Paper 231-2010

The Most Communication-Effective and Most Usable Information Delivery

LeRoy Bessler, Bessler Consulting and Research, Fox Point, Milwaukee, Wisconsin, USA
Le_Roy_Bessler@wi.rr.com

Abstract

Graph = quick, easy inference
Precise Numbers = correct inference
Excel = data recipient option for self-directed post-processing

Data created with SAS® often must be packaged in a spreadsheet for users who want flexibility to manipulate it further with Excel. You can tap the power of SAS/GRAPH® and ODS to deliver data in a web-enabled graph linked to a SAS-created spreadsheet. The web-enabled graph includes a hyperlink to the spreadsheet, which is hyperlinked back to the graph.

This paper focuses on the example of trend plot data. The web-enabled graph provides pop-up text so that the user can read off EXACT values for plot points, rather than estimate them, without being forced to jump to the spreadsheet. Annotating a plot with precise values is not practical when points are dense or the plot is multi-line.

This solution works for any graph, not just a plot, and avoids trying to fit a graph and table on the same page. The HTML, graph, and spreadsheet files can be stored on local disk, a LAN file server, or a web server. They can be emailed as attachments, possibly zipped first. (FYI, SAS can be used to send email.)

This paper is for new or experienced users of Version 9.1.3 or Version 9.2. No prior knowledge of SAS/GRAPH or ODS is assumed.

Basic Solution

I will show you how to package your results so that the user can view it as a maximally communication-effective web graph, but can toggle to a spreadsheet to inspect the supporting data as a table, with the option to manipulate it further non-graphically with Excel. For someone wanting data in Excel, but appreciating a better and finished graph, this is the best deliverable.

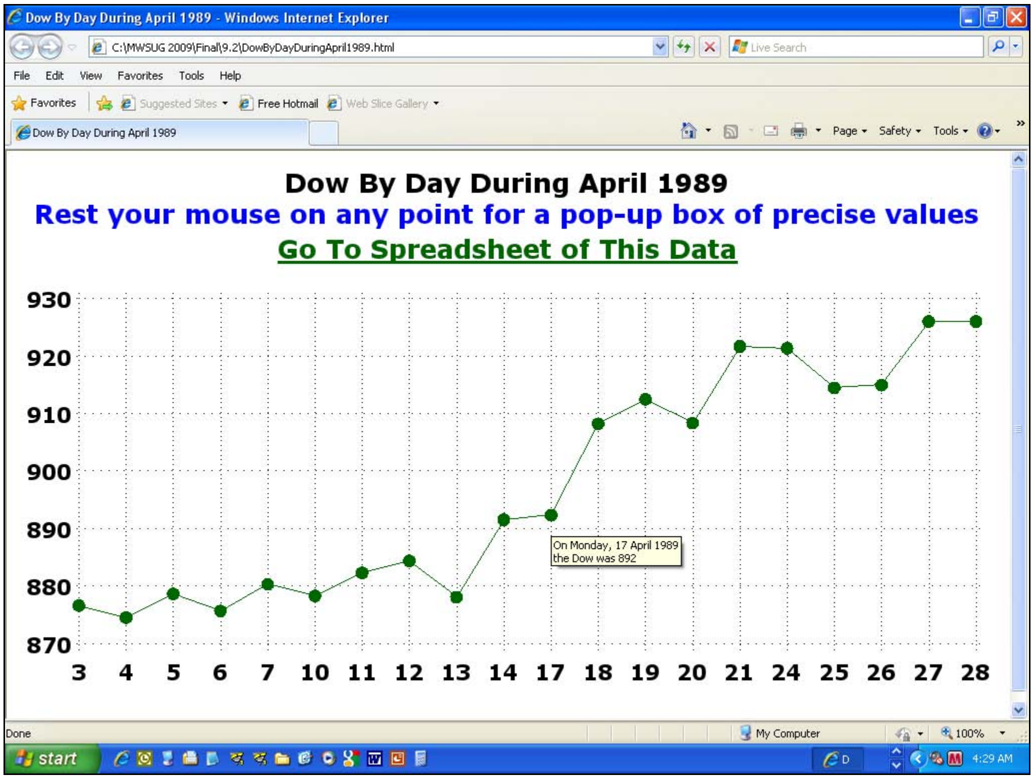
Web deployment permits imbedding hyperlinks in a graph, plus ALT text (also variously called pop-up text, flyover text, floatover text, hover text, or a tool tip). A graph most likely to benefit from ALT text is a plot. Even with tick marks and/or reference/grid lines, it is impossible to know the precise coordinates for a plot point. With ALT text, simply resting the mouse on a plot point displays its x and y values to any precision you choose to provide, in any format you like, with whatever description you like for the x and y variables, and even with extra information. For bar charts and pie charts (where, IF you use my design for them, ALT text is never necessary to get at the precise detail), please see my other work.

