

Paper 034-2008

## SAS® Graphs for a BlackBerry, iPhone, or Other Small Email Screen: Extreme SAS/GRAPH® and the Necessity and Power of Simplicity

LeRoy Bessler, Assurant Health, Milwaukee, Wisconsin, USA

### Abstract

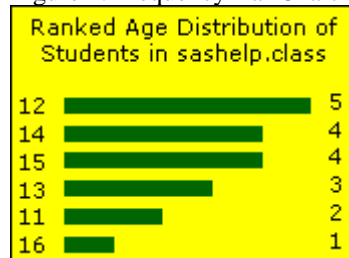
Delivery of a readable graph to a BlackBerry, an iPhone, or any small screen demands a simple design. For over twenty years, I have been advocating for minimalist graphic design to focus on the message, not the media. I have demonstrated the power of simplicity by stripping out traditional graphic presentation paraphernalia while making communication-effective use of SAS and SAS/GRAPH. This paper shows you how to provide not only visual indication of size or trend for quick and easy inference, but also precise numbers for reliable inference. You can manually attach the graphs to an email message, or you can make SAS programmatically email the graphs for you.

### Introduction

You might already know how to send email from SAS. The challenge addressed here is how to format a readable graph for a BlackBerry, an iPhone, or other very small screen communication device. The minimalist image in Figure 1 is one example, published in the August 2007 issue of *SAS Tech Report* and of *SAS Training Report*, both available at <http://support.sas.com>. The example in Figure 4 was published in *VIEWS News: News, Hints, Tips, and Information for SAS Users*, Issue 39, 3rd Quarter 2007, available at <http://www.views-uk.org>.

In this paper you will find communication-effective design for the small screen with image and sufficient detail, and SAS/GRAPH code needed, for all of the popular management reporting formats. Here is a simple bar chart:

Figure 1. Frequency Bar Chart



Here is the code used to create and send the image above:

```
goptions reset=all;
goptions device=GIF;
goptions ypixels=129 xpixels=172; /* Default size for a GIF image is 800 X 600.
  Preserve 4-to-3 aspect ratio, but adjust the size for a BlackBerry or iPhone. */
goptions border; /* Put the graph in a box. */
goptions cback=CXFFFF00; /* Use RGB Yellow for background. */
goptions htext=10pt ftext='Verdana';
  /* Specify height and font used for those parts of the graph
  for which you do not make an explicit assignment,
  or for which no direct controls are available in SAS/GRAPH. */
goptions gsfmode=replace gsfname=anyname;
filename anyname "C:\FolderName\File Name.GIF";

title1 font='Verdana'
  color=CX000000 /* RGB black */
  height=3 PCT ' ' /* Insert space at the top. */
  height=10pt
  justify=CENTER /* Force a new line. */
  "Ranked Age Distribution of"
  justify=CENTER /* Force a new line. */
  "Students in sashelp.class";
```

```

footnote1 angle=+90 font=none height=1 ' ';
/* This "footnote" is a blank "right-side-note"
to increase space at the side of the image. */
/* Use angle=-90 to create a "left-side-note" if needed. */

pattern1 v=solid color= CX006600; /* Use an RGB Medium Dark Green for the bars. */

/* Remove all axis paraphernalia: */
axis1 label=none major=none minor=none style=0;
axis2 label=none major=none minor=none style=0 value=none
offset=(5 PCT,5 PCT);
/* Offset the horizontal axis to insert extra space
between bar midpoint label and bar start,
and between bar end and bar response value. */

proc gchart data=sashelp.class;
hbar Age /
discrete /* Use Ages, not Age subranges. */
freq
freqlabel=' ' /* Suppress the Frequency column heading. */
descending
maxis=axis1 raxis=axis2
width=2.5 space=2.5; /* bar width and spacing */
run; quit;

filename anyname2 email
to="bessler@execpc.com"
subject="Graph for Your BlackBerry"
attach="C:\FolderName\File Name.GIF";

data _null_;
file anyname2;
put 'Please see the attached graph.';
run;

```

Your SAS program can send email to multiple addresses, with multiple attachments, with a CC and/or BCC, etc. For more information about SAS email in the SAS 9.1.3 OnlineDoc, go to:  
<http://support.sas.com/onlinedoc/913/getDoc/en/irdict.hlp/a002058232.htm>

**Note:** To get a similar looking result with a different release of SAS/GRAPH, it might be necessary to change one or more of the following: bar width, bar space, inserted space at the top of the graph, or inserted space at the right-hand side of the graph.

