

Communication-Effective and -Efficient Enterprise Performance Reporting

LeRoy Bessler, Bessler Consulting & Research, bessler@execpc.com

Introduction

The need for, and existence of, Enterprise Performance Reporting (EPR) long antedates the Balanced Scorecard concept and SAS Strategic Vision. I created my first EPR system about fifteen years ago, with Base SAS®, SAS/GRAPH®, and SAS macro language (the most useful second language for any SAS programmer), and have been fine-tuning my ideas and tools for SAS- and SAS/GRAPH-enabled reporting ever since.

This paper first discusses some simple design concepts that can be applied in any EPR application, and probably in any reporting applications with a similar function, but less grand functional description. Today, the concepts certainly seem trivial and obvious, now that we have the convenient and quick hyperlinked drilldown paradigm of the web. But they weren't apparent to the builders of the system that I replaced.

(To see one perspective on how to do web-deployment of information, please refer to the VIEWS 2001 paper "Your Graphs and Tables at their Best on the Web with ODS".)

Rather than attempt exhaustive analysis of structures for EPR, this paper will focus on some design principles and coding solutions that are applicable to the elements of any EPR. For expediency in publication, I have simply concatenated some prior works, in whole or in part. I hope you can forgive the seams (and variation in format and fonts—in one I gratify my preference for Rockwell over Times New Roman), and find the tools, the techniques, and the tips that work for you.

Chapter 1. Key Ideas and Two Simple Tools for Enterprise Performance Reporting

A SAS-implemented EIS or a Balanced Scorecard is often designed, in its quest for communication effectiveness, with "Traffic Lighting" and/or "Radio Dials", using red, yellow, and green for Danger/Disaster/Bad, Caution, and OK/Safe/Good. These widely used features won't go away.

However, color blindness occurs in 8-10% of American males, and the commonest type of color blindness cannot distinguish red from green. For the visually superior American females, incidence is very small by comparison. I regret that I cannot report the numbers for the UK. If you wish to prevent this problem, you can substitute, say, blue for green. I am fond of red, white (or light grey—if you dislike an absence of color), and blue. Then, if you decide to expand your range with shades of bad and good, the natural solution is to use red, pink (light red), white (light grey), light blue, and blue.

In any case, I have a simpler design for communication effectiveness and efficiency, which does not require traffic lighting, radio dials, or any colors whatsoever.

Show them only what they need to know. Make it easy for them to stop looking when they have had enough. And remember that old (American?) adage, "No news is good news." Here's how.

And, coincidentally, I will first talk about news reporting. Here is how a newspaper article may be structured. There is a headline. If you read nothing else, this should telegraph to you the essence of the news report. There is often a subhead. This will amplify the headline a bit. There is an opening paragraph. This will summarize the whole item, or at least highlight the most important aspects of what is being reported. The remainder of the article is laid out so that you can stop reading at any point, and always come out with the most important of what was available altogether. But what does this have to do with Enterprise Performance Reporting?

Enterprise Performance Reporting

My first adventure in the area of EPR was to remake a complicated report that was going to my company's Chief Information Officer. It covered performance, capacity, and usage of computing and network resources. Someone's original design of this report was a concatenation of detail-oriented information from each of the various reporting/operating departments. It did not invite reading. I replaced it with a three-tier report.

Page 1. Exception Report

If no goals were missed and no dangerous thresholds were reached, it simply said "No Exceptions to Report." Unless an executive or manager were unusually curious, or had time to use up, there was no need to go on.

OK, but some people like to "take the pulse of the enterprise". They turn, or now they would click, to Page 2.

Page 2. Summary Report

Every measurement criterion was reported with: Description; Value During Latest Reporting Period; and Goal or Threshold. In hindsight, a simple-to-digest fourth column to add would have been "Increase/Decrease", either simply stating the direction, or—better—listing the signed amount. That measure of change could be the absolute amount, or could be a percentage if that were the better measure of significance. E.g., when a stock market performance indicator drops 100 points on a base of what might be 10000, that is not as significant a concern as a drop of 100 points when the indicator may have been in the low 1000s.

And there are people for whom even this is not enough. They want to study the enterprise's complete medical history, and to look at the results from Xray, CAT Scan, MRI, PET Scan, UltraSound, or, worse, the endoscope. Their next step is to hunt through supporting detail pages, or to click on individual Summary Table entries of interest to them.

Section 3. Trend Charts and Subsetted Ranking Reports

The value of, and appropriate application for, trend charts need no explanation. In a related paper, I will show you the %TRNDANNO macro, to reliably produce a sparsely, selectively, or fully annotated trend line, with collision-free annotation and in the most communication-effective manner. In the VIEWS web paper cite above, you can find how to provide soft annotation with flyover text (which you cannot print).

If a monitored performance concern requires a list of information, rather than a trend chart, it is almost always the case that the bulk of the information of significance can fit on one page.

When I was a young, freshly appointed information technology manager, the senior member of my staff, who was more worldly wise than I, counseled me that the reports we prepared for submission to higher management should be designed to fit on one page. He said, "If a report is longer than one page, they just won't read it."

It wasn't until some years later, however, that I perfected a technique to put that concept into practice for measurement-oriented reports.

The idea of a ranking report is certainly not unique. But the ranking report, dedicated to my principle "Show Them What's Important", is most successfully augmented with the embodiment of my other principle "Let Part Stand for the Whole" (an aphorism actually enunciated earlier by someone else, whose name I cannot remember), in the subsetted ranking report.

And even that idea is not unique. The American popular music sales industry is still using the Top 40 List (at least I think it is), and I recall a television programme called “Top Ten”, from the time when my family and I lived in London in the early 1970’s.

But my subsetting ranking report is distinguished by its maximally effective and simple titles. See the sample output in The Maximally Informative Subsetting Ranking Report section. You can do this the hard way, with a preliminary calculation and subsequent hard-coding in the report-writing program. Or you can do this the easy way. With a few preliminary steps you can sum the measurement of interest across all possible reportable entities, and across only those being reported, and create macro variables to be picked up in the dynamically generated titles.

A simple, and simply supplied, subtitle reassures the report viewer that *what is left out just doesn’t matter very much*. It has been my experience that what fits on one page, or in a small table, typically covers about 70% to 99% of the concern. For the real-data example below of The Maximally Informative Subsetting Ranking Report, the Top 10 account for only 66.4% of the total. If that’s not enough for comfort, just turn up the Top N selection value until you are satisfied that you can “Show Them What’s Important”.

Now, rather than doing report iterations until enough observations are picked up to calm the nerves of the report user, another variation on the macro shown (but left out for brevity), is to make the selection criterion parameter be the specific comfort-level threshold that is desired. I.e., supply a parameter to specify that as many observations will be selected—in ranked sequence—for the list until they account for “&top_pct” of the total. Even if you choose to alter the generic part of the text of the titles, the same four variables are dynamically loaded.

But, before we actually make a foray into coding the ranking report, let’s make a quick, easy digression.

The Best of All Possible Reports

In Enterprise Performance Reporting, the happiest result is that there are no exceptions to report. The enterprise is safely on course.

There are undoubtedly multiple ways to do this. Here’s something like what I have often used.