Since the hyperlink is inside the graph area, a visit to the link does not change the color displayed inside the graph, upon return to the graph.

Version 9.2 was used here. The results for 9.1.3 would be identical except that the fonts would be smaller and not be as thickly drawn, and the plot symbols would be smaller. If you do not specify DEVICE=, then ODS HTML in 9.1.3 selects GIF, but in 9.2 selects PNG, which does not support transparency—which can be used to let the web page background (which here is plain white, rather than colored) show through the graph. Also, Version 9.2 permits thickening reference lines. However, line type 33 (used here) becomes very thick short dashes at a thickness of 2.

The structure of the main body of this paper is as follows. The graph and spreadsheet are displayed with blow-ups of two features and a very few comments. The displays should be self-explanatory. The complete code is presented with copious imbedded comments, and then the same code is presented without any comments. An alternative graph design that I prefer is displayed, followed by needed code changes. Then the paper closes with bibliographic and author contact information.

Here is the graph web page, with the mouse resting on a point:


Dow By Day During April 1989 - Windows Internet Explorer
C:\MWSUG 2009\Final\9.2\DowByDayDuringApril1989.html
File Edit View Favorites Tools Help
Favorites
Dow By Day During April 1989
930
920
910
900
890
880
870
3 4 5 6 7 10 11 12 13 14 17 18 19 20 21 24 25 26 27 28
Done
My Computer
100%
start
4:29 AM

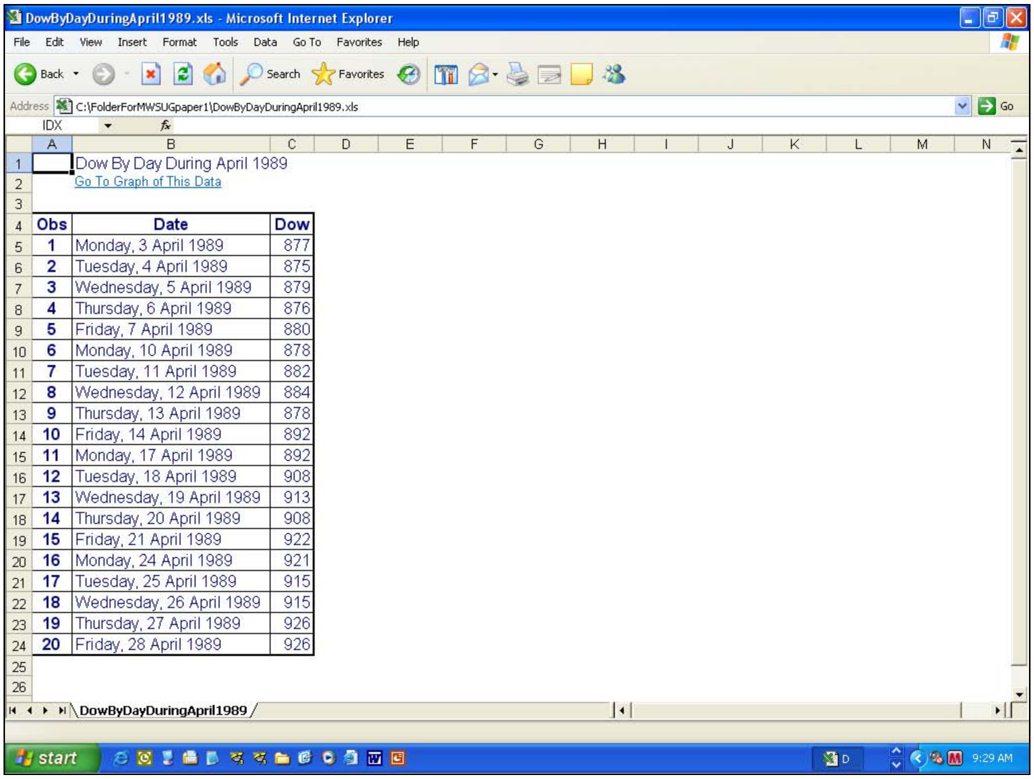
Dow By Day During April 1989

Rest your mouse on any point for a pop-up box of precise values

Go To Spreadsheet of This Data

On Monday, 17 April 1989 the Dow was 892

Here is the spreadsheet after being linked to from the graph:

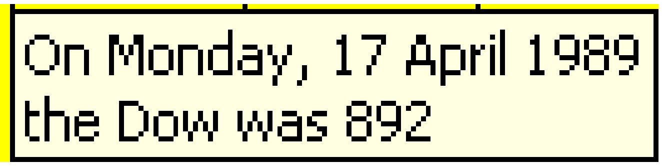

DowByDayDuringApril1989.xls - Microsoft Internet Explorer
File Edit View Insert Format Tools Data Go To Favorites Help
Back Search Favorites
Address C:\FolderForMWSUGpaper1\DowByDayDuringApril1989.xls
Go
IDX
DowByDayDuringApril1989
start
9:29 AM

Dow By Day During April 1989

Go To Graph of This Data

Obs	Date	Dow
1	Monday, 3 April 1989	877
2	Tuesday, 4 April 1989	875
3	Wednesday, 5 April 1989	879
4	Thursday, 6 April 1989	876
5	Friday, 7 April 1989	880
6	Monday, 10 April 1989	878
7	Tuesday, 11 April 1989	882
8	Wednesday, 12 April 1989	884
9	Thursday, 13 April 1989	878
10	Friday, 14 April 1989	892
11	Monday, 17 April 1989	892
12	Tuesday, 18 April 1989	908
13	Wednesday, 19 April 1989	913
14	Thursday, 20 April 1989	908
15	Friday, 21 April 1989	922
16	Monday, 24 April 1989	921
17	Tuesday, 25 April 1989	915
18	Wednesday, 26 April 1989	915
19	Thursday, 27 April 1989	926
20	Friday, 28 April 1989	926

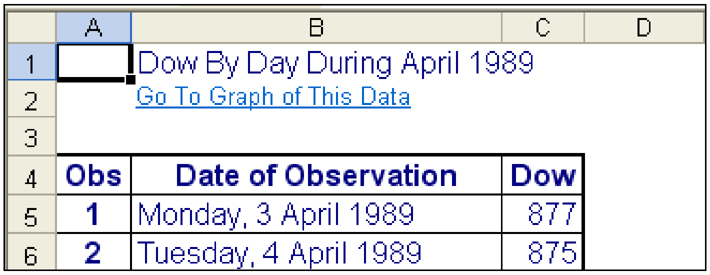
Here is an enlargement of the ALT text that popped up on the graph:



On Monday, 17 April 1989
the Dow was 892

HTML does not allow specification of font, size, or color for ALT text, only line breaks.

Here is an enlarged picture of the upper left corner of the spreadsheet:



	A	B	C	D
1		Dow By Day During April 1989		
2		Go To Graph of This Data		
3				
4	Obs	Date of Observation	Dow	
5	1	Monday, 3 April 1989	877	
6	2	Tuesday, 4 April 1989	875	