In the remainder of this paper I present groups of related figures and comments, but the associated SAS code is in the Appendix. For some figures, the code is not provided. Omitted code can be requested from the author. Code to send email attachments from SAS is presented only in the Introduction above.

I want to display the first gallery of graphs all on one page. So, I have some extra space here. Let me fill it with a macro that is used to prepare for the creation of each graph and that is referenced in the Appendix. Explanatory comments for some of its internal statements were given earlier in the Introduction.

```

%macro CommonPreliminaryCode(PathToGraph=,FigureNumber=,BackgroundColor=CXFFFF00);
goptions reset=all;
goptions device=GIF;
goptions ypixels=129 xpixels=172;
goptions border;
goptions cback=&BackgroundColor;
goptions gsfmode=replace gsfname=anyname;
filename anyname "&PathToGraph.\Figure&FigureNumber..GIF";
%mend CommonPreliminaryCode;

```

**Bar Charts and Pie Charts for the Small Screen**

Figure 2. Sum Bar Chart

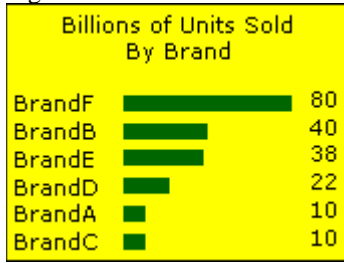
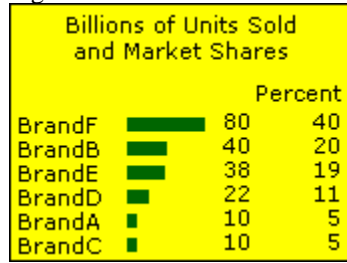


Figure 3. Pie Chart Alternative



To maximize the number of bars that can be displayed, remove the space between them and append the column of bar values to the bar names as shown in Figures 4 and 5.

Figure 4. Top Ten Chart

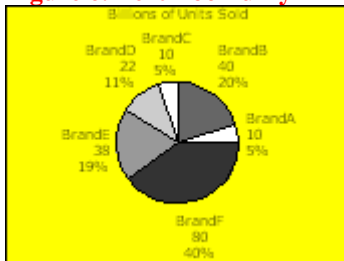


Figure 5. Maximum Capacity



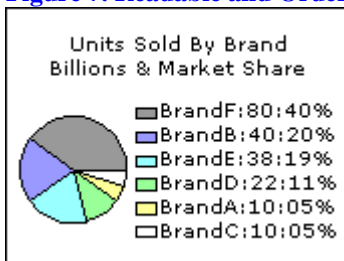
With 13 bars, the rows of text start to touch each other. The next smaller font is readable, but fuzzy as in Figure 6.

Figure 6. Text Too Fuzzy



The text in Figure 6 is too fuzzy, but the next larger font, used in Figures 4 and 5, causes pie slice label overlap. I have long recommended ordering bars and pie slices by size to “Show the Viewer What Is Important.” Even with the small fuzzy font, descending slice order would force slice label overlap. The custom legend in Figure 7 is the only very readable pie chart solution and can always support descending or ascending slice size. Since Figure 7 uses a color-coded legend, I prefer to use a white background. Amazingly, the fuzzy text in Figure 6 is harder to read on a white background than on a yellow background.

