

Paper 278-2011

Time-Critical Concise Reporting in the Age of Twitter and Wireless Mobility: Continuous Real-Time Monitoring, Sparse Line Charts, and Compact Tables

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

Abstract

According to its web site, “Twitter is without a doubt the best way to share and discover what is happening right now.” This paper will show you that SAS® is an excellent way to discover and share what is happening right now.

You can keep in touch with your data by creating a program to continuously monitor critical values and, when appropriate, send email alerts to people with a need to know. The email can optionally include a link to, attach, or imbed a tabular or graphic report.

What’s happening? Are we up? Are we down? What direction are we going? I presented my sparse line chart concept in 1991, long before the “spark line” emerged. I have long claimed that usually the only important aspects of a trend, besides current direction, are where it started, where it is now, and its prior high and low, if any. The paper includes several sparse line charts, some developed for the BlackBerry, iPhone, etc.

The subsetted ranking report “shows you the most with the least” by implementing my long advocated design mantra of “Show Them What’s Important” and the concept of “Let Part Stand for the Whole”. The subsetted ranking report can also be implemented as a horizontal bar chart.

The transposed rows report provides a “twist”, literally and figuratively, on the traditional default wide display of columns, which often entails the inconvenient, anti-communicative requirement for sideways scrolling on a web page.

Introduction

Back in 1987, when asked to replace an unwieldy and nearly unusable report to executive management, I developed the concept implemented in the SubsettedRankingReport macro below to “Show Them What’s Important” and “Let Part Stand for the Whole”.

Much more recently I created the TransposedPrint macro as a convenience to be able to easily inspect the content of a data set. This macro presents the column content within each row in a vertical stack, rather than spread across the display area in the traditional manner.

In 2009 I reported about my tool to monitor various measures of user activity on a SAS BI server, and in 2010 I reported about my adaptation and extension of it to specifically monitor CPU consumption and send email alerts to anyone whose SAS process might be in a loop. I don’t present that code here, but rather a simple, derivative model program that provides the logical structure that you can adapt and use to Alert Them When They Need To Know about any particular measure(s) of interest that your application beneficiaries care about.

Except for the MonitorAndAlert macro, which produces only an email message (but could be adapted to send an attachment or to include a link, or at least identify the path, to a file of more information), examples of output are included.

Alert Them When They Need To Know

More important than the fashionable concept of “dashboards” is the act of notifying any interested and/or responsible party immediately when there is a data situation of concern.

Since SAS can send email whenever you want it to, and since SAS can go to sleep when you don't want it to do anything, it is easy to create a system that does continuous (at intervals) monitoring and notifies people if a threshold has been reached or a goal has been missed. I first developed such an application to notify users of a shared SAS BI server whenever their SAS programs had consumed more than a defined amount of CPU time. (See Reference 1.)

The program below is a much simplified example of logic to monitor data and send email alerts.

For a real application, replace the TO and SUBJECT assignments in the FILENAME EMAIL statement, and insert your monitor logic at the point indicated. With SAS email you can send messages to a distribution list, use CC and BCC, attach files, imbed a URL in the message body, etc. For more about SAS email, see Reference 2 and/or the SAS Companion Online Doc for your operating system.

```
%macro MonitorAndAlertDemo(WaitInSeconds=60,NumberOfChecksIfNotForever=);

FILENAME anyname EMAIL
TO = 'Le_Roy_Bessler@wi.rr.com'           /* customize this */
SUBJECT = "Too Many Observations in Data Set"; /* customize this */

%let NumberOfChecks = 0;

%Check:

* START your monitor logic here *;

data _null_;
set sasuser.DataSetToChk nobs=Count;
call symput('ObsCount',trim(left(Count)));
stop;
run;

%if %eval(&ObsCount GT 1)
%then %do;
data _null_;
file anyname;
put "Number of Observations in Data Set is &ObsCount";
run;
%end;

* END your monitor logic here *;

%if %length(&NumberOfChecksIfNotForever) NE 0 %then %do;
  %let NumberOfChecks = %eval(&NumberOfChecks + 1);
  %if &NumberOfChecks EQ &NumberOfChecksIfNotForever
  %then %goto MacroExit;
%end;

data _null_;
WaitUntilNextCheck = sleep(&WaitInSeconds);
run;

%goto Check;
```

```
%MacroExit;

%mend MonitorAndAlertDemo;

options mprint;

* START of test that will generate NO alert *;

data sasuser.DataSetToChk;
x = 1;
output;
run;

%MonitorAndAlertDemo(WaitInSeconds=10,NumberOfChecksIfNotForever=1);

* END of test that will generate NO alert *;

* START of test that will generate AN alert *;

data sasuser.DataSetToChk;
x = 1;
output;
x = 1;
output;
run;

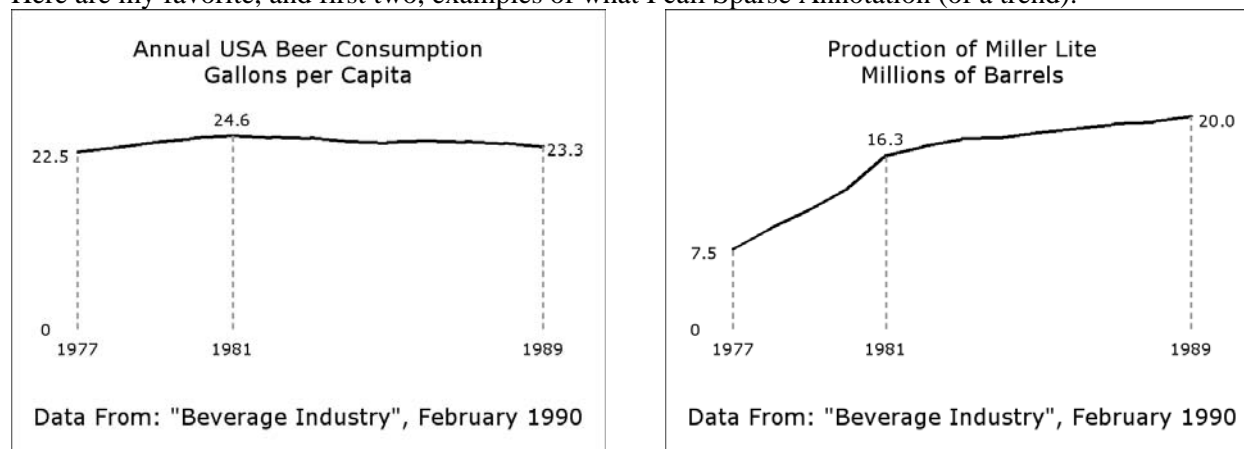
%MonitorAndAlertDemo(WaitInSeconds=10,NumberOfChecksIfNotForever=1);

* END of test that will generate AN alert *;
```