You first must do the processing to prepare the items for the exception report. They are in a data set called WORK.EXCEPTS. Or you may have filed them in a permanent, report-period-keyed data set, so that you maintain an easily auditable exception history.

```
DATA _NULL_;
IF HOWMANY EQ 0
THEN CALL SYMPUT('excepts', 'NONE');
ELSE CALL SYMPUT('excepts', 'YES');
STOP;
SET EXCEPTS NOBS = HOWMANY;
RUN;

%macro maybe rpt;
%if &excepts eq NONE %then %do;
DATA NOREPORT;
LINE = 'None To Report';
LABEL LINE = 'Exceptions';
RUN;
OPTIONS NOCENTER;
PROC PRINT DATA=NOREPORT LABEL
NOOBS; /* suppress observation numbers */
VAR LINE;
RUN;
```

```

%end;
%else %do;
PROC PRINT DATA=EXCEPTS LABEL NOOBS;
VAR EXCPDESC EXCPACTL EXCPSTD;
RUN;
%end;
%mend maybe rpt;

```

```

%maybe rpt
RUN;

```

I don't think that I need to show you sample outputs from this report writer.

The Maximally Informative Subsetted Ranking Report (basic implementation)

I call this the *maximally informative* subsetted ranking report because of the helps provided in the title and subtitle.

Here are the data and the program:

```

DATA SUGI14;
LENGTH STATE $ 14;
INFILE CARDS;
INPUT ST $ ATTENDEES;
IF ST NE 'DC'
THEN STATE = STNAMEL(ST);
ELSE STATE = 'Washington DC';
CARDS;
AK    4
AL   12
AR   16
.
.
.
WI   50
WV    9
WY    1
;
run;

%let n = 10;

PROC SORT DATA=SUGI14 OUT=TOSUBSET;
BY DESCENDING ATTENDEES;
RUN;
DATA SUBSET;
RETAIN TOT4TOP TOT4ALL COUNTALL 0;
SET TOSUBSET END=THATSALL;
TOT4ALL  = TOT4ALL + ATTENDEES;
COUNTALL = COUNTALL + 1;
IF _N_ LE &n THEN DO;
    TOT4TOP = TOT4TOP + ATTENDEES;
    OUTPUT;
END;
IF THATSALL;
CALL SYMPUT('top_pct',
TRIM(LEFT(PUT((TOT4TOP / TOT4ALL),PERCENT7.1))));
CALL SYMPUT('all',TRIM(LEFT(COUNTALL)));
CALL SYMPUT('tot_meas',TRIM(LEFT(TOT4ALL)));

```

```

RUN;
OPTIONS NOCENTER;
TITLE1 "Top &n of &all States With Attendees at SUGI 14";
TITLE3 "Account for &top_pct of the Total (Total = &tot_meas)";
PROC PRINT DATA=SUBSET LABEL;
ID STATE;
VAR ATTENDEES;
LABEL STATE      = 'State';
LABEL ATTENDEES = 'Attendees';
RUN;

```

Here's the output (the values 10, 51, 66.4, and 3350 are dynamically program-supplied):

Top 10 of 51 States With Attendees at SUGI 14

Account for 66.4% of the Total (Total = 3350)

State	Attendees
California	736
North Carolina	447
Texas	178
New York	173
New Jersey	141
Pennsylvania	127
Ohio	110
Illinois	109
Virginia	103
Maryland	99

Before leaving this topic, I want to mention that, if you were to do a, presumably multi-column, report with PROC TABULATE in which you wish to list the rows based on the rank of observations in one of the response columns, that PROC, despite its (for me, frustratingly named) ORDER= parameter, will not let you do that ranking. I developed a simple solution. If you ever need to use it, please send me an email.

Author Information

LeRoy Bessler, Ph.D.
 Bessler Consulting & Research
 PO Box 96
 Milwaukee, WI 53201-0096, USA
 bessler@execpc.com
 1-414-351-674

Chapter 2. Design and Construction for ANY Report

The Power to Show: Using Information and Image to Reveal and Persuade

LeRoy Bessler, Bessler Consulting & Research, bessler@execpc.com

Introduction

Some TV ads make you remember action and images, but not what was being promoted. The time and attention of the “audience” for your charts are precious, as are your time and effort to create those charts. Software and hardware are power tools, but defaults produce lots of sub-optimal results quickly.

This paper and presentation show: (a) how to design charts to inform and to influence, not to impress, so you can deliver the messages and the meaning in your data with images easily and quickly interpreted, and having memorable *significance*; and (b) how to design charts with overview impact, to accelerate decision-making, and—for situations where it is needed—precise detail, to facilitate reliable decision-making.

My focus is on some guidelines and examples, not code. Take-away knowledge is SAS/GRAPH® applicable, but is also software- and hardware-independent. Learn how to make and to let the information and the image reveal and persuade: customizing for communication, not for decoration.

B. Basic Graphic Guidelines

There is no Section A, only B, W, C, and T.

1. 3D is for Three Variables

Good design and interesting data can stand on their own. Omit drop-shadows, clip art, etc.

3D pie charts *always distort* relative size of shares of the whole, defeating the visual communication purpose of pie charts (see Figure 2). 3D bar charts needlessly complicate a simple image, usually making it more difficult to interpret (see Figure 3).

2. For line charts or bar charts, usually start the response axis at zero. Minimize meaningless fluctuations/differences point-to-point, and avoid magnified “growth” or “decline”. Prevent needless questions, and undeserved elation or alarm. See Figures 4 & 5. Note the desirable near-flatness of the upper chart in Figure 4—the variation over twenty-two years was less than ten percent. (But, of course,

when the fine structure of a chart is really important, devote all the available space to the actual data range.)

3. Remove axis clutter. (See all custom charts.)

Turn off axis lines—they tell nothing. Turn off tick marks. Label (invisible) tick marks sparingly. If not turning off axis labels, supply your own, but not to state the self-evident (e.g., that dates are dates). Annotate plot points and ends of bars.

3. Use Sparse Annotation. (See Figure 4.)

Annotate y-values for the critical points of a plot line, and label only corresponding (invisible) x-axis tick marks. For a crowded or crossing multi-line plot, put critical point values in the legend. The *critical points* are start, end, maximum, minimum, and points at which the rate of growth or decline persistently changes.

4. Use area fill wisely, if at all.

Omit area fill on line charts and on radar charts. On simple bar charts, omit area fill; but use light gray if bars are close together. When fill is really needed, use solid colors or grays; use parallel lines or cross-hatching only in desperation.

5. Rank data with descending/ascending bar or pie charts. (See Figures 2, 3, & 6.)

6. To rank shares of the whole when numerous, use an Annotated Ranked Horizontal Bar Chart. (See Figure 6.)

When the shares of the whole are too numerous or too small, there is not room to display pie slice name, value, and percent of whole.

7. Use the powerful Pac-Man Pie Chart to summarize shares of the whole into two. (See Figure 1.) The idea of a two-part pie chart may seem trivial, if not silly. However, when the share of interest to your message is either tiny or huge, the image is very “impactful” and, therefore, memorable. If needed, supply details for “Other” with a table below the pie chart. Do not blunt the message by splitting the big wedge into little ones that may be as small as or

smaller than the wedge whose smallness you are emphasizing.