Figure 7. Readable and Ordered



## Trend Charts for the Small Screen

For at least fifteen years, I have been making the case that a sparse trend chart is a sufficient trend chart. It enables easy and quick visual inferences. Precise numbers are best provided in a companion table. However, there are always four precise values of interest to a typical viewer of a trend chart: start, end, minimum and maximum. There can be other points of interest on a trend chart, but those points are not amenable to automated programmatic detection and can be leveraged only by iterative ad hoc graph creation. I prefer a robust solution for hands-off, fully automated graph production applications. For iterative development of a graph for an ad hoc request, a robust solution reduces your turnaround time to meet the needs of the requestor.

Figure 8. Trend by Trading Day

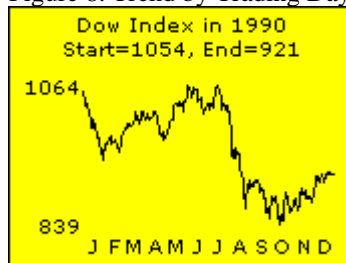


Figure 9. Different Use of Display Space

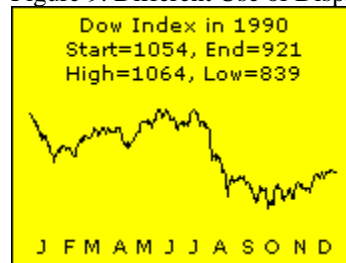


Figure 10. Reference Lines

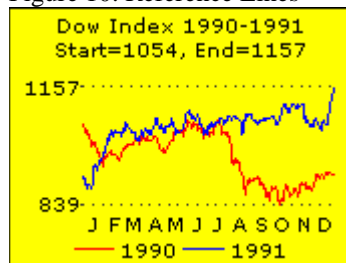


Figure 11. Best In Category

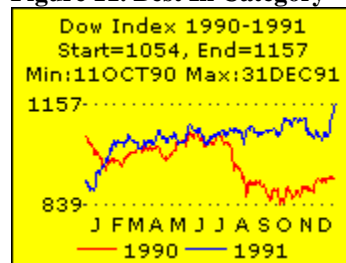
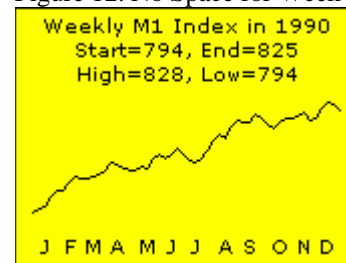


Figure 12. No Space for Week Numbers



In all of these trend charts, there is not enough space on the horizontal axis to present many tick mark labels. The irregular spacing of the month name initials above is due to the programming algorithm used to locate them. For Figures 8-11, since there are an average of 22 trading days in each month, trading day 11 is used. Since the actual number of trading days varies by month, so does the resulting spacing of labels. For Figure 12, where there are a variable number of weekly values in each month, week 3 is used, with irregular consequences for label spacing.

Figure 13. Vertical Tick Values

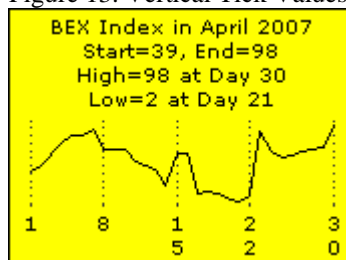
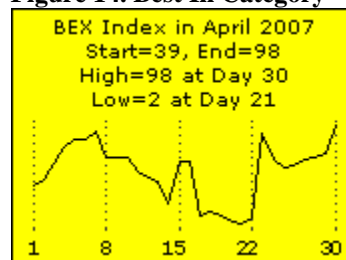


Figure 14. Best In Category



The stacked digits in Figure 13 are SAS/GRAPH default action for this data. I do not recommend vertical or tilted horizontal axis values—hence my preference for Figure 14. English is read left-to-right, not vertically or tilted. The solution for daily tick marks in a month always displays only 1, 8, 15, 22, and the last day of the month. The trick used for Figure 14 actually lays down the first digit of each two-digit day at the location of the day prior—to outwit SAS/GRAPH. The reference lines are used to overcome ambiguity. For a month of days, there are few enough points to be able to display plot symbols/markers, but I did not do so. For denser trends, they merge into a thick line.