The titles must start in Column B, rather than Column A, not by my choice and not due to any explicit coding. I used column spanning for the title, and this is the result. Without use of column spanning, the default result is to insert the full title into the heading for Column A, which looks peculiar. Column A is stretched to fit the title.

```
  /* Here is preliminary code used to prepare data for the graph and spreadsheet: */

DATA work.ToReport
  (KEEP=date snydjcm Day AltTextVar);
LENGTH AltTextVar $ 200 Day $ 2;
  /* Defining AltTextVar longer than it needs to be will do no harm.
     The Alt Text pop-up box will be sized only as big as it really needs to be. */
SET sashelp.citiday (KEEP=date snydjcm
  WHERE=(YEAR(date) EQ 1989 AND MONTH(date) EQ 4));
AltTextVar='alt="On ' ||
  TRIM(LEFT(PUT(date,weekdatx.))) || '0D'X || 'the Dow was ' ||
  TRIM(LEFT(PUT(snydjcm,5.))) || '"';
  /* '0D'X forces a line break. You can do line breaks,
     but HTML does not support font, size, or color for ALT text. */
Day = PUT(DAY(date),2.);
  /* Day is numeric day of the month.
     There will be gaps since these are trading days,
     not every possible calendar in the month. */
RUN;

  /* Here is the symbolic variable set-up for web and graphs: */

%LET Path  = C:\SGF2010\RandIV\Output\FromCommentedCode\9.2;
  /* This folder will be used for all output. */

%LET Title = Dow By Day During April 1989;
  /* This text will be used for the plot title in the graph and
     for the web page title at the top of the browser window */

%LET GraphDescription = &Title;
  /* This will be the pop-up description when resting the mouse on the web graph,
     but not on a plot point.*/
```

```
%LET BodyFileName  = %SYSFUNC(COMPRESS(&Title));
  /* Remove blanks to avoid problems for links.
     This will be the filename for the web page with file extension html. */
%LET GraphFileName = %SUBSTR(&BodyFileName,1,8);
  /* Will be the filename for the graph with file extension gif
     that is stored in the Folder specified above as Path. */
  /* See the note later about filename length. */

  /* Here is more set-up for web and graph: */

ODS NORESULTS;
  /* At end of run, do NOT automatically open the web page
     in SAS Display Manager or Enterprise Guide. */

ODS LISTING CLOSE;
  /* We are not producing any text output at this phase. */

GOPTIONS RESET=ALL;
  /* Turn off all pre-existing GOPTIONS.
     Also turns off any TITLE or FOOTNOTE statements. */

  /* If using SAS Enterprise Guide, go to Tools > Options > Results > Result Formats
     and turn off HTML and all other Result Formats. */

 GOPTIONS DEVICE=GIF; /* In V9.2, ODS HTML would select PNG as default. */

* GOPTIONS TRANSPARENCY;
  /* Uncomment the above statement if using a colored web page,
     in order to let the web page background show through.
     NOTE: PNG device driver would not support transparency. */

GOPTIONS XPIXELS=980 YPIXELS=525; /* Size to fit in a 1024 X 768 screen
    with an extra tool bar using Internet Explorer 8 as the web browser.
    This assures no need to scroll. */

GOPTIONS FTEXT='Verdana/Bold' HTEXT=4PCT;
  /* Font and Size of any text in graph for which they are not explicitly assigned,
     or for which no direct, explicit controls are provided in SAS/GRAPH. */
  /* Verdana is my recommended sans serif font for small to medium text.
     Georgia is my recommended serif font for large text. These fonts were developed
     for Microsoft expressly for web and other non-print display readability. */
  /* Using PCT (percent) of graph area height with goptions htext=N.NNPCT; as above
     or height=N.NNPCT on other SAS/GRAPH statements removes dependence on the unknown
     algorithm used to translate point size to displayed size, or on knowing the
     number of rows in the graph area. */
  /* If using the GIF (or PNG) device driver, the font need not be available on the
     displaying PC: the font is used by SAS/GRAPH to draw the image. It is, in effect,
     imbedded in the raster image. */


PROC CATALOG CAT=work.gseg KILL; RUN; QUIT;
  /* This code PREVENTS other messages during reruns of your code during the session,
     BUT the FIRST run of your code will produce THESE messages in the SAS log:
     ERROR: Catalog WORK.GSEG does not exist.
     WARNING: Command CATALOG not processed because of errors noted above. */

ODS HTML PATH="&Path" (URL=NONE) BODY="&BodyFileName..html" (TITLE="&Title")
    STYLE=Styles.Minimal;
  /* The URL=NONE assures that the web page and graph are portable. They can be moved
     to a different folder and/or can be attached to an email message to be filed off
     by the recipient, and the graph will still display in the web page.
     You can omit STYLE= , or specify any style that you prefer. */
  /* If you are experienced with use of symbolic variables, ignore this comment.