W. Web Design

For further discussion of this topic, please see the related VIEWS paper “Your Graphs and Tables at the Best on the Web with ODS”.

Dilbert (Scott Adams’ cartoon character) said, “I made a study of Internet use in the workplace. I found that the time spent waiting for web pages to load has wiped out all the productivity gains of The Information Age.”

1. Keep your image simple. Simple may load faster. And never underestimate power of simplicity.

2. Use browser-safe color. See Section C.

3. Can they read it? Verify that the text parts of your image will, in fact, be readable by other people who web-browse your image. A graphic image accelerates decision-making; text-borne details may be essential to decision reliability. For more about text, see Section T (on Text).

C. Communicate with Color

1. Color does not improve bad design.

Use color to communicate, not to decorate. So,

- No response levels/categories, black & white
- Few levels or categories, grey shades maybe
- Many levels or categories, color necessary

2. Beware of SAS® Predefined Color Names.

There are almost 300 of these, listed in Table 7.2 in both the Version 6 and the Version 8 edition of the SAS/GRAPH manual. They have English language names or abbreviations, such as “VIOLET”, or “LIV” for “Light Violet” (where “Light Violet” itself is not an assignable value). There’s a problem. Many names are misleading. If you display or print VIOLET and LIV, you will see that Light Violet is darker than Violet. There are other contradictions like this, such as PINK and LIPK (Light Pink). Furthermore, many of the predefined colors are too dark to be useful. Even in Version 8 these problems are not fixed. In fact, in Release 6.12 some colors were changed (e.g., GREEN), but not always for better (more about Green later). The new VIOLET, like the old, is lighter than LIV, but the new VIOLET is not Violet, for which any

reasonable standard is the color of the flower. The new PINK is a better Pink, but, unlike the old (which was actually a non-Pink), is lighter than Light Pink. To make wise color choices, create samples and pick yourself a palette. Here’s an easy way to make color samples:

```
proc gslide;
note f='Georgia' c=black ' C=PINK'
f=SWISSB c=PINK h=7 move=(+0,-1) '03'X;
note f='Georgia' c=black ' C=LIPK'
f=SWISSB c=LIPK h=7 move=(+0,-1) '03'X;
run; quit;
```

SAS Color Names: Pink & Light Pink

C=PINK



C=LIPK



3. What are “browser-safe” colors?

Well, unlikely as we may think it, there are people out there with monitors or video cards that are limited to 256 colors. How many I don’t know. Unless sure of your entire audience, you should design for the lowest common denominator. To deal with this problem, web browsers determine the limits of the video display, and, if needed, remap colors. All web browsers agree on a universal common subset, which all support.

There are 216 “browser-safe” colors. They are RGB colors, with odd names, in the SAS/GRAPH language, of the form CXrrggbb. The browser-safe RGB colors restrict rr, gg, and bb to six values 00, 33, 66, 99, CC, and FF. (6 X 6 X 6 = 216.) When a browser detects a color outside this set on a web page to be displayed on a 256-color monitor, it remaps the color to a browser-safe one. Any color that is already browser-safe displays as expected.

All SAS predefined colors have RGB equivalents, but only seven are among the safe 216 values. (GREEN used to be, but despite the listed RGB value in the manual, it was changed in Version 6.12, and is now unsafe—ironic, since true green is one of the RGB primaries, Red, Green, and Blue.)

See Figure 7 for examples of browser-safe colors. White samples are due to a coding convenience in the macro used to make the charts, not an attempt to be weird. The basic browser-safe colors are Red (CXFF0000), Yellow (CXFFFF00), Green (CX00FF00), Cyan (CX00FFFF), Blue (CX0000FF), Magenta (CXFF00FF), Black (CX000000), and White (CXFFFFFF). The upper chart in the figure is the only easy way for RGB colors to vary lightness for constant hue. The lower chart shows small sets of varied lightness, plus Purple. It is much easier to vary lightness using the HLS color system, but which HLS colors are equivalent to browser-safe RGB colors is unknown to me at this time.

T. Text Is For Reading

0. Pulitzer's First Rule for Journalists:

"Make it brief so that they will read it."

1. Make your graph title your headline. You are using the graphic image to persuade and/or reveal. Don't be reluctant to tell the viewer what you know or think it means.

2. Sparse Text makes the graph talk. Be sure every letter or number in the graph *must* be there. Superfluity detracts from emphasis.

3. Use high contrast (e.g., Black on White). On light backgrounds, colored text must be sufficiently thick. Never use Yellow on White.

4. For emphasis, usually use **Bold**, *Italic*, Underline, or ALL CAPS, not color. If using ALL CAPS for emphasis, use SPARINGLY. ALL CAPS is hard and slow to read. To verify this, prepare, and read, a long paragraph in ALL CAPS.

5. Check your SAS log for messages that the ugly SIMULATE font has been substituted for your requested font. (Unfortunately, you are not always notified. Why not?) Fix the problem, if you can. The commonest error is a misspelled font name, in which case the SAS log kindly notifies you of substitution. For a sample of this font (or of any other SAS/GRAPH font specified with **name=**), use:

```
proc gfont name=SIMULATE nobuild
SHOWROMAN HEX H=2; run; quit;
```

Wherever you use **f=NONE** in your SAS/GRAPH program, the device driver uses its default font. If you also specify a height other than one cell (i.e., not **h=1**), then SIMULATE will be used, but you will not be notified. Be aware that some drivers (e.g., the EMF driver I use to create graphs to be INSERTed into my word processing documents) do not have a default font—SIMULATE is used without notification. See *default graphs in Figures 3 & 5*. GOPTIONS SIMFONT= assigns the default font. The software as shipped has this as SIMULATE.

6. Set defaults with FTEXT, HTEXT, CTEXT.

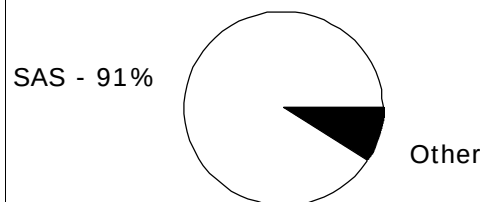
7. We are fortunate to have now access to Windows TrueType fonts in SAS. Best fonts include: Rockwell (serif—used for this paper, my readability favorite because it's the only elegant serif font, since the loss of Clarendon, that is thickly drawn along the entire contour of every character); Courier New (fixed width—essential when you need columns of numbers to line up); and Matthew Carter's designs for readability on the screen and the web, Verdana (sans serif—useful for small letters and numbers on your graph) and Georgia (serif—useful for titles, prominent footnotes, etc.). The prior sentence uses each font. Can you determine how each font has been used in the Figures.