If the trend chart design solutions above do not provide sufficient information, additional precise numbers can be provided in a companion text message or a companion table. The usability of a large table on the small screen is a challenge—scrolling is a sub-optimal user experience even on the large screen.

## Fonts and Font Sizing for the Small Screen

I evaluated all of the SAS/GRAPH “Software Fonts”, i.e., the fonts that are supplied with SAS/GRAPH and therefore work on any computing platform. I also evaluated all of the common Windows fonts, which I prefer. The SAS/GRAPH software fonts did not look very good, but if you do not have access to Windows fonts on your computer, the SWISS sans serif font and ZAPF and CENTX serif fonts might be acceptable, even if sub-optimal.

My favorite Windows sans serif font is Verdana. (This font was developed for Microsoft specifically for readability in on-screen display and in web pages.) I recommend its use on the small screen and anywhere else. Arial, undoubtedly the most commonly used Windows sans serif font, is acceptable, and offers the benefit of being narrower. This allows you to squeeze more text into the narrow width of the small screen, as seen by comparing Figures 15 and 16. For both of these fonts, the smallest practical point size is 9 pt. For the small text parts of graphs, sans serif fonts are more readable. The samples below are not a full display of the upper and lower case alphabets.

Figure 15. Verdana Point Sizes

```
Verdana Point Sizes:
h=8pt ABC abc 1234567890
h=9pt ABC abc 1234567890
h=10pt ABC abc 1234567890
h=12pt ABC abc 1234567
h=13pt ABC abc 12345
h=14pt ABC abc 123
h=16pt ABC abc 12
```

Figure 16. Arial Point Sizes

```
Arial Point Sizes:
h=8pt ABC abc 1234567890
h=9pt ABC abc 1234567890
h=10pt ABC abc 1234567890
h=12pt ABC abc 1234567
h=13pt ABC abc 12345
h=14pt ABC abc 123
h=16pt ABC abc 12
```

Figure 17. Sans Serif vs. Serif

```
Common Windows Fonts
All h=9pt Samples:
Sans Serif Fonts
Verdana ABC abc 1234567890
Arial ABC abc 1234567890
Serif Fonts
Georgia ABC abc 1234567890
Times New Roman ABC abc 1234567890
```

## Plot Symbols/Markers for the Small Screen

I evaluated all of the possibilities in the SAS/GRAPH symbol-oriented fonts. I also evaluated symbols that are available by specifying a string of two hexadecimal characters, as in VALUE='HH'X, on the SYMBOL statement when using the SAS/GRAPH text-oriented fonts. I further evaluated symbols available in a wide variety of Windows fonts—not only symbols available, by using VALUE='HH'X, in the extensions to fonts that are primarily Windows text fonts, but also symbols that are provided in Windows fonts of symbols, icons, and widgets—fonts such as Webdings, Wingdings, Monotype Sorts, etc. After all of this experimentation, I identified only two acceptable symbol shapes for the small screen. The color of tiny plot symbols is too hard to distinguish. Therefore, on the small screen it is necessary to use line color in order to distinguish several trend lines in the same graph.

On the small screen, it is infeasible, at least with SAS/GRAPH, to get nicely formed round plot symbols of significant size by any means. VALUE=POINT appears to produce round points, but they are very small and would not enlarge by increasing the HEIGHT (the size parameter for the SYMBOL statement). The larger VALUE=DOT produces varying amorphous shapes rather than round symbols of consistent shape. When trying to get squares, I usually got rectangles. I found no other shapes that were drawn well and consistently for all points in the graph. So, when you have few enough plot points so that they do not just merge into a thick line or curve, I recommend:

```
symboln color=black interpol=none height=1 font=Special value='K';
```