     The symbolic variable BodyFileName was defined earlier during the web and graph
     set-up. You might wonder why the double dot .. in:
                                                        BODY="&BodyFileName..html"
     The first dot signals the end of the symbolic variable. The second dot is needed,
     as usual, between filename and file extension. The resolved result at execution
     time is:
              BODY="WhateverWasYourAssignedBodyFileName.html"
     Without the second dot, the resolved result would be unusable:
              BODY="WhateverWasYourAssignedBodyFileNamehtml" */

PROC GPLOT DATA=ToReport;
TITLE FONT='Verdana/Bold' HEIGHT=5PCT COLOR=CX000000 "&Title"
      JUSTIFY=CENTER COLOR=CX0000FF
      "Rest your mouse on any point for a pop-up box of precise values"
      JUSTIFY=CENTER COLOR=CX006600 UNDERLIN=2
      LINK="&BodyFileName..xls"
      'Go To Spreadsheet of This Data';
   /* JUSTIFY=CENTER forces a link break. */

SYMBOL1 INTERPOL=JOIN /* connect the plot point markers */
        LINE=1  /* type of line, 1 is default, 1 = solid line */
        WIDTH=1 /* width of line, 1 is same as default */
        VALUE=DOT  /* select the plot point marker */
        HEIGHT=1.5 /* size   the plot point marker */
        COLOR=CX006600; /* Use medium dark browser-safe green
                           for plot line & plot point marker. */

AXIS1 LABEL=NONE /*              no axis label */
      MAJOR=NONE MINOR=NONE   /* no tick marks */
      STYLE=0; /*  line style 0: no axis line  */

PLOT  snydjcm*Day /

  NAME="&GraphFileName"

  DESCRIPTION="&GraphDescription"

  HTML=AltTextVar /* The value of each point’s AltTextVar serves as the pop-up text
                     when the mouse is rested on the point. Though not the case in
                     this example, AltTextVar can also contain HTML for a hyperlink
                     to any web page that is deemed relevant. */

  NOFRAME /* The suppressed graph frame would be on the top and the right side only.
             The horizontal and vertical axes, which are also being suppressed here,
             by specifiying STYLE=0 in the AXIS1 statement above,
             would supply the bottom and left side of the frame. */

  VAXIS=AXIS1 HAXIS=AXIS1

  AUTOVREF LAUTOVREF=33 WAUTOVREF=1
  AUTOHREF LAUTOHREF=33 WAUTOHREF=1;
  /* I am adding reference lines, for demonstration purposes, but they are not really
     needed. The viewer can use pop-up text for precise detail, rather than try to
     estimate point values based on the grid. */
  /* AUTOVREF and AUTOHREF produce a reference line at every major vertical and
     horizontal axis tick mark, even if the tick mark and tick mark value are
     suppressed. Line type 33 is the finest dotted line. Line type 35 produces
     equally fine dots, but with the widest spacing for that dot size. */
  /* Version 9.1.3 does not support WAUTOVREF and WAUTOHREF.
     The reference line thickness value of 1 is actually the default.
     In Version 9.2, WAUTOVREF and WAUTOHREF are available to specify reference line
     width. Providing an option to make reference lines more conspicuous is curious in
     light of a concurrent development decision in Version 9.2 to make their color
     less conspicuous. In Version 9.2, if you use STYLE=Styles.Default (or omit to
     specify STYLE=), the default color of reference lines is very light grey (nearly
     white). You can overcome that by, in the PLOT statement, adding:
     CAUTOVREF=BLACK CAUTOHREF=BLACK */

RUN; QUIT;

ODS HTML CLOSE;

  /* If you omit the QUIT statement above, the graph PROC will not terminate, and the
     consequences might be not be what you expect. The reason for the QUIT statement
     is to allow you to create multiple graphs in one graphic procedure invocation and
     to share any graphic controls that are to be in common, using code like this:

     PROC GPLOT ...;
     < some statements in common here >
     PLOT ...;
     RUN;
     PLOT ...;
     RUN;
     QUIT; */

  /* Now create the companion spreadsheet. */

GOPTIONS RESET=ALL;
  /* Prior GOPTIONS can affect non-SAS/GRAPH PROC output. */

ODS HTML PATH="&Path"
    (URL=NONE) /* portable web page, graph, & hyperlink */
    BODY="&BodyFileName..XLS";

  /* Use COLSPAN= below so that TITLE lines do not get put only into Column A and
     expand its width to length of longest TITLE line.
     Spread them across four columns, which expand, if needed */

TITLE
  JUSTIFY=LEFT "<td COLSPAN=4>&Title</td>"
  JUSTIFY=LEFT
  "<td COLSPAN=4>
   <a href='" "&BodyFileName..html" "'>
   Go To Graph of This Data
   </a>
   </td>";

PROC PRINT DATA=ToReport LABEL;
VAR date snydjcm;
LABEL snydjcm='Dow';
FORMAT date weekdatx.;
FORMAT snydjcm 5.; /* Use format 5. to prepare for when the Dow breaks 10000. */
RUN;
ODS HTML CLOSE;
```