Author Information

LeRoy Bessler, Ph.D.
Bessler Consulting & Research
Milwaukee, WI 53201-0096, USA
bessler@execpc.com
1-414-351-6748

Figure 1. Pac-Man Pie Chart

Mainframe Data Analysis Software
Market Shares



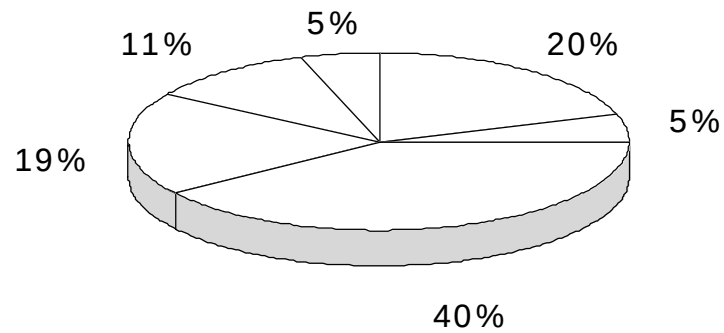
Source: Computer Intelligence, 1993

bessler@execpc.com - 22Jan2001

The Power to Show

Figure 2. Comparison of 3D and 2D Pie Charts

3D Pie Chart Always Distorts Apparent Relative Size



2D Pie Chart Always Presents Accurate Relative Size
Descending Pie Chart Shows Them What's Important

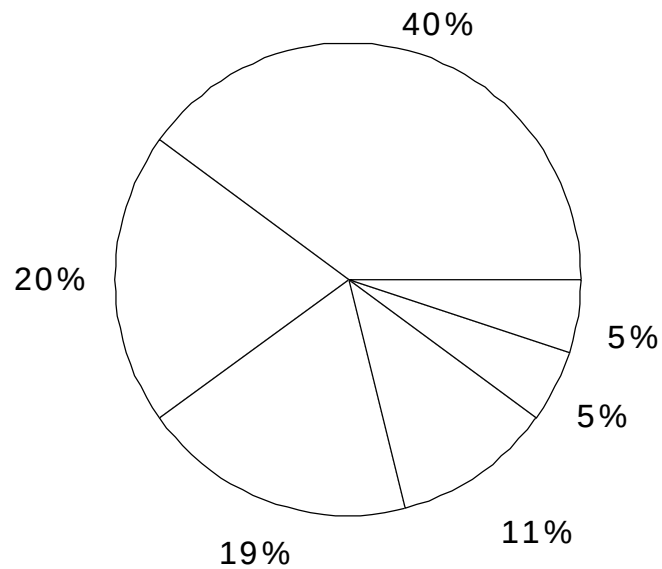
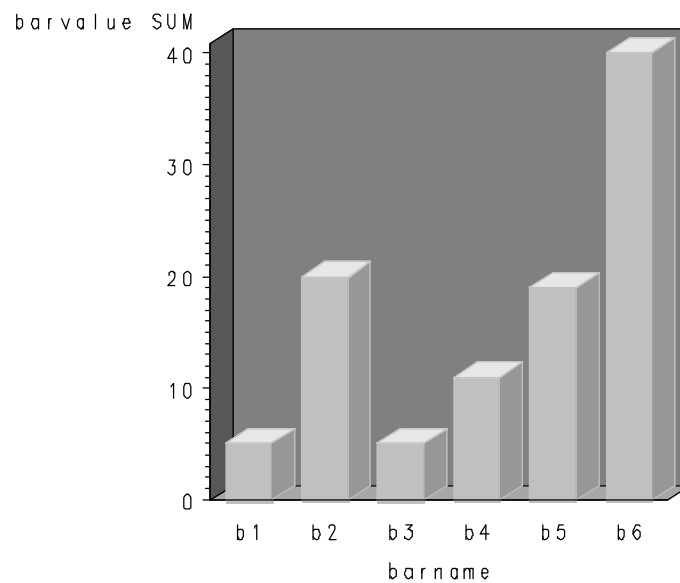


Figure 3. Comparison of 3D and 2D Bar Charts

Default 3D Bar Chart Unnecessarily Complex and Harder To Interpret



Optimized 2D Bar Chart Shows Them What's Important

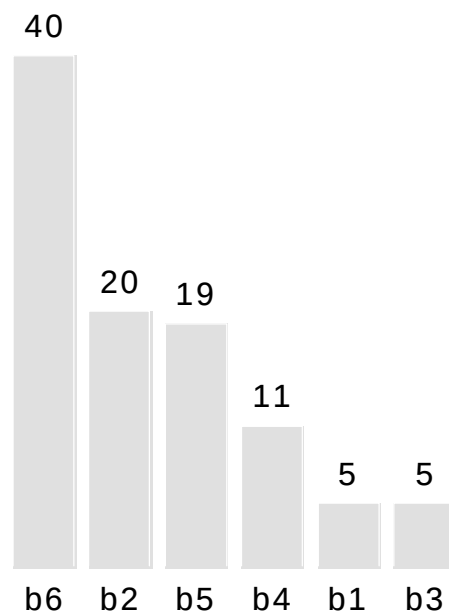
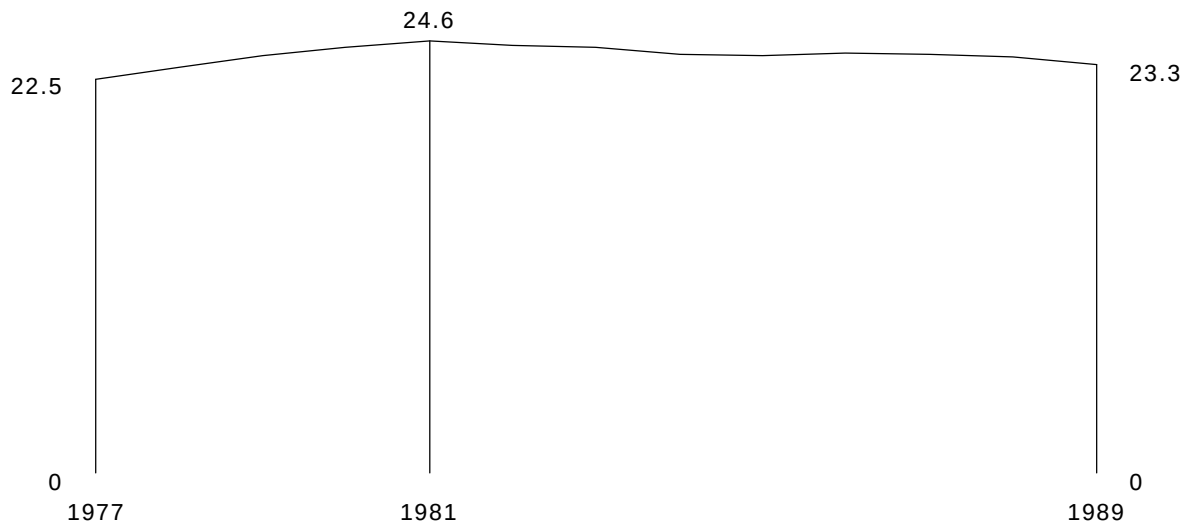