Sparse Line Charts: Sparse Annotation = Visual + Only Detail That Is Essential for a Trend Plot

I put the word “Essential” into the title of this section for this reason. I am convinced that, most of the time, there are only four points in a trend that are of real interest: Start, End, Minimum, and Maximum. There are, of course, other points of potential interest: (a) points when there was some significant environmental event which could be expected (or hoped?) to influence the trend; and (b) points not a minimum or maximum but where the trend experienced a permanent change in slope. Case (b) includes a point of inflection, of which there are four possibilities: (1) increase at an increasing rate; (2) increase at a decreasing rate; (3) decrease at a decreasing rate; and (b) decrease at an increasing rate. For purposes of computer-generated trend plots, it is trivially easy with SAS software to programmatically identify Start, End, Minimum, and Maximum. For me, at least so far, it has not been possible to programmatically discover points of inflection in trend data. I suppose that if one knew how to reliably fit a continuous function to a set of data points, there might be some way to discover points of inflection.

Here are my favorite, and first two, examples of what I call Sparse Annotation (of a trend):



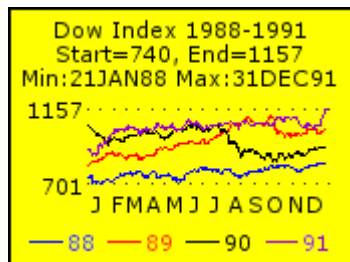
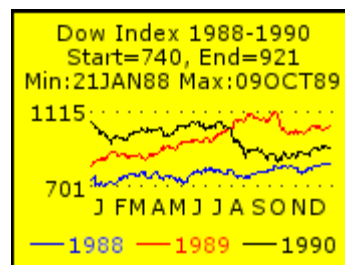
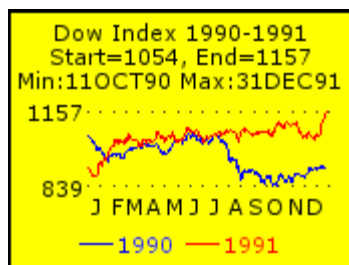
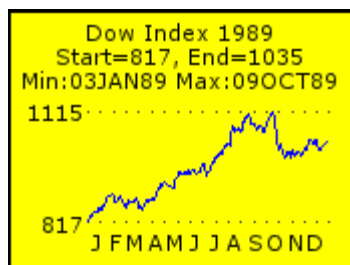
As you can guess from the dates of the data, I coined my graphic design term “Sparse Annotation” a long time ago, but the utility of these two examples does not diminish with age. The national trend had only three points of interest: Start, End, and Maximum. The Miller Lite trend had only three points of interest: Start, End, and a point where the rate of growth permanently, at least over the range of this graph, changes.

These two designs also embody one of my preferred design rules for trends: whenever your audience will let you, and when your target medium is not a tiny screen, start the vertical axis at zero. Yes, this does diminish the viewer’s ability to discern small changes. However, at the same time it diminishes the propensity for needless elation or despair about what might be insignificant fluctuations. The flat appearance of the national trend is not a failure of graphic communication—industry observers at the time described the trend as “flat”. The best way to assess the significance of a change is to look at the precise numbers and the percent of change and the impact of that data on the business, the organization, or the process.

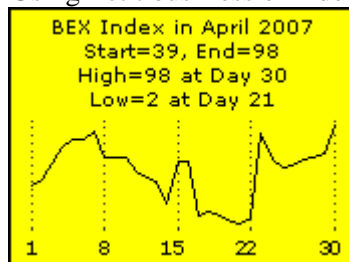
Sparse Line Charts for the Small Screen (iPhone, BlackBerry, etc.): Which Also Would Work Well on the Large Screen

Because I wanted to work with input data that any SAS user has available, I used the SASHELP.CITIDAY data set, which is four-plus years of daily data. It involves trading days, not 365/366 calendar days. So, there is a little bit of extra data preparation, which is unique to the data used. However, to apply the code to your own presumably more typical daily data would make your adapted code simpler.

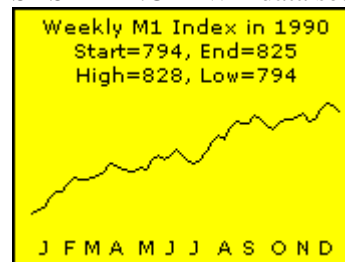
Three years ago I faced the challenge of developing code that could present readable trends on the small screen—BlackBerry, iPhone, etc. The size of the graph in pixels had to be limited to 172 X 129. The results are presented below, in Actual Size. Code for month of days and for the weekly graph is in Reference 3. Code for the other four examples all involves the macros and the preparation and invocation code presented here.