For your possible future reference, the generic form of the PLOT statement for a multi-line plot is:

```
PLOT yVariable*xVariable=zVariable / . . . ;
```

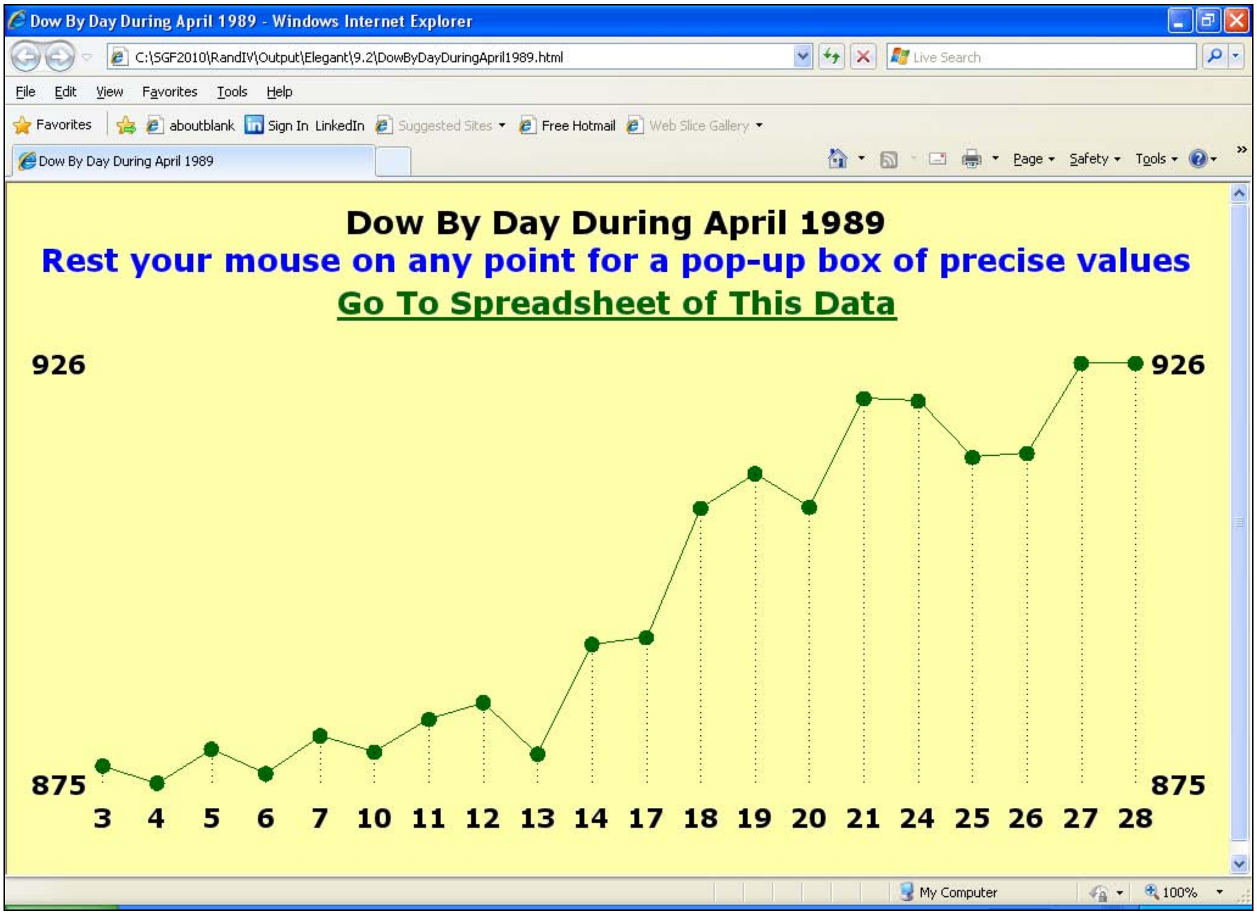
where zVariable is a categorical variable. For each distinct value of zVariable, it is necessary to specify a SYMBOLN statement (unless you want to take the defaults). You can use the N sequence suffix of the SYMBOLN statements to correlate to the sort sequence of the values of the zVariable. E.g., for a three-line plot where zVariable has values of Oranges, Apples, and Bananas, their associated SYMBOL statements would be SYMBOL3, SYMBOL1, and SYMBOL2.

Code Only (No Comments) for the Basic Solution

```
DATA work.ToReport(KEEP=date snydjcm Day AltTextVar);
LENGTH AltTextVar $ 200 Day $ 2;
SET sashelp.citiday
  (KEEP=date snydjcm WHERE=(YEAR(date) EQ 1989 AND MONTH(date) EQ 4));
AltTextVar='alt="On ' || TRIM(LEFT(PUT(date,weekdatx.))) || '0D'X ||
           'the Dow was ' || TRIM(LEFT(PUT(snydjcm,5.))) || '"';
Day = PUT(DAY(date),2.);
RUN;
%LET Path  = C:\SGF2010\RandIV\Output\FromUNcommentedCode\9.2;
%LET Title = Dow By Day During April 1989;
%LET GraphDescription = &Title;
%LET BodyFileName  = %SYSFUNC(COMPRESS(&Title));
%LET GraphFileName = %SUBSTR(&BodyFileName,1,8);
ODS NORESULTS;
ODS LISTING CLOSE;
GOPTIONS RESET=ALL;
GOPTIONS DEVICE=GIF;
GOPTIONS XPIXELS=980 YPIXELS=525;
GOPTIONS FTEXT='Verdana/Bold' HTEXT=4PCT;
PROC CATALOG CAT=work.gseg KILL; RUN; QUIT;
ODS HTML PATH="&Path" (URL=NONE) BODY="&BodyFileName..html" (TITLE="&Title")
    STYLE=Styles.Minimal;
PROC GPLOT DATA=ToReport;
TITLE FONT='Verdana/Bold' HEIGHT=5PCT "&Title"
  JUSTIFY=CENTER COLOR=CX0000FF
  "Rest your mouse on any point for a pop-up box of precise values"
  JUSTIFY=CENTER COLOR=CX006600 UNDERLIN=2
  LINK="&BodyFileName..xls" 'Go To Spreadsheet of This Data';
SYMBOL1 INTERPOL=JOIN LINE=1 WIDTH=1 VALUE=DOT HEIGHT=1.5 COLOR=CX006600;
AXIS1 LABEL=NONE MAJOR=NONE MINOR=NONE STYLE=0;
PLOT  snydjcm*Day / NAME="&GraphFileName" DESCRIPTION="&GraphDescription"
  HTML=AltTextVar NOFRAME VAXIS=AXIS1 HAXIS=AXIS1
  AUTOVREF LAUTOVREF=33 WAUTOVREF=1 AUTOHREF LAUTOHREF=33 WAUTOHREF=1;
RUN; QUIT;
ODS HTML CLOSE;
GOPTIONS RESET=ALL;
ODS HTML PATH="&PATH" (URL=NONE) BODY="&BodyFileName..xls";
TITLE JUSTIFY=LEFT "<td COLSPAN=4>&Title</td>"
  JUSTIFY=LEFT "<td COLSPAN=4><a href='"
  "&BodyFileName..html"
  "'>Go To Graph of This Data</a></td>";
PROC PRINT DATA=ToReport LABEL;
VAR date snydjcm;
LABEL snydjcm='Dow';
FORMAT date weekdatx. snydjcm 5.;
RUN;
ODS HTML CLOSE;
```