Figure 4. Sparse Annotation: Focus on Critical Points

Annual U.S. Beer Consumption Peaked in 1981
Gallons per Capita



Annual Miller Lite Production Growth Slowed in 1981
Millions of Barrels

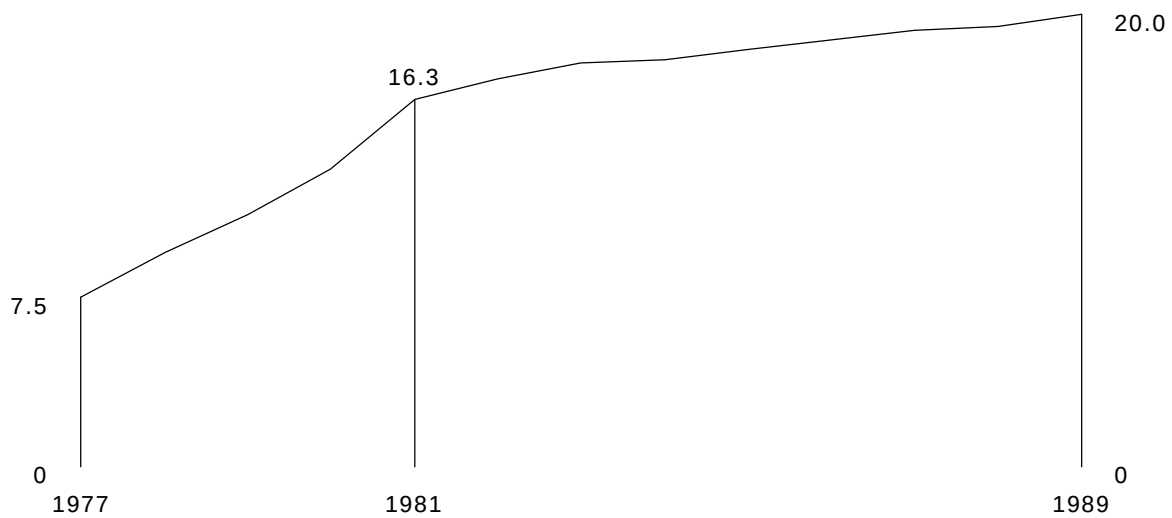
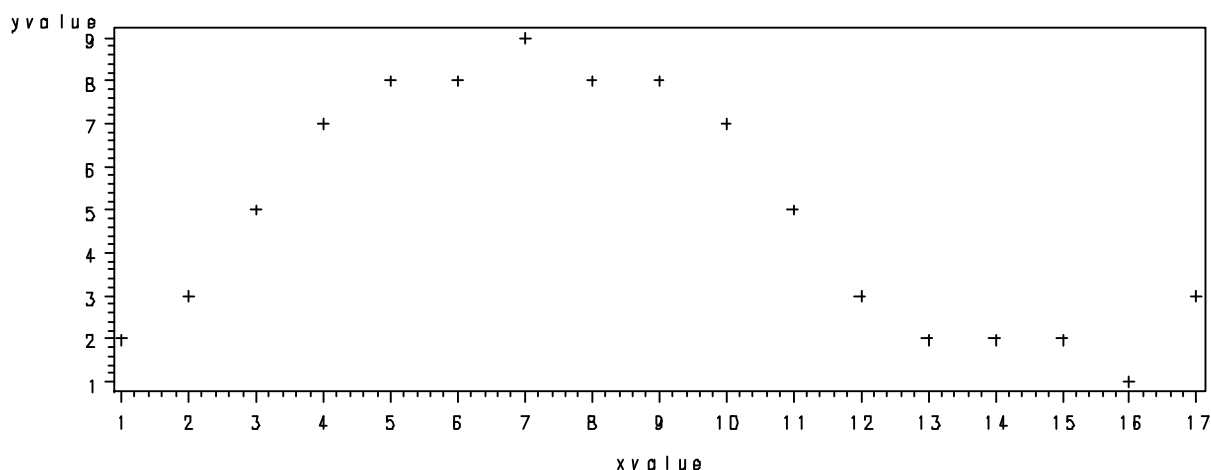


Figure 5. Trend Chart - Default vs. Custom

Default Plot of all fourteen possible 3-point / 2-segment trend changes:

up-steeper, up-same-slope, up-less-steep, up-flat, flat-up, up-down, down-flat, flat-down, down-steeper, down-same-slope, down-less-steep, down-flat, flat-flat, and down-up



Reliably Collision-Free Annotated Plot Using My Custom TRNDANNO Macro

Version 8 SAS/GRAPH has a POINTLABEL option for automatic annotation, but can suffer line-value collisions in some cases (e.g., for displays in small areas such as a quarter-page). %TRNDANNO does Full or Sparse Annotation, or Selective Annotation for a list of x-values.

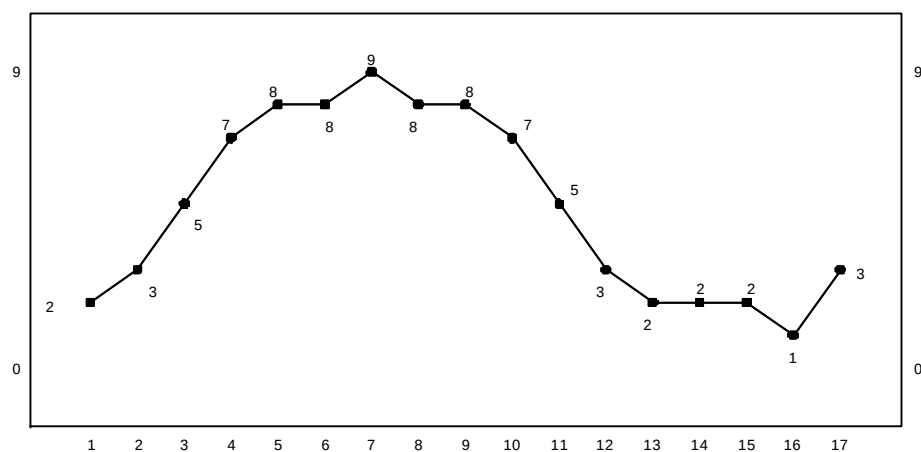
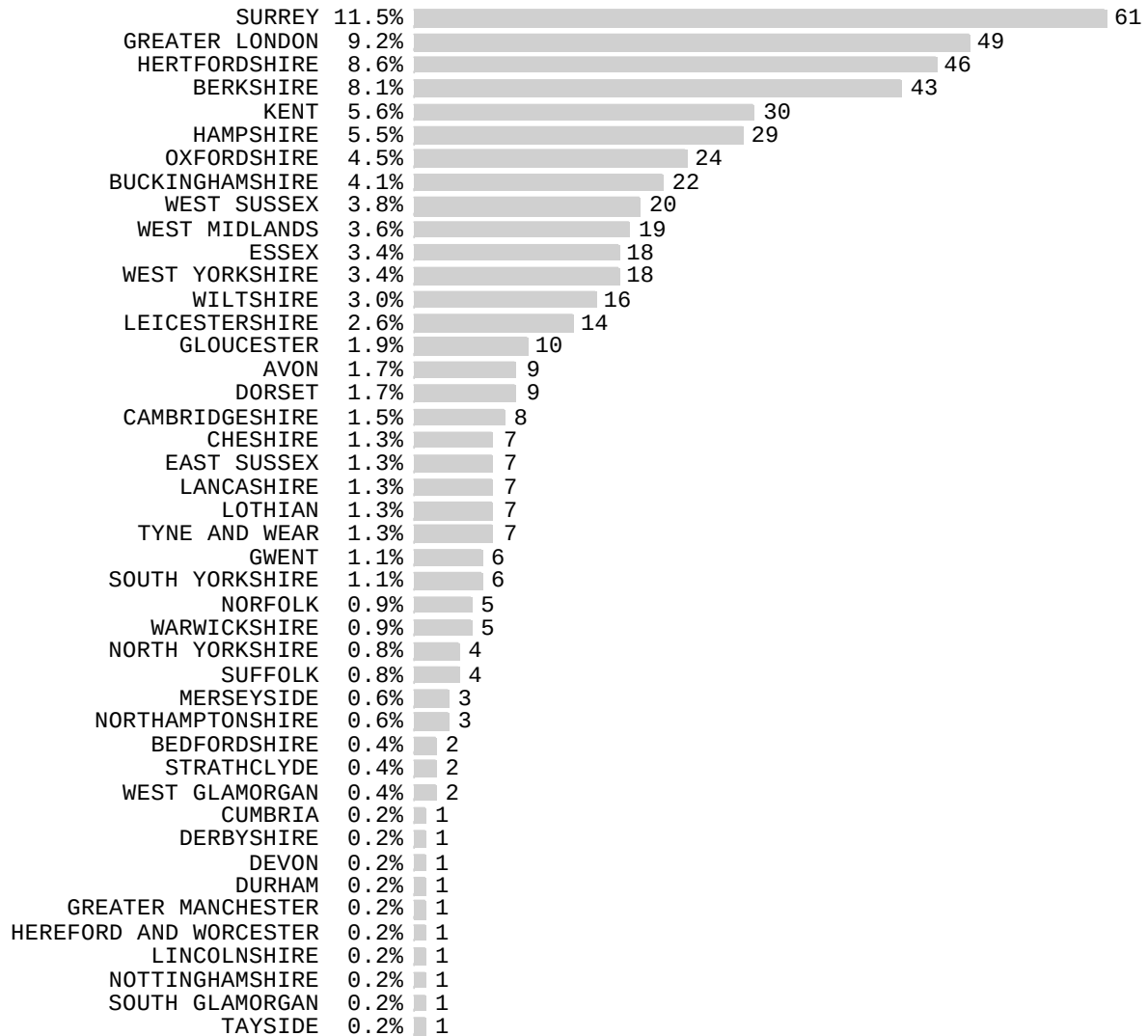