Using fictitious Bessler Index



Using MF3505 variable in SASHELP.CITIWK data set



Here is the code used to create the four sparse line charts for daily trading data:

```
ods noresults;
ods listing;

data work.DowByDay;
set sashelp.citiday(keep=date snydjcm where=(snydjcm ne .));
Year = year(date);
run;

data work.DowByTradingDay(drop=FirstTradingDayInYearFound snydjcm);
retain FirstTradingDayInYearFound 'N' TradingDay 0;
length MonthAbbrev $ 1;
set work.DowByDay end=LastOne;
by Year;
if First.Year then do;
  TradingDay = 1;
  FirstTradingDayInYearFound = 'Y';
end;
else do;
  TradingDay + 1;
  if Last.Year
  then FirstTradingDayInYearFound = 'N';
end;
Dow=round(snydjcm,1);
Month = month(date);
MonthAbbrev=substr(left(put(date,monname3.)),1,1);
run;

%macro GetHtickmarkValues;

%do i = 1 %to &NumberOfTickMarks %by 1;
  "&htickvalue&i"
%end;

%mend GetHtickmarkValues;

%macro IdentifyYaxisCriticalPoints(data=,datevar=,yVar=,yVarFormat=);

%global yVarStart yVarEnd;

proc sort data=&data out=SortedByDate;
by &datevar;
run;

data _null_;
```

```

set SortedByDate end=LastOne;
if _N_ eq 1
then call symput('yVarStart',trim(left(put(&yVar,&yVarFormat))));
else
if LastOne
then call symput('yVarEnd',trim(left(put(&yVar,&yVarFormat))));
run;

proc means data=&data min max range noprint;
var &Yvar;
output out=MinMaxRange min=yMin max=yMax range=yMaxMinDiff;
run;

%global
  yAxisStartTickMarkValue  yAxisStartTickMarkDisplay
  yAxisEndTickMarkValue    yAxisEndTickMarkDisplay
  yAxisIncrement;

data _null_;
set MinMaxRange;
call symput('yAxisStartTickMarkValue',trim(left(yMin          )));
call symput('yAxisStartTickMarkDisplay',trim(left(put(yMin,&yVarFormat))));
call symput('yAxisEndTickMarkValue',trim(left(yMax          )));
call symput('yAxisEndTickMarkDisplay',trim(left(put(yMax,&yVarFormat))));
call symput('yAxisIncrement',trim(left(yMaxMinDiff)));
run;

data _null_;
set Selected;
if &yVar LT &yAxisStartTickMarkValue
then put &yVar= " is below yAxis Start &yAxisStartTickMarkValue";
if &yVar GT &yAxisEndTickMarkValue
then put &yVar= " is above yAxis End &yAxisEndTickMarkValue";
run;

%mend IdentifyYaxisCriticalPoints;

%macro CreateDayHtickmarkValues(data=,filter=,TradingDaysOrCalendarDays=Calendar);

%if %upcase(&TradingDaysOrCalendarDays) EQ TRADING
%then %let MiddleDayOfMonth = 11;
%else %let MiddleDayOfMonth = 15;

* Maximum number of trading days in a year is 253. *;
* Possible extra GLOBAL statements are harmless. *;
%do i = 1 %to 366 %by 1;
  %global htickvalue&i;
%end;

* For the actual number of trading days found: *;
%global NumberOfTickMarks;

data Selected;
set &data;
%if %length(&filter) ne 0 %then %do;
where &filter;
%end;
run;

data _null_;
retain DayInMonth 0;
set Selected end=LastOne;
by Month;
if first.Month
then DayInMonth = 1;
else DayInMonth = DayInMonth + 1;
* The monthly average number of trading days is 22. *;
* Want to put the month abbrev label at the mid-month. *;
if DayInMonth eq &MiddleDayOfMonth
then call symput('htickvalue' || trim(left(_N_)),MonthAbbrev);
else call symput('htickvalue' || trim(left(_N_)),');
if LastOne;
call symput('NumberOfTickMarks',trim(left(_N_)));
run;

```

```

%mend CreateDayHtickmarkValues;

%macro IdentifyXdatesForCriticalPoints(data=,yVar=,xDateVar=);

%global LocateMinAndOrMaxWhenUnique;

data _null_;
length MinLoc MaxLoc $ 11;
retain HowManyXfoundForMinY HowManyXfoundForMaxY 0;
retain xForMinY . xForMaxY .;
set &data end=LastOne;
if &Yvar EQ &yAxisStartTickMarkValue
then do;
  if HowManyXfoundForMinY eq 0
  then do;
    HowManyXfoundForMinY = 1;
    xForMinY = &xDateVar;
  end;
  else HowManyXfoundForMinY + 1;
end;
else
if &Yvar EQ &yAxisEndTickMarkValue
then do;
  if HowManyXfoundForMaxY eq 0
  then do;
    HowManyXfoundForMaxY = 1;
    xForMaxY = &xDateVar;
  end;
  else HowManyXfoundForMaxY + 1;
end;
if LastOne;
put HowManyXfoundForMinY= HowManyXfoundForMaxY= xForMinY= xForMaxY=;
if HowManyXfoundForMinY eq 1
then MinLoc = "Min:" || trim(left(put(xForMinY,date7.)));
else MinLoc = " ";
if HowManyXfoundForMaxY eq 1
then MaxLoc = "Max:" || trim(left(put(xForMaxY,date7.)));
else MaxLoc = " ";
call symput('LocateMinAndOrMaxWhenUnique',trim(left(MinLoc)) || " " || trim(left(MaxLoc)));
run;

%mend IdentifyXdatesForCriticalPoints;

options mprint;

%let FontSize = 8pt;          /* Use 9pt for SAS/GRAPH® 9.1.3, 8pt for SAS/GRAPH 9.2 */
/* %let FontSize = 4PCT; If using 800 X 600 default dimensions of GIF or PNG */
%let LineWidth = 1;          /* Use a thicker line for 800 X 600 images if it looks OK */
%let ReferenceLineType = 35; /* Use 33 or 20 for 800 X 600 images */

goptions reset=all;
goptions device=GIF;
goptions xpixels=172 ypixels=129; /* defaults are xpixels=800 ypixels=600
  which are the proportions of screens used for meeting projectors and of non-wide-screen monitors */
* goptions xpixels=800 ypixels=600;
goptions border;
goptions cback=CXFFFF00; /* RGB Yellow */
goptions htext=&FontSize ftext='Verdana'; /* used for axis tick mark values */
goptions gsfmode=replace gsfname=anyname;

title1 font=none height=1 ' ';
footnote1 angle=+90 font=none height=1 ' ';
footnote2 angle=-90 font=none height=1 ' ';
/* for multi-line plots, presence of legend creates extra white space at bottom. */
footnote3; /* turn off white space at bottom. */

symbol1 color=CX0000FF interpol=join v=none width=&LineWidth; /* RGB Blue */
symbol2 color=CXFF0000 interpol=join v=none width=&LineWidth; /* RGB Red */
symbol3 color=CX000000 interpol=join v=none width=&LineWidth; /* RGB Black */
symbol4 color=CX9900CC interpol=join v=none width=&LineWidth; /* RGB Purple */

filename anyname "C:\SparseOneLineChartForTheSmallScreen.GIF";

data work.ToPlot;
set work.DowByTradingDay(where=(Year in (1989)));