An Elegant Alternative Version of the Trend Chart

The best solution is one where the trend chart only includes what is really needed. Most of the vertical axis tick marks and reference lines have no real practical value. The precise y values can only be obtained from the pop-up ALT text or from the companion spreadsheet. It is sufficient to show the precise minimum and maximum y values. There is no reason for the horizontal axis reference lines to stretch all the way to the top of the plot area frame. They can be replaced with needle lines that extend only to the points. As the number of plot points increases, the value of the needle lines increases. Of course, you eventually need to reduce the size of the horizontal tick mark values. In principle, one can assert that the pop-up text makes the needle lines superfluous, but it is useful to be able to inspect a trend chart with the eye before resorting to the mouse to display the precise y-value for the plot point at a particular x-value of interest.



Dow By Day During April 1989

Rest your mouse on any point for a pop-up box of precise values

Go To Spreadsheet of This Data

The PROC GPLOT code must be changed as shown below. A PLOT2 statement and its SYMBOL2 statement are added. An AXIS2 statement is added so that vertical tick marks can be handled differently from the horizontal tick marks. The OFFSET= is added to the AXIS1 statement for the horizontal axis. The code for the reference lines is removed.

```
PROC GPLOT DATA=ToReport;
TITLE . . .; /* Use the same TITLE statement as in other code. */
SYMBOL1 INTERPOL=JOIN   LINE=1  WIDTH=1 VALUE=DOT HEIGHT=1.5 COLOR=CX006600;
SYMBOL2 INTERPOL=NEEDLE LINE=33 WIDTH=1 VALUE=NONE           COLOR=CX000000;
AXIS1 LABEL=NONE MAJOR=NONE MINOR=NONE STYLE=0
  OFFSET=(1 PCT,1 PCT); /* With the vertical axis reference lines removed,
  the vertical axis tick mark values are displayed too close to the plot area.
  Use OFFSET to insert some space. */
AXIS2 LABEL=NONE MAJOR=NONE MINOR=NONE STYLE=0
      ORDER=(&Ymin &Ymax) VALUE=("&YminDisplay" "&YmaxDisplay");
PLOT  snydjcm*Day / NAME="Elegant" DESCRIPTION="&GraphDescription"
  HTML=AltTextVar NOFRAME VAXIS=AXIS2 HAXIS=AXIS1;
PLOT2 snydjcm*Day /
  HTML=AltTextVar VAXIS=AXIS2;
RUN; QUIT;
```

The following preparatory code must be run after the work.ToReport SAS data set has been created:

```
PROC MEANS DATA=work.ToReport NOPRINT MIN MAX;
VAR snydjcm;
OUTPUT OUT=work.MinMax MIN=MIN MAX=MAX;
RUN;
```

```
%GLOBAL Ymin Ymax;
DATA _NULL_;
SET work.MinMax;
CALL SYMPUT('Ymin',MIN); CALL SYMPUT('Ymax',MAX);
CALL SYMPUT('YminDisplay',TRIM(LEFT(PUT(MIN,5.)))); /* display as integer */
CALL SYMPUT('YmaxDisplay',TRIM(LEFT(PUT(MAX,5.)))); /* display as integer */
RUN;
```

To get the light yellow background, uncomment the GOPTIONS TRANSPARENCY statement, run the code below, and on the ODS HTML statement specify STYLE=Styles.MinimalLightYellow. For more on effective use of color, please see Reference 1.

```
PROC TEMPLATE;
EDIT styles.Minimal AS styles.MinimalLightYellow;
  STYLE body        / background = CXFFFFAA;
  STYLE systemtitle / background = CXFFFFAA;
  /* color CXFFFFAA is browser-safe light yellow */
END; RUN;
```

Communication-Effective Graphic Design – A Short Introduction

- Show Them What Is Important.
- Rely on The Power of Simplicity.
- Deliver Elegance: Everything that is needed, Only what is needed.
- Let your data talk.