Figure 6. Ranked Bar Chart with Percent of Whole

1999 Distribution of VIEWS Members by County

County, Percent of Total Members, and Members



Total All 44 Counties = 532

Figure 7. Some of the Browser-Safe Colors

Shades of Basic Safe Hues with RGB Codes, Including White (CXFFFFFFF)

CX330000	CX333300	CX003300	CX003333	CX000033	CX330033	CXFFFFFF
CX660000	CX666600	CX006600	CX006666	CX000066	CX660066	CXFFFFFF
CX990000	CX999900	CX009900	CX009999	CX000099	CX990099	CXFFFFFF
CXCC0000	CXCCCC00	CX00CC00	CX00CCCC	CX0000CC	CXCC00CC	CXFFFFFF
CXFF0000	CXFFFF00	CX00FF00	CX00FFFF	CX0000FF	CXFF00FF	CX000000
CXFF3333	CXFFFF33	CX33FF33	CX33FFFF	CX3333FF	CXFF33FF	CX333333
CXFF6666	CXFFFF66	CX66FF66	CX66FFFF	CX6666FF	CXFF66FF	CX666666
CXFF9999	CXFFFF99	CX99FF99	CX99FFFF	CX9999FF	CXFF99FF	CX999999
CXFFCCCC	CXFFFFCC	CXCCFFCC	CXCCFFFF	CXCCCCFF	CXFFCCFF	CXCCCCCC

Not all shades of the same hue are easily distinguishable

Other Safe RGB Colors

CX6633CC	CX9966FF	CX663300	CX996633	CXCC9966	CXFFCC99	CX9900CC
CX663333	CX996666	CXCC9999	CX999966	CXCCCC99	CXCC6600	CXFF9933
CX663366	CX996699	CXCC99CC	CX99CC00	CXCCFF33	CX336666	CX669999

bessler@execpc.com - 22Jan2001

The Power to Show

Chapter 3. Show Them the Trend, Show Them the Numbers: %TRNDANNO Macro for Collision-Free Automatic Annotation of Trend Plots

LeRoy Bessler, Bessler Consulting & Research, bessler@execpc.com

Abstract & Introduction

Images assist faster decision-making, but details assist more reliable decision-making. An annotated graph does both.

After you have already graphed a trend line, you can consult the manual, study fifteen different POSITION variable options for the Annotate facility, and write a SAS/GRAPH® PROC GPLOT program to annotate it, and perhaps do it without a huge number of frustrating iterations where some of your annotated y-values intersect the line, until you get POSITION exactly right for every data point—at least for today’s edition of the data. Instead, you can point my %TRNDANNO macro at any data, specify some options, and get a collision-free annotated trend line for every data point. You don’t need to be an Annotate or GPLOT expert.

%TRNDANNO also supports “Sparse Annotation”. You may (rightly) feel that knowledge of precise values for some trend points is unimportant. The macro lets you optionally annotate just the Start, End, Maximum, and Minimum y-values. It also lets you instead specify a list of x-values for which you want the y-value annotated—only certain dates might have x-values of significant interest. You can Show Them What’s Important.

The POINTLABEL option for SYMBOL statements with PROC GPLOT does not provide reliably collision-free annotation. If the plot is very jagged and dense, successive x positions may be so close together that even TRNDANNO cannot annotate them without collisions. But TRNDANNO succeeds much more often than POINTLABEL, and TRNDANNO supports the option of Sparse Annotation, which may eliminate collisions when the default of full annotation is infeasible even for TRNDANNO.

Macro Functions and Use

Code for the %TRNDANNO macro and example invocations is given in the Appendix. I developed the macro under Version 8, but did not knowingly use anything incompatible with earlier SAS® versions. For details of macro syntax and SAS/GRAPH, please consult vendor documentation. The focus here is on how to use the macro. Here is a sample invocation (except for *D*, *V*, *H*, *F*, and *L*, all values are same as the macro defaults—if they are acceptable, normally do not re-specify them):

```
%trndanno(data=D, yvar=V, xvar=H,
annotype=FULL, xvalpick=,
dotsize=MEDIUM, axislins=NO,
horder=, htkvals=, htkfmt=,
vtrigger=NO, vaxsvals=NO,
vaxszero=NO, vaxsmax=, vaxsmin=,
annofont=NONE, annocol=black,
annosize=1, annofmt=F, annolen=L)
run;
```

D is the name of the SAS dataset whose content is to be plotted. *V* (*H*) is the name of SAS variable to be plotted in the vertical (horizontal) direction. *F* is any of the SAS formats. *L* is the length of the formatted output that will be produced by SAS format *F*. It is the user’s obligation to adjust *L* appropriately for the *F* chosen.