Figure 18: Point Symbol

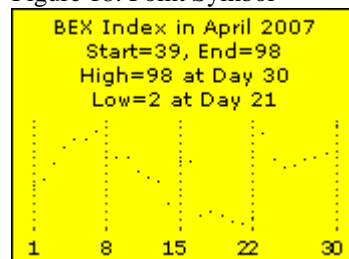


Figure 19: Symbol Special 'K'

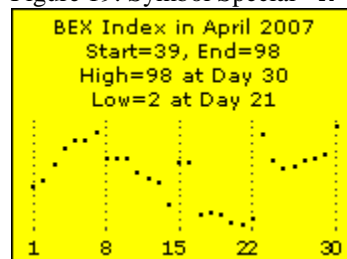
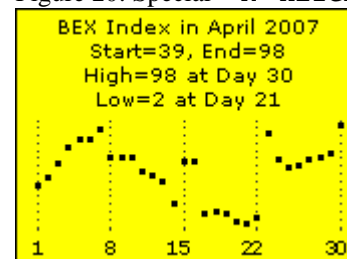


Figure 20: Special 'K' HEIGHT=2



Adding a plot line with a symbol present adds no value and has an unattractive effect. Enlarging the symbol does not improve the result with a line present. For plot symbols on the small screen, these are the only good choices.

## Conclusion

SAS/GRAPH can create readable images that work in the extreme conditions of the small screen. As has long been known, SAS can email graphs (and tables). Except for the problem demonstrated in Figure 6, the designs here are communication-effective for their respective purposes. Graphs facilitate and accelerate inference, but reliable inference requires detail numbers. All of the designs shown here provide both visual and precision. However, there is an obvious limitation on how much text and numbers you can present on the small screen.

## References

This recent related work by the author on Communication-Effective Graphic Design and Construction is all available in Online Proceedings at <http://support.sas.com/events/sasglobalforum/previous/online.html>

1. Get the Best out of SAS/GRAPH and ODS, *Proceedings of the SAS Global Forum 2007 Conference*. Cary, NC: SAS Institute Inc.
2. Communication-Effective Pie Charts, *Proceedings of the SAS Global Forum 2007 Conference*. Cary, NC: SAS Institute Inc.
3. How to Make the “Best Choice” from the Many Ways to Create and Deliver SAS Graphs, *Proceedings of the SAS Global Forum 2007 Conference*. Cary, NC: SAS Institute Inc.

## Acknowledgment

I am grateful to Alexandra Riley for her helpful review of this paper and for testing the output on an iPhone.

## Contact Information

Your comments, questions, and suggestions are welcome. I am always interested in design ideas or construction solutions that enhance graphic communication or address new information delivery needs.

LeRoy Bessler PhD  
Email: [Le\\_Roy\\_Bessler@wi.rr.com](mailto:Le_Roy_Bessler@wi.rr.com)

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.

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

**Appendix. Code to Create Graphs for the Small Screen.** NOTE: For some figures no code is presented.

**NOTE:** When creating graphs for the small screen, you might get messages in the SAS log saying that axis values have been overwritten. Only believe the warning if your inspection of the graph confirms it.

## Input Data Not in SASHELP Data Library

### Data for Figures 2, 3, 6, 7

```
data DataForSimpleCharts;
infile datalines; input @1 Name $6. @8 Value 2.;
datalines;
BrandA 10
BrandB 40
BrandC 10
BrandD 22
BrandE 38
BrandF 80
; run;
```

Data for Figures 4 & 5

```

data CountriesbySizeTop12
  CountriesbySizeTop10;
length NameAndValue $ 19;
infile datalines;
input @1 Name $13. @15 Value comma6.;
NameAndValue = trim(left(Name)) || ':' || trim(left(put(Value,5.)));
if _N_ LE 12 then output CountriesbySizeTop12;
if _N_ LE 10 then output CountriesbySizeTop10;
datalines; /* extra data present below */
Russia      17,075
Canada      9,971
China       9,561
Brazil      8,512
Australia   7,682
India       3,287
United States 9,373
Argentina   2,767
Kazakhstan  2,717
Sudan       2,506
Algeria     2,382
Congo       2,345
Saudi Arabia 2,200
Greenland   2,176
Mexico      1,973
;
run;