```



```

run;

%CreateDayHtickmarkValues(data=work.ToPlot,
  filter=%str(Year EQ 1989),
  TradingDaysOrCalendarDays=Trading);

%IdentifyYaxisCriticalPoints(data=work.ToPlot,datevar=date,yVar=Dow,yVarFormat=4.);
/* Run macro above before running macro below. */
%IdentifyXdatesForCriticalPoints(data=work.ToPlot,yVar=DOW,xDateVar=date);

axis1 label=none major=none minor=none style=0
  order=1 to &NumberOfTickMarks by 1
  value=(%GetHtickmarkValues);

axis2 label=none major=none minor=none style=0
  order=&yAxisStartTickMarkValue to &yAxisEndTickMarkValue by &yAxisIncrement
  value=("&yAxisStartTickMarkDisplay" "&yAxisEndTickMarkDisplay");

title2 font='Verdana' height=&FontSize justify=CENTER "Dow Index 1989"
  justify=CENTER "Start=&yVarStart, End=&yVarEnd"; /* justify=CENTER forces new line */
title3 font='Verdana' height=&FontSize
  "&LocateMinAndOrMaxWhenUnique";
footnote3 font=none height=1 ' '; /* add white space at bottom when no legend is used */

proc gplot data=work.ToPlot;
plot Dow*TradingDay=Year / haxis=axis1 vaxis=axis2
  nolegend /* for only one line, no legend is needed */
  vref=&yAxisStartTickMarkValue &yAxisEndTickMarkValue
  cvref=CX333333 lvref=&ReferenceLineType;
/* dark gray reference lines at top and bottom of vaxis range */
run; quit;

filename anyname "C:\SparseTwoLineChartForTheSmallScreen.GIF";

data work.ToPlot;
set work.DowByTradingDay(where=(Year in (1990 1991)));
run;

%CreateDayHtickmarkValues(data=work.ToPlot,
  filter=%str(Year EQ 1991),
  TradingDaysOrCalendarDays=Trading);

%IdentifyYaxisCriticalPoints(data=work.ToPlot,datevar=date,yVar=Dow,yVarFormat=4.);
/* Run macro above before running macro below. */
%IdentifyXdatesForCriticalPoints(data=work.ToPlot,yVar=DOW,xDateVar=date);

axis1 label=none major=none minor=none style=0
  order=1 to &NumberOfTickMarks by 1
  value=(%GetHtickmarkValues);

axis2 label=none major=none minor=none style=0
  order=&yAxisStartTickMarkValue to &yAxisEndTickMarkValue by &yAxisIncrement
  value=("&yAxisStartTickMarkDisplay" "&yAxisEndTickMarkDisplay");

title2 font='Verdana' height=&FontSize justify=CENTER "Dow Index 1990-1991"
  justify=CENTER "Start=&yVarStart, End=&yVarEnd";
title3 font='Verdana' height=&FontSize
  "&LocateMinAndOrMaxWhenUnique";
footnote3; /* no added white space at bottom when a legend is used */

legend1 label=none
  shape=symbol(8,5) /* 8,5 instead of 8,1 mysteriously creates white space below legend */
  value=(color=CX0000FF '1990' /* color the year itself */
  color=CXFF0000 '1991'); /* color of thin lines samples can be hard to interpret */

proc gplot data=work.ToPlot;
plot Dow*TradingDay=Year / haxis=axis1 vaxis=axis2 legend=legend1
  vref=&yAxisStartTickMarkValue &yAxisEndTickMarkValue
  cvref=CX333333 lvref=&ReferenceLineType;
run; quit;

filename anyname "C:\SparseThreeLineChartForTheSmallScreen.GIF";

data work.ToPlot;
set work.DowByTradingDay(where=(Year in (1988 1989 1990)));

```

```

run;

%CreateDayHtickmarkValues(data=work.ToPlot,
  filter=%str(Year EQ 1990),
  TradingDaysOrCalendarDays=Trading);

%IdentifyYaxisCriticalPoints(data=work.ToPlot,datevar=date,yVar=Dow,yVarFormat=4.);
/* Run macro above before running macro below. */
%IdentifyXdatesForCriticalPoints(data=work.ToPlot,yVar=DOW,xDateVar=date);

axis1 label=none major=none minor=none style=0
  order=1 to &NumberOfTickMarks by 1
  value=(%GetHtickmarkValues);

axis2 label=none major=none minor=none style=0
  order=&yAxisStartTickMarkValue to &yAxisEndTickMarkValue by &yAxisIncrement
  value=("&yAxisStartTickMarkDisplay" "&yAxisEndTickMarkDisplay");

title2 font='Verdana' height=&FontSize justify=CENTER "Dow Index 1988-1990"
  justify=CENTER "Start=&yVarStart, End=&yVarEnd";
title3 font='Verdana' height=&FontSize
  "&LocateMinAndOrMaxWhenUnique";
footnote3;

legend1 label=none shape=symbol(8,5)
  value=(color=CX0000FF '1988' color=CXFF0000 '1989' color=CX000000 '1990');

proc gplot data=work.ToPlot;
plot Dow*TradingDay=Year / haxis=axis1 vaxis=axis2 legend=legend1
  vref=&yAxisStartTickMarkValue &yAxisEndTickMarkValue
  cvref=CX333333 lvref=&ReferenceLineType;
run; quit;

filename anyname "C:\SparseFourLineChartForTheSmallScreen.GIF";

data work.ToPlot;
set work.DowByTradingDay(where=(Year in (1988 1989 1990 1991)));
run;

%CreateDayHtickmarkValues(data=work.ToPlot,
  filter=%str(Year EQ 1991),
  TradingDaysOrCalendarDays=Trading);

%IdentifyYaxisCriticalPoints(data=work.ToPlot,datevar=date,yVar=Dow,yVarFormat=4.);
/* Run macro above before running macro below. */
%IdentifyXdatesForCriticalPoints(data=work.ToPlot,yVar=DOW,xDateVar=date);

axis1 label=none major=none minor=none style=0
  order=1 to &NumberOfTickMarks by 1
  value=(%GetHtickmarkValues);

axis2 label=none major=none minor=none style=0
  order=&yAxisStartTickMarkValue to &yAxisEndTickMarkValue by &yAxisIncrement
  value=("&yAxisStartTickMarkDisplay" "&yAxisEndTickMarkDisplay");

title2 font='Verdana' height=&FontSize justify=CENTER "Dow Index 1988-1991"
  justify=CENTER "Start=&yVarStart, End=&yVarEnd";
title3 font='Verdana' height=&FontSize
  "&LocateMinAndOrMaxWhenUnique";
footnote3;

legend1 label=none shape=symbol(8,5)
  value=(color=CX0000FF '88' color=CXFF0000 '89' color=CX000000 '90' color=CX9900CC '91');

proc gplot data=work.ToPlot;
plot Dow*TradingDay=Year / haxis=axis1 vaxis=axis2 legend=legend1
  vref=&yAxisStartTickMarkValue &yAxisEndTickMarkValue
  cvref=CX333333 lvref=&ReferenceLineType;
run; quit;

```

“Show Them What’s Important” and “Let Part Stand for the Whole”

Many years ago when I had to prepare a report for executive management, Kenneth J. Wesley counseled me with this: “If you want them to read it, it must fit on one page.”