Use of **xvalpick** is demonstrated in invocation code for Figure 2. When **xvalpick** is not used (i.e., it is equated to no value, as shown in the sample above), the user has annotation type options of FULL (the default) or SPARSE, as demonstrated in invocation code for Figures 1 and 3, respectively. Plot dot size options are SMALL, MEDIUM (the default), and LARGE. The font, color, and size selected for annotated values (**annofont**, **annocol**, and **annosize**) are also used for the values listed along the horizontal axis, and for values along the vertical axis, if any.

axislins=YES turns on axis lines. **vaxsvals=YES** displays values along the (visible or invisible) vertical axis line, but only the maximum and minimum values. **vaxsmax=** and **vaxsmin=** override the data’s maximum and minimum for the (visible or invisible) vertical axis line. But if the data’s actual maximum (minimum) value is greater (less) than the user-specified value, the actual value will be used. **vaxszero=YES** forces the vertical axis to start at zero. If **vaxsmax** is also specified, it determines the top of the vertical axis; if not, then the data’s actual maximum is used. It is *not* necessary to turn on axis lines or to display values along the vertical axis in order to use the **vaxsmax**, **vaxsmin**, or **vaxszero** parameters. However, if **vtrigger=YES**, use of **vaxsmax**, **vaxsmin**, or **vaxszero** automatically induces **vaxsvals=YES**. The **horder** and **htkvals** options are available to explicitly assign parameters for the ORDER= and VALUE= in the PLOT statement built by the macro. For SPARSE annotation, or when **xvalpick** is used, it is mandatory to either use **horder=** and **htkvals=**, or to specify the x-value format with **htkfmt=**.

Why vertical axis control when the y values are annotated? First, you may wish to suppress misleading apparently steep growth or decline, or apparently wild fluctuations—to prevent unjustified enthusiasm or anxiety. Second, you may have multiple separate charts that you need to compare—it is impossible to do reliably collision-free automated annotation on a multi-line overlay plot.

Caveats for This Edition of the Macro

The macro embodies design choices that make sense to the author. The focus is on collision-free automated annotation. It could be enhanced to support further customizability. The macro does not edit invocation parameters extensively. When you install the macro, set the defaults to suit yourself or your site. The macro works with *typical* trend data, where there is only one value of y for any x. It is not for a “Step Chart”.

Notices

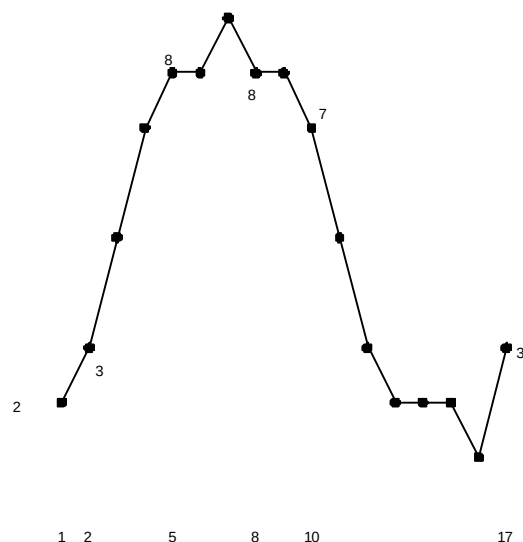
SAS/GRAPH®, SAS®, and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Author Information

LeRoy Bessler, Ph.D.
Bessler Consulting & Research
PO Box 96
Milwaukee, WI 53201-0096, USA
bessler@execpc.com
1-414-351-6748

Dr. LeRoy Bessler is a SAS consultant, with over twenty years of experience with SAS software and other information technology, in a variety of applications and industries. He has a special interest in macro-based Software-Intelligent Application Development, to deliver solutions that are reliable, reusable, maintainable, and extendable. LeRoy has presented in England, Ireland, and the USA on topics in application development, and in the areas of graphic design for effective communication, data visualization, visual communication, color, and InfoGeographics.

Figure 2. Selectively Annotated Trend Line
Using %TRNDANNO(XVALPICK=%str(2 5 8 10),...)
Annotates only y values
for start, end, and x values 2, 5, 8, 10
Automatically identifies only those six x values



The Power to Show

Figure 1. Fully Annotated Trend Line

All possible 3-point / 2-segment trend transitions:
up-steeper up-same-slope up-less-steep up-flat flat-up
up-down down-flat flat-down down-steep down-same-slope
down-less-steep down-flat flat-flat down-up

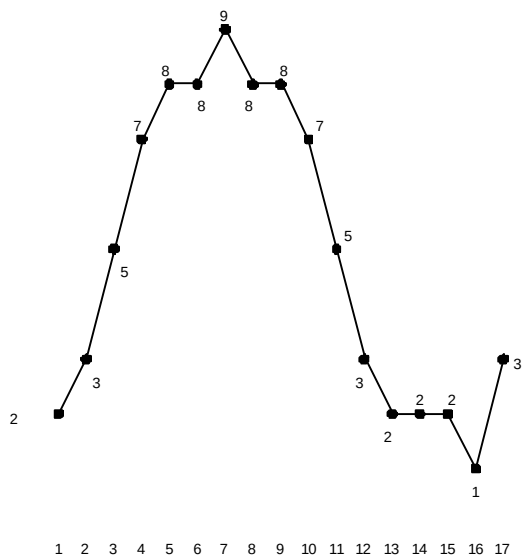
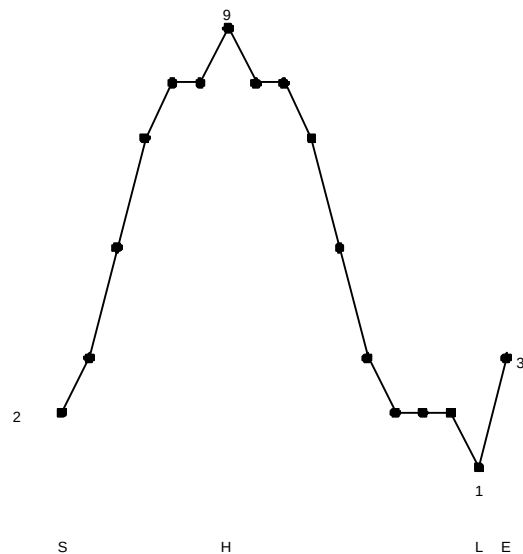


Figure 3. Sparsely Annotated Trend Line

Using %TRNDANNO(ANNOTYPE=SPARSE,...)
Annotates only start, end, maximum, minimum y values
Automatically identifies only corresponding x values
But HTKVALS= overrides the default x tickmark values