```

Data for Figures 8-11

```

data work.DOW19901991;
drop snydjcm;
retain FirstTradingDayIn1991Found 'N' TradingDay 0;
length MonthAbbrev $ 1;
set sashelp.citiday
  (keep=date snydjcm
   where=(snydjcm ne .));
Year = year(date);
if Year in (1990 1991);
if Year eq 1990
  or
  FirstTradingDayIn1991Found eq 'Y'
then TradingDay = TradingDay + 1;
else do;
  TradingDay = 1;
  FirstTradingDayIn1991Found = 'Y';
end;
Dow=round(snydjcm,1);
Month = month(date);
MonthAbbrev=substr(left(put(date,monname3.)),1,1);
run;

```

Data for Figures 13 & 14

```

data BEX2007April;
infile datalines;
input @1 Day 2. @4 BEX 2.;
datalines;
1 39 200704
2 46 200704
3 61 200704
4 76 200704

```

```

5 84 200704
6 84 200704
7 93 200704
8 67 200704
9 67 200704
10 67 200704
11 53 200704
12 48 200704
13 43 200704
14 21 200704
15 63 200704
16 63 200704
17 11 200704
18 13 200704
19 11 200704
20 4 200704
21 2 200704
22 8 200704
23 90 200704
24 65 200704
25 57 200704
26 59 200704
27 65 200704
28 67 200704
29 70 200704
30 98 200704
;
run;

```

## Macros

### Macros for Figure 7

```

%macro PatternStatements;
%do i = 1 %to 6;
pattern&i v=psolid color=&&SliceColor&i repeat=1;
%end;
%mend PatternStatements;

```

```

%macro LegendEntries;
%do i = 1 %to 6;
  "&&LegendEntry&i"
%end;
%mend LegendEntries;

```

### Macros for All Trend Charts

```

%macro GetHtickmarkValues;

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

%mend GetHtickmarkValues;

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

%global yVarStart yVarEnd;

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

```



```

%end;
run;

data _null_;
set Selected 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=Selected 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 for Trend Chart Figures 8-11

```

%macro CreateTradeDayHtickmarkValues(data=,filter=);

* Maximum number of trading days in a year is 263. *;
* Possible extra GLOBAL statements are harmless. *;
%do i = 1 %to 262 %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 11
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 CreateTradeDayHtickmarkValues;

```

### Macro for Trend Chart Figure 11

```

%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;

```

### Macros for Trend Chart Figures 13 & 14

```

%macro IdentifyXvalsForCriticalPoints(data=,yVar=);

```

```

%global AtXvalueWhenMinYisUnique AtXvalueWhenMaxYisUnique;

data _null_;
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 = Day;
  end;
  else HowManyXfoundForMinY + 1;
end;
else
if &Yvar EQ &yAxisEndTickMarkValue
then do;
  if HowManyXfoundForMaxY eq 0
  then do;
    HowManyXfoundForMaxY = 1;
    xForMaxY = Day;
  end;
  else HowManyXfoundForMaxY + 1;
end;
if LastOne;
put HowManyXfoundForMinY= HowManyXfoundForMaxY= xForMinY= xForMaxY=;
if HowManyXfoundForMinY eq 1
then call symput('AtXvalueWhenMinYisUnique',
" at Day " || trim(left(xForMinY)));
if HowManyXfoundForMaxY eq 1
then call symput('AtXvalueWhenMaxYisUnique',
" at Day " || trim(left(xForMaxY)));
run;

%mend IdentifyXvalsForCriticalPoints;

```

#### Macro for Trend Chart Figure 14

```

%macro BetterDayOfMonHtickmarkValues(data=);

%global NumberOfTickMarks