-enabled graphs in this paper are designed to display what is variously called tool tips, data tips, hover text, mouseover text, flyover text, floatover text, etc., but was originally called ALT text, for “alternative text”. Its original purpose was to provide descriptive text for images that could be converted to audio by screen reader software for visually impaired users of web browsers who could not see the image. Its function was expanded to provide a pop-up box of the text that could be read by users with no visual impairment.

For purposes of this paper, I will call this pop-up text “data tips”, which is the verbiage used in ODS Graphics.

ALT text was enabled in two ways:

(1) In an HTML tag, which is used to display an image file on a web page, an tag might look like:

(2) In the HTML imagemap, which defines all of the mouseover hot spots to bring up data tips, each spot was defined with an <area> tag, which might look like:
<area shape="poly" alt=" Name = Alfred Age (Sum) = 14" coords="104,41,559,41,559,58,104,58,104,41,"/>

Prior to Internet Explorer Version 11, this worked fine. In the SAS/GRAPH coding that feeds the imagemap, alt= must be replaced with title= for the data tip to be opened with the mouse. However, since I don't have access to an Internet Explorer version prior to 11, I do not know whether data tips for web pages using title= in the imagemap will open.

A further problem with IE 11 is that, by default, it does not open the imagemap hot spots. The default View Zoom setting is 125%. You need to do two things. On the Menu toolbar, click View, and then set Zoom to 100%. On the Menu toolbar, click Tools > Internet options > Advanced. On the Advanced tab, under the Accessibility section, Uncheck “Reset zoom level for new windows and tabs”.

For SAS/GRAPH, you code alt= (or title=) HTML in your program and deliver it with an HTML parameter on the graph creation statement in the PROC step, where HTML=<a SAS variable that contains the HTML code for alt= or title=>.

For ODS Graphics, you simply request the creation of an imagemap on the ODS GRAPHICS statement, but in Version 9.4 there are three new parameters that allow you to specify which data set variables you want to see in the data tip box and to control their SAS format and label text. For use on the graph creation statement, such as HBAR, V9.4 provides three new options:

TIP=(variable list)
TIPFORMAT=(format list)
TIPLABEL=(label list)

For detailed information on how to use them, see the Online Documentation.

NOTE: Not mentioned in the documentation is that for numeric variables, if you wish to use formats such as 2. or 4.1, you must instead specify Best2. or F2. and Best4.1 or F4.1 in order to avoid an error. When the software is fixed to accept the conventional, simpler numeric formats, you will not need to use the Best or F prefix.

With SAS/GRAPH, the programmer must code a DATA step that adds a variable which is coded to create HTML that will perform analogous functions.

Below is a comparison of creating data tips with ODS Graphics and with SAS/GRAPH:

```
goptions reset=all;
ods noresults;
ods listing close;
ods graphics on / imagemap=on imagename='SGSCATTERplotWithDataTips';
ods html path="D:\! SAS Global Forum 2014\Results\A\ODS Graphics" (url=none)
  gtitle gfootnote style=Minimal
  body="V94 ODS Graphics SGSCATTER Plot With Data Tips.html"
  (title="V9.4 ODS Graphics SGSCATTER Plot With Data Tips");
proc sgscatter data=sashelp.class;
title "V9.4 ODS Graphics SGSCATTER Plot With Data Tips";
plot weight*height /
  tip=(name age sex weight height)
```

```

tipformat=($16. F2. $1. F5.1 F4.1)
tiplabel=('Student' 'Age' 'Gender' 'Weight in Pounds' 'Height in Inches' );
run;
ods html close;
ods listing;

data Work.ToPlotUsingTITLEforDataTips;
length htmlVarForDataTips $ 200;
set sashelp.class;
htmlVarForDataTips =
  'title='
  || 'Student: ' || trim(left(name))
  || '0D'X /* force a line break in the TITLE text display */
  || 'Age: ' || trim(left(put(age,2.)))
  || '0D'X /* force a line break in the TITLE text display */
  || 'Gender: ' || trim(left(sex))
  || '0D'X /* force a line break in the TITLE text display */
  || 'Weight in Pounds: ' || trim(left(put(weight,5.1)))
  || '0D'X /* force a line break in the TITLE text display */
  || 'height in Inches: ' || trim(left(put(height,4.1)))
  || '';
run;

proc catalog cat=work.gseg kill;
run; quit; /* to assure that reruns during this session
           will not add a suffix to image file name */

ods noresults;
ods listing close;
ods html path="D:\! SAS Global Forum 2014\Results\A\SASGRAPH" (url=none)
  gtitle gfootnote style=Minimal
  body="V94 GPLOT Plot with Data Tips.html"
  (title="V9.4 SAS/GRAPH GPLOT Plot with Data Tips");
proc gplot data=Work.ToPlotUsingTITLEforDataTips;
title "V9.4 SAS/GRAPH GPLOT Plot with Data Tips";
plot weight*height / html=htmlVarForDataTips name='GPLOTplotWithDataTips';
run; quit;
ods html close;
ods listing;

```

The Google Chrome web browser does not recognize alt=, and requires it to be replaced with title=, both in the tag and in the <area> tag.

ODS Graphics Version 9.4 creates <area> tags with the text description assigned using both alt= and title=, but uses only alt= for the tag. SAS/GRAPH also uses only alt= for the tag.

With Google Chrome, a hyperlink in a spreadsheet of input data back to the graph built from that data does not work.