Appendix. Code Listing for Macro Source and Sample Invocations *(code not exhaustively tested)*

```
%macro trndanno(data=,yvar=,xvar=,dotsize=MEDIUM,annotype=FULL,xvalpick=,
    horder=,htkvals=,htkfmt=,
    vtrigger=NO,axislins=NO,vaxsvals=NO,vaxszero=NO,vaxsmax=,vaxsmin=,
    annofont=NONE,annocol=BLACK,annosize=1,annofmt=,annolen=);
%let errorflg = N;
%let annotype = %upcase(&annotype);
%if "%substr(&annofont,1,1)" ne "" %then %do;
    %if %upcase(%substr(&annofont,%length(&annofont),1)) eq U %then %do;
        %put ERROR: Uniform SAS/GRAPH Software Fonts NOT SUPPORTED by TRNDANNO Macro;
        %let basefont = %substr(&annofont,1,%eval(%length(&annofont)-1));
        %put ERROR: ANNOFONT value &annofont interpreted as uniform version of font
&basefont;
        %goto macexit;
    %end;
%end;
%if (&annotype eq SPARSE) or (%length(&xvalpick) gt 0) %then %do;
    %if %length(&htkfmt&htkvals) eq 0 %then %do;
        %put ERROR: When ANNOTYPE is not FULL, either HTKFMT or HTKVALS must be
specified;
        %goto macexit;
    %end;
%end;
data y_vs_x(drop=&yvar &xvar); set &data; y=&yvar; x=&xvar; run;
proc sort data=y_vs_x; by x; run;
data y_vs_x(drop=prevy prevx);
retain prevy prevx;
set y_vs_x;
if _N_ gt 1 then do;
    if x eq prevx then do;
        put 'ERROR: More than one y value for an x value NOT SUPPORTED by TRNDANNO
Macro';
        put 'ERROR: For ' x= 'TRNDANNO Macro found ' prevy= 'and ' y=;
        call symput('errorflg','Y');
        stop;
    end;
end;
prevy=y; prevx=x;
run;
%if &errorflg eq Y %then %goto macexit;
%if %upcase(&dotsize) eq LARGE %then %let sym_no = 3;
%else
%if %upcase(&dotsize) eq SMALL %then %let sym_no = 1;
%else
%let sym_no = 2;
%let xoffsetl = %eval(&annolen + 2);
%let xoffsetr = %eval(&annolen + 2);
%let textlen = %eval(&annolen + 1);
%if %upcase(&annofont) eq NONE %then %do;
    %let xtralpad = NO;
%end;
%else %do;
    %let xtralpad = YES;
    %let textlen = %eval(&textlen + 1);
%end;
%if %upcase(&axislins) eq YES %then %let axistyle = 1;
%else
%let axistyle = 0;
%let vaxsvals = %upcase(&vaxsvals);
%let vaxszero = %upcase(&vaxszero);
%if &vaxsvals eq YES
    or
    &vaxszero eq YES
    or
    %length(&vaxsmax&vaxsmin) ne 0
%then %do;
```

```

    %if &vtrigger eq YES %then %let vaxsvals = YES;
proc means noprint data=y_vs_x min max;
var y;
output out=min_max min=ymin max=ymax;
run;
data _null_;
set min_max;
    %if %length(&vaxsmin) ne 0 %then %do;
ymin = min(&vaxsmin,ymin);
    %end;
    %if %length(&vaxsmax) ne 0 %then %do;
ymax = max(&vaxsmax,ymax);
    %end;
    %if &vaxszero eq YES %then %do;
if ymin gt 0 then ymin = 0;
else
if ymax lt 0 then ymax = 0;
    %end;
call symput('vaxismin',trim(left(put(ymin,&annofmt)))));
call symput('vaxismax',trim(left(put(ymax,&annofmt)))));
call symput('cntlaxis','YES');
run;
%end;
%else %let cntlaxis = NO;
data annodata(drop=b4prevy prevy b4prevx prevx holdy holdx shftleft);
length b4prevy prevy b4prevx prevx holdy holdx 8;
retain b4prevy prevy b4prevx prevx 0;
format y &annofmt;
length text $ &textlen;
length function style color $ 8 size 3
    position xsys ysys hsys when $ 1;
retain function 'label' style "&annofont" color "&annocol" size &annosize
    xsys ysys '2' hsys '4' when 'a';
set y_vs_x end=lastobs;
if _N_ eq 1 then do;
    call symput('first_x',trim(left(x)));
    prevx=x; prevy=y;
end;
else
if _N_ eq 2 then do;
    holdx=x; holdy=y;
    position = '6';
    text=
%if &xtralpad eq YES %then %do;
        ' ' ||
%end;
        ' ' || put(prevy,&annofmt);
    xsys='1'; hval=prevx; x=0; y=prevy;
    output;
    xsys='2'; b4prevx=prevx; b4prevy=prevy; prevx=holdx; prevy=holdy;
end;
else do;
    holdx=x; holdy=y;
    if _N_ gt 2 then do;
        shftleft = 'Y';
        if      b4prevy eq prevy eq y then position='2'; /* flat-flat */
        else if b4prevy eq prevy gt y then position='3'; /* flat-down */
        else if b4prevy gt prevy gt y then do;           /* down-down */
            if (prevy-b4prevy)/(prevx-b4prevx) ge (y-prevy)/(x-prevx)
            then do; position='3'; shftleft='N'; end;
            else position='7';
        end;
        else if b4prevy gt prevy lt y then position='8'; /* down-up */
        else if b4prevy lt prevy lt y then do;           /* up-up */
            if (prevy-b4prevy)/(prevx-b4prevx) le (y-prevy)/(x-prevx)
            then do; position='9'; shftleft='N'; end;

```

```

        else position='1';
    end;
    else if b4prevy lt prevy gt y then position='2'; /* up-down */
    else if b4prevy gt prevy eq y then position='7'; /* down-flat */
    else if b4prevy eq prevy lt y then position='9'; /* flat-up */
    else if b4prevy lt prevy eq y then position='1'; /* up-flat */
    if shftleft eq 'Y'
    then text=trim(left(put(prevy,&annofmt)));
    else text=' ' || put(prevy,&annofmt);
    hval=prevx; x=prevx; y=prevy;
    output;
    if lastobs then do;
        call symput('last_x',trim(left(holdx)));
        call symput('nmbrhtks',trim(left(_N_)));
        hval=holdx; x=holdx; y=holdy;
        position = '6';
        text=
%if &xtralpad eq YES %then %do;
        ' ' ||
%end;
        ' ' || put(y,&annofmt);
        output;
    end;
    end;
    b4prevx=prevx; b4prevy=prevy; prevx=holdx; prevy=holdy;
end;
run;
%if &annotype eq SPARSE %then %do;
proc means data=annodata noprint min max;
var y;
output out=minmax min=ymin max=ymax;
run;
data xtoanno(keep=x);
set annodata end=lastobs;
if _N_ eq 1 then do;
    output;
    set minmax;
end;
if y eq ymin then output;
if y eq ymax then output;
if lastobs then output;
run;
proc sort data=xtoanno nodupkey; by x; run;
data annovals;
merge annodata xtoanno(in=doit); by x; if doit; run;
%end;
%else %if %length(&xvalpick) ne 0 %then %do;
data annovals;
set annodata end=lastobs;
if (_N_ eq 1) or (x in (&xvalpick)) or (lastobs);
run;
%end;
%else %do;
data annovals; set annodata; run;
%end;
%if (&annotype eq SPARSE) or (%length(&xvalpick) gt 0) %then %do;
data forticks;
merge y_vs_x(in=plotpt rename=(x=hval)) annovals(in=annopt);
by hval;
if plotpt;
call symput('hord' || left(_N_),hval);
if not annopt then call symput('htk' || left(_N_), ' ');
else
    call symput('htk' || left(_N_),trim(left(put(hval,&htkfmt))));
run;
%end;
%macro varlist(varprfx=,varcount=);

```

[illegible]