Omit graph frame and axis lines.

Use only tick mark values and reference/grid lines that add communication or interpretation value.

Avoid background images, textured backgrounds, and color gradient backgrounds. Such backgrounds can impair the readability of text and numbers.

Make all text and numbers readable: sufficient size with high contrast on a plain background.

Use big fonts wherever practical. Your graph should not be a vision test.

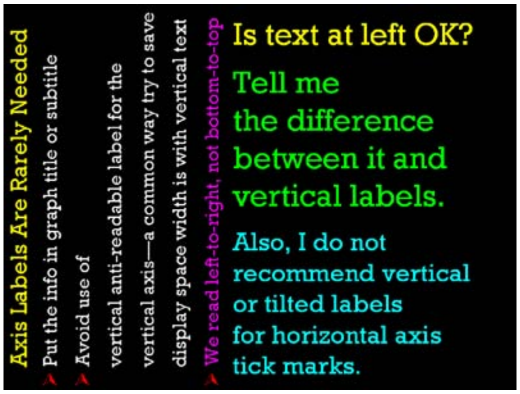
If possible, make every graph title a headline—TELL viewers the implication or revelation, rather than only what variables are being graphed. This principle was not exemplified in this paper—the message in the data was self-evident.

Do not use area fill under a plot line. It would have no communication value. If the graph is a multi-line plot, one plot line's area fill can overlay part of one or more other plot lines.

Unless it includes negative numbers, normally start your vertical axis at zero. (For bar charts, this avoids distortion of the true relative magnitude of the y-values.) For trend lines, devoting the entire vertical space to fluctuations in the y-value can create needless anxiety or unjustified elation about what are really small changes of no great consequence. (This principle was not exemplified in this paper. The plot line would have been nearly flat for a very boring illustration. However, for a real application, boring is OK. The data and image need to inform, not excite.)

Anything that might be in an axis label can instead be put into the graph's subtitle or title. This happens to eliminate a common practice which I find sub-optimal, even if popular—namely, the often needless use of vertical text.

Make a reasonable best effort to keep all of your text horizontal. English text is read left-to-right, not bottom-to-top. Here is a slide image that makes the point well:



Axis Labels Are Rarely Needed

- Put the info in graph title or subtitle
- Avoid use of vertical anti-readable label for the vertical axis—a common way try to save display space width is with vertical text
- We read left-to-right, not bottom-to-top

Is text at left OK?

Tell me the difference between it and vertical labels.

Also, I do not recommend vertical or tilted labels for horizontal axis tick marks.

If you are using color, realize that if lines or characters are too thinly drawn, their color is difficult or impossible to distinguish. Similarly, plot point markers and legend color samples must be sufficiently big.

If your graph does not <u>require</u> color (e.g., for a legend to be able to distinguish different plot lines, pie slices, or bars), the best color combination is black and white. For more about communication-effective color combinations or communication with color in other respects, please see Reference 1, my most comprehensive paper on this topic.

Bibliography

1. "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*, SAS Institute Inc. (Cary, NC), 2004. Find it on the web at http://www2.sas.com/proceedings/sugi29/176-29.pdf.

To find more of my publications on related topics, please go to www.lexjansen.com. You can search by author name, etc. through Lex Jansen's vast treasure trove of links to over 10,000 SAS papers (back to some as early as 1996) to find any author's contributions to the SAS Global Forum, SUGI, MWSUG, NESUG, PharmaSUG, PhUSE, PNWSUG, SCSUG, SESUG, and WUSS conferences.

Contact Information, Etc.

I am always interested in comments and questions about visual communication and/or SAS software.

LeRoy Bessler PhD
Le_Roy_Bessler@wi.rr.com

Zum sehen geboren, zum schauen bestellt.
(Born to see, meant to look.)
—Goethe

The Power to Show
The Power of Simplicity

A SAS user since 1978, Dr. LeRoy Bessler has shared his knowledge and experience with other users at conferences throughout the USA and in Montreal, London, Heidelberg, and Dublin. Though a SAS generalist with long experience in Base SAS, SAS macro language, and SAS tools for access to non-SAS data, his special strengths include communication-effective visual communication and reporting, web information delivery, highly formatted Excel reporting, SAS/GRAPH, ODS, Software-Intelligent Application Development for Reliability, Reusability, Extendibility, and Maintainability, and creation of unique tools to support the SAS BI server and its users. He is a regular contributor to *VIEWS News*, the web newsletter of the VIEWS International SAS Programmer Community.