Well, the fact of the matter is that usually a huge percent of what matters can be fit in a subsetting ranking report, with ranking from high to low if high is good, or vice-versa.

There are three ways to subset ranked data:

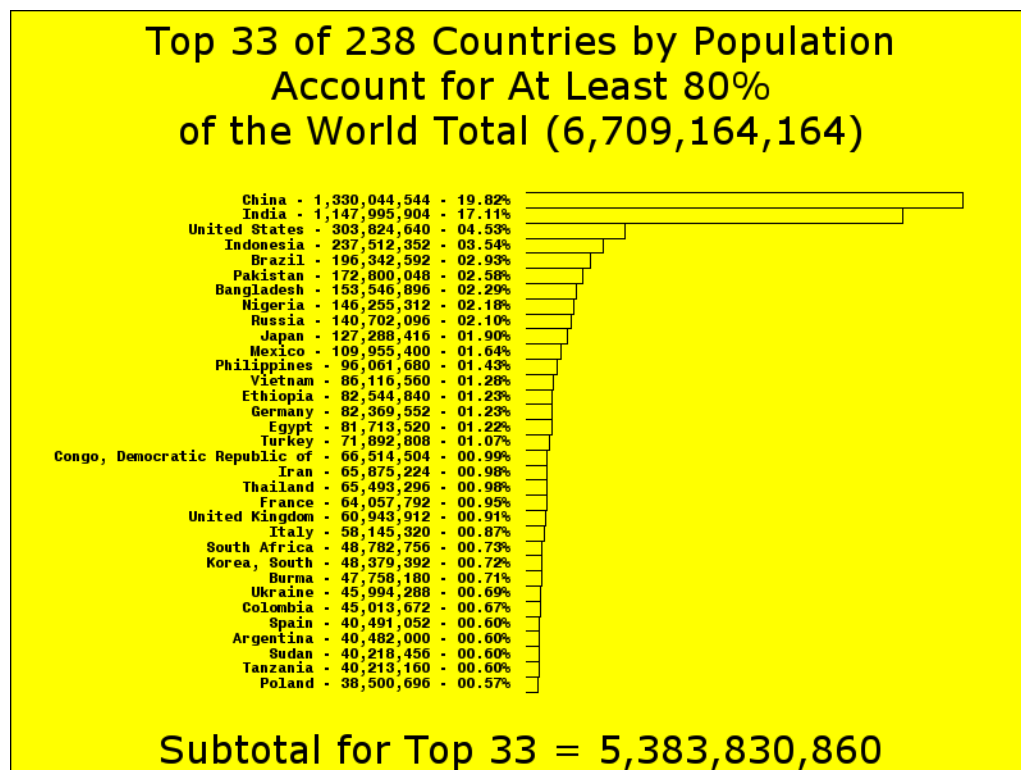
- The Top N;
- The Top P Percent (Show them as many observations from the top of the ranked list until you account for at least P Percent of the grand total of the measure of interest);
- Every observation where the measure of interest exceeds a threshold or fails to meet a goal.

Ranking can be from high to low, or low to high, depending on whether large or small is good.

On the following page are examples of these three types of reports. A listing of the SubsettingRankingReport macro and the macro invocations used to produce the examples follow.

Note that the design of the reports and the construction of the macro is such that the reader not only gets the display of what’s important, but also the reader is told the grand total of the measure of interest, the grand total number of observations, and what part of these grand totals are accounted for by the subset that is presented. It is important to anticipate and answer questions as to how much has been left out.

For an adaptation of this concept to a horizontal bar chart, please see Reference 4. For this Top 80% chart, revise that code with: axis1 value=(font='SAS Monospace/Bold' justify=right); hbar ... maxis=axis1; The horizontal bar chart alternative, in effect, simply adds a visual comparator to the tabular listings below. Prior to Version 9.2, to deliver the long label in this example (i.e., Congo) would require use of SAS/GRAPH Annotate.



Case 1. Top N Report

Top 10 Shoe Sales By City

This subset accounts for 50.6% of Total Shoe Sales

Grand Totals: City Count = 53, Shoe Sales = \$33,851,566

Rank	City	Shoe Sales
1	Vancouver	\$3,227,768
2	Tel Aviv	\$2,567,568
3	Kingston	\$2,235,204
4	Dubai	\$1,910,544
5	Chicago	\$1,565,585
6	New York	\$1,489,207
7	Al-Khobar	\$1,153,667
8	Minneapolis	\$1,099,937
9	Heidelberg	\$967,739
10	Lisbon	\$898,345
		=====
		\$17,115,564

Case 2. Top P Percent Report

Top 25 Shoe Sales By City

Selected to list the Top 80% of Total Shoe Sales
(This subset actually accounts for 80.5%)

Grand Totals: City Count = 53, Shoe Sales = \$33,851,566

Rank	City	Shoe Sales
1	Vancouver	\$3,227,768
2	Tel Aviv	\$2,567,568
3	Kingston	\$2,235,204
4	Dubai	\$1,910,544
5	Chicago	\$1,565,585
6	New York	\$1,489,207
7	Al-Khobar	\$1,153,667
8	Minneapolis	\$1,099,937
9	Heidelberg	\$967,739
10	Lisbon	\$898,345
	. . .	
21	Paris	\$621,877
22	Seattle	\$611,945
23	Moscow	\$560,976
24	Montreal	\$531,028
25	La Paz	\$530,506
		=====
		\$27,258,392

Case 3. Threshold Report

Ranked Shoe Sales not less than \$1,000,000 By City

This subset accounts for 45.0% of Total Shoe Sales

Grand Totals: City Count = 53, Shoe Sales = \$33,851,566

Rank	City	Shoe Sales
1	Vancouver	\$3,227,768
2	Tel Aviv	\$2,567,568
3	Kingston	\$2,235,204
4	Dubai	\$1,910,544
5	Chicago	\$1,565,585
6	New York	\$1,489,207
7	Al-Khobar	\$1,153,667
8	Minneapolis	\$1,099,937
		=====
		\$15,249,480

Here is the code used to produce the reports:

```

%MACRO SubsettedRankingReport
(DATA=,
CLASSVAR=, CVARLABL=,
RANKVAR=, RVARFMT=, RVARLABL=,
NN=, MINVALUE=, PERCENT=,
TITLESUFFIX=, DateAndPageNumber=NO);

PROC SORT DATA=&DATA(keep=&CLASSVAR &RANKVAR)
          OUT=RANKED;
BY DESCENDING &RANKVAR;
RUN;

PROC MEANS DATA=&DATA NOPRINT SUM N;
VAR &RANKVAR;
OUTPUT OUT=TOTSTATS SUM=TOTSUM N=TOTNN;
RUN;

DATA SUBSETTED;
RETAIN TOPSUM TOPPCT 0 TOTSUM;
SET RANKED END=LAST;
IF &_N_ EQ 1 THEN DO;
  SET TOTSTATS;
  CALL SYMPUT('TOTSUM', trim(left(put(TOTSUM, &RVARFMT))));
  CALL SYMPUT('TOTNN', trim(left(TOTNN)));
END;
RANK = &_N_;
%if %length(&MINVALUE) ne 0
%then %do;
IF &RANKVAR GE &MINVALUE
THEN DO;
  TOPSUM = TOPSUM + &RANKVAR;
END;
ELSE DO;
  TOPPCT = (TOPSUM / TOTSUM) * 100;
  CALL SYMPUT('TOPPCT', trim(left(put(TOPPCT, 5.1))));
  CALL SYMPUT('TOPSUM', trim(left(put(TOPSUM, &RVARFMT))));
  CALL SYMPUT('TOPNN', trim(left(&_N_ - 1)));
  STOP;
END;
%end;
%else
%if %length(&PERCENT) ne 0
%then %do;
TOPSUM = TOPSUM + &RANKVAR;
TOPPCT = (TOPSUM / TOTSUM) * 100;
IF TOPPCT GE &PERCENT
THEN DO;
  OUTPUT;
  CALL SYMPUT('TOPPCT', trim(left(put(TOPPCT, 5.1))));
  CALL SYMPUT('TOPSUM', trim(left(put(TOPSUM, &RVARFMT))));
  CALL SYMPUT('TOPNN', trim(left(&_N_)));
  STOP;
END;
%end;
%else
%if %length(&NN) ne 0
%then %do;
IF &_N_ LE &NN
THEN DO;
  TOPSUM = TOPSUM + &RANKVAR;
END;
ELSE DO;
  TOPPCT = (TOPSUM / TOTSUM) * 100;
  CALL SYMPUT('TOPPCT', trim(left(put(TOPPCT, 5.1))));
  CALL SYMPUT('TOPSUM', trim(left(put(TOPSUM, &RVARFMT))));
  CALL SYMPUT('TOPNN', trim(left(&_N_ - 1)));
  STOP;
END;
%end;
OUTPUT;
IF LAST;
TOPPCT = (TOPSUM / TOTSUM) * 100;
CALL SYMPUT('TOPPCT', trim(left(put(TOPPCT, 5.1))));

```

```

CALL SYMPUT('TOPSUM',trim(left(put(TOPSUM,&RVARFMT))));
CALL SYMPUT('TOPNN',trim(left(_N_)));
RUN;

DATA _NULL_;
CALL SYMPUT('RANKLEN',trim(left(length(trim(left(&TOPNN))))));
%if %eval(&TOPNN < &TOTNN)
%then %do;
  %if %length(&NN) ne 0
  %then %do;
CALL SYMPUT('TITLEPREFIX',"Top &TOPNN &RVARLABL By &CVARLABL");
  %end;
  %else
  %if %length(&MINVALUE) ne 0
  %then %do;
CALL SYMPUT('TITLEPREFIX',
  "Ranked &RVARLABL not less than "||
  trim(left(put(&minvalue,&RVARFMT)))||
  " By &CVARLABL");
  %end;
  %else
  %if %length(&PERCENT) ne 0
  %then %do;
CALL SYMPUT('TITLEPREFIX',"Top &TOPNN &RVARLABL By &CVARLABL");
  %end;
%else %do;
CALL SYMPUT('TITLEPREFIX',"Ranked &RVARLABL By &CVARLABL");
%end;
RUN;

%if %upcase(&DateAndPageNumber) eq NO %then %do;
OPTIONS NODATE NONUMBER;
%end;
%if %length(&TITLESUFFIX) eq 0
%then %let TITLE1 = &TITLEPREFIX;
%else %let TITLE1 = &TITLEPREFIX &TITLESUFFIX;

TITLE1 "&TITLE1";
%let LastTitleNumber = 5;
%if %length(&PERCENT) ne 0
and
  %eval(&TOPNN < &TOTNN)
%then %do;
TITLE3
"Selected to list the Top &PERCENT.% of Total &RVARLABL";
  %if %sysevalf(&TOPPCT > &PERCENT)
  %then %do;
TITLE4 "(This subset actually accounts for &TOPPCT.%)";
  %let LastTitleNumber = %eval(&LastTitleNumber + 1);
  %end;
%end;
%else
%if %eval(&TOPNN < &TOTNN)
%then %do;
TITLE3 "This subset accounts for &TOPPCT.% of Total &RVARLABL";
%end;
%if %eval(&TOPNN < &TOTNN)
%then %do;
TITLE&LastTitleNumber
"Grand Totals: &CVARLABL Count = &TOTNN, &RVARLABL = &TOTSUM";
%end;
  /* to force line breaks in your labels, use _ */
PROC PRINT DATA=SUBSETTED NOOBS U LABEL SPLIT='_';
FORMAT RANK      &RANKLEN.;
FORMAT &RANKVAR &RVARFMT;
LABEL RANK      = 'Rank'
      &CLASSVAR = "&CVARLABL"
      &RANKVAR  = "&RVARLABL";
VAR RANK &CLASSVAR &RANKVAR;
SUM &RANKVAR;
RUN;

%MEND SubsettedRankingReport;

```

```
proc summary nway data=sashelp.shoes;
class subsidiary;
var sales;
output out=work.ShoeSalesByCity(rename=(subsidiary=City)) sum=Sales;
run;
%SubsettedRankingReport(DATA=work.ShoeSalesByCity,
    CLASSVAR=City,
    CVARLABL=City,
    RANKVAR=Sales,
    RVARFMT=dollar11.,
    RVARLABL=Shoe Sales,
    NN=10);

%SubsettedRankingReport(DATA=work.ShoeSalesByCity,
    CLASSVAR=City,
    CVARLABL=City,
    RANKVAR=Sales,
    RVARFMT=dollar11.,
    RVARLABL=Shoe Sales,
    PERCENT=80);

%SubsettedRankingReport(DATA=work.ShoeSalesByCity,
    CLASSVAR=City,
    CVARLABL=City,
    RANKVAR=Sales,
    RVARFMT=dollar11.,
    RVARLABL=Shoe Sales,
    MINVALUE=1000000);
```

Create a Transposed Listing of Your Table Content

Typical listings from PROC SQL, PROC PRINT, PROC REPORT or PROC TABULATE display data in a table with variable names, or labels, in the top row with values of each variable stacked in a column below its heading. If you have numerous variables, this standard convention can produce an excessively wide listing. Even for just a few observations, sideways scrolling is an on-line viewing nuisance, and the hard copy result of printing is unacceptable. It can be more usable for the on-line viewer or the hard-copy recipient to work with a transposed listing. On the next three pages are examples of transposed prints. A listing of the TransposedPrint macro and the macro invocations used to produce the examples follow. For the first publication of this work, see Reference 5.

One Observation in HTML Format (The data set, a copy of SASHELP.CLASS, has variable labels.)

Listing of WORK.LABELSCLASS
for Name EQ 'Carol'

Column Label	Value
Student	Carol
Gender	F
Age in Years	14
Hgt in Inches	62.8
Wgt in Pounds	102.5

Four Observations in RTF Format (The data set has no variable labels. The header for the values in each row is customized with the word "Student".)

Transposed Listing of SASHELP.CLASS
for Age EQ 15

Column	Student 1	Student 2	Student 3	Student 4
Name	Janet	Mary	Ronald	William
Sex	F	F	M	M
Age	15	15	15	15
Height	62.5	66.5	67	66.5
Weight	112.5	112	133	112

Full Listing in PDF Format (The bottom of the page was trimmed off for publication. Note the repeated column names when the listing wraps.)

Transposed Listing of SASHELP.CLASS

Column	Value Row 1	Value Row 2	Value Row 3	Value Row 4	Value Row 5	Value Row 6	Value Row 7
Name	Alfred	Alice	Barbara	Carol	Henry	James	Jane
Sex	M	F	F	F	M	M	F
Age	14	13	13	14	14	12	12
Height	69	56.5	65.3	62.8	63.5	57.3	59.8
Weight	112.5	84	98	102.5	102.5	83	84.5

Column	Value Row 8	Value Row 9	Value Row 10	Value Row 11	Value Row 12	Value Row 13	Value Row 14
Name	Janet	Jeffrey	John	Joyce	Judy	Louise	Mary
Sex	F	M	M	F	F	F	F
Age	15	13	12	11	14	12	15
Height	62.5	62.5	59	51.3	64.3	56.3	66.5
Weight	112.5	84	99.5	50.5	90	77	112

Here is the code used to produce the reports:

```
%MACRO TransposedPrint(
Data=
,Where=
,PreTitle=
,UseLabelsIfAvailable=YES
,ValueHeadingPrefix=Value_Row
,FontFace=Verdana
,FontSize=16pt
,VarNameHeaderTextColor=CXFF0000
,VarNameHeaderBackgroundColor=CX0000FF
,VarNameTextColor=CX0000FF
,VarNameBackgroundColor=CXFF0000
,ValueHeaderTextColor=CXFFFF00
,ValueHeaderBackgroundColor=CX006600
,ValueTextColor=CX006600
,ValueBackgroundColor=CXFFFF00
,UseCustomColors=YES);

/* Note: In this macro, the use of
   data set variable labels, Where=, and PreTitle= are optional. */

%LET FontFormatting = %STR(FONT_FACE=&FontFace FONT_SIZE=&FontSize);
%LET UseCustomColors = %UPCASE(&UseCustomColors);
%LET Library = %SUBSTR(&Data,1,%INDEX(&Data,.) - 1);
%LET DataSet =
  %SUBSTR(&Data,
    %EVAL(%INDEX(&Data,.) + 1),
    %EVAL(%LENGTH(&Data) - %INDEX(&Data,.) ));
%LET Library = %UPCASE(&Library);
%LET DataSet = %UPCASE(&DataSet);
```

```

PROC SQL NOPRINT;
SELECT COUNT(*) INTO :N
  FROM DICTIONARY.COLUMNS
  WHERE LIBNAME EQ "&Library" AND MEMNAME EQ "&DataSet";
SELECT NAME INTO :COLUMN1 - :COLUMN%TRIM(%LEFT(&N))
  FROM DICTIONARY.COLUMNS
  WHERE LIBNAME EQ "&Library" AND MEMNAME EQ "&DataSet";
QUIT;

%MACRO VarList;
%DO I = 1 %TO &N %BY 1;
&&COLUMN&I
%END;
%MEND VarList;

DATA WORK.ToTranspose;
SET &Library..&DataSet;
%IF %LENGTH(&Where) NE 0 %THEN %DO;
WHERE &Where;
%END;
RUN;

DATA _NULL_;
SET WORK.ToTranspose NOBS=HowMany;
CALL SYMPUT('ValueCount',HowMany);
STOP;
RUN;

PROC TRANSPOSE DATA=WORK.ToTranspose
OUT=WORK.ToPrint;
VAR %VarList;
RUN;

%MACRO Labels;
%DO I = 1 %TO &ValueCount %BY 1;
LABEL COL&I = "&ValueHeadingPrefix &I";
%END;
%MEND Labels;

%MACRO Cols;
%DO I = 1 %TO &ValueCount %BY 1;
COL&I
%END;
%MEND Cols;

%LET LabelVarFound = NO;
%IF %UPCASE(&UseLabelsIfAvailable) EQ YES
%THEN %DO;
PROC SQL NOPRINT;
SELECT NAME INTO :LabelVarFound
  FROM DICTIONARY.COLUMNS
  WHERE LIBNAME EQ "WORK" AND MEMNAME EQ "TOPRINT" AND NAME EQ "_LABEL_";
QUIT;
%LET LabelVarFound = %TRIM(%LEFT(&LabelVarFound));
%END;
OPTIONS NOCENTER LINESIZE=MAX PAGESIZE=MAX;
TITLE1 FONT="&FontFace" HEIGHT=&FontSize
%IF %LENGTH(&PreTitle) NE 0 %THEN %DO;
  "&PreTitle"
  JUSTIFY=LEFT
%END;
%IF &ValueCount GT 1 %THEN %DO;
  "Transposed "
%END;
  "Listing of &Library..&DataSet"
%IF %LENGTH(&Where) NE 0 %THEN %DO;
  JUSTIFY=LEFT
  "for &Where"
%END;
;
PROC PRINT DATA=WORK.ToPrint LABEL SPLIT='_';
%IF &LabelVarFound EQ _LABEL_ %THEN %DO;
  ID _LABEL_ /
%END;

```

```

%ELSE %DO;
  ID _NAME_ /
%END;
  STYLE(HEADER) = [&FontFormatting
%IF &UseCustomColors EQ YES %THEN %DO;
  BACKGROUND=&VarNameHeaderBackgroundColor
  FOREGROUND=&VarNameHeaderTextColor
%END;
  ]
  STYLE(DATA) = [&FontFormatting
%IF &UseCustomColors EQ YES %THEN %DO;
  BACKGROUND=&VarNameBackgroundColor
  FOREGROUND=&VarNameTextColor
%END;
  ];
%IF &LabelVarFound EQ _LABEL_ %THEN %DO;
  LABEL _LABEL_ = 'Column_Label';
%END;
%ELSE %DO;
  LABEL _NAME_ = 'Column';
%END;
  VAR %Cols /
  STYLE(HEADER) = [&FontFormatting
%IF &UseCustomColors EQ YES %THEN %DO;
  BACKGROUND=&ValueHeaderBackgroundColor
  FOREGROUND=&ValueHeaderTextColor
%END;
  ]
  STYLE(DATA) = [&FontFormatting
%IF &UseCustomColors EQ YES %THEN %DO;
  BACKGROUND=&ValueBackgroundColor
  FOREGROUND=&ValueTextColor
%END;
  ];
%IF &ValueCount EQ 1 %THEN %DO;
  LABEL COL1 = 'Value';
%END;
%ELSE %DO;
  %Labels;
%END;
RUN;
%MEND TransposedPrint;

DATA work.LabelsCLASS;
SET sashelp.class;
LABEL Name='Student'
      Sex='Gender'
      Weight='Wgt in Pounds'
      Height='Hgt in Inches'
      Age='Age in Years';
RUN;

ODS LISTING CLOSE;
ODS NORESULTS; * Do not open output in SAS *;

GOPTIONS RESET=ALL; * Always do this. *;
ODS HTML PATH="C:\\" (URL=NONE)
  BODY="OneLabeledObs.html" (TITLE="One Labeled Observation");
%TransposedPrint(Data=work.LabelsCLASS,Where=Name EQ 'Carol');
ODS HTML CLOSE;

OPTIONS NODATE NONUMBER;

GOPTIONS RESET=ALL; * Always do this. *;
ODS RTF FILE="C:\FourObs.rtf";
%TransposedPrint(Data=sashelp.class,Where=Age EQ 15,ValueHeadingPrefix=Student);
ODS RTF CLOSE;

GOPTIONS RESET=ALL; * Always do this. *;
ODS PDF NOTOC FILE="C:\AllObs.PDF";
%TransposedPrint(Data=sashelp.class,FontFace=Helvetica);
ODS PDF CLOSE;

ODS LISTING;

```

Conclusion

I hope that you find some use for the tools provided above. They were fun (and some were easy) to develop. Their use offers a big potential return on the small development investment.

References

1. LeRoy Bessler, “More Ways to Use SAS to Manage, Monitor, and Control SAS or the SAS BI Server: Tools for the SAS User, Server Administrator, or Manager”, *Proceedings of SAS Global Forum 2010 Conference*, SAS Institute, Inc. (USA), 2010. Find this paper on the web at:
<http://support.sas.com/resources/papers/proceedings10/279-2010.pdf>
2. Erik W. Tilanus, “Sending E-mail from the DATA step”, *Proceedings of SAS Global Forum 2008 Conference*, SAS Institute, Inc. (USA), 2008. Find this paper on the web at:
www2.sas.com/proceedings/forum2008/038-2008.pdf
3. LeRoy Bessler, “SAS Graphs for a BlackBerry, iPhone, or Other Small Email Screen: Extreme SAS/GRAPH and the Necessity and Power of Simplicity”, *Proceedings of SAS Global Forum 2008 Conference*, SAS Institute, Inc. (USA), 2008. Find this paper on the web at: www2.sas.com/proceedings/forum2008/038-2008.pdf
4. LeRoy Bessler, “Show Them What’s Important with Communication-Effective SAS Graphs: Solutions for a Finite Work Day in an Era of Information Overload”, *Proceedings of SAS Global Forum 2009 Conference*, SAS Institute, Inc. (USA), 2009. Find this paper on the web at:
<http://support.sas.com/resources/papers/proceedings09/220-2009.pdf>
5. LeRoy Bessler, “A Transposed Listing of Your Table Content”, *IEWS News*, Issue 41, *IEWS International SAS Programmer Community (UK)*, 2008. Find this paper on the web at:
http://www.sascommunity.org/mwiki/images/5/5d/IEWS_News_Issue41.pdf

Acknowledgement

Thanks to Alexandra Riley for her helpful review of this paper.

Author Information

Your questions, comments, suggestions, and examples of other concise tabular or graphic SAS reporting methods and tools are always welcome.

LeRoy Bessler PhD

Bessler Consulting and Research, Mequon, Milwaukee, Wisconsin, USA

Le_Roy_Bessler@wi.rr.com

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 interests include communication-effective visual communication and reporting, web information delivery, highly formatted Excel reporting, SAS/GRAPH, ODS, creation of unique tools to support the SAS BI server and its users, and Software-Intelligent Application Development for Reliability, Reusability, Extendibility, and Maintainability. He is a regular contributor to *IEWS News*, the web newsletter of the *IEWS International SAS Programmer Community*, and has been an organizer of SAS users conferences since 1989.

SAS and SAS/GRAPH are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

® indicates USA registration. Other product and brand names are trademarks or registered trademarks of their respective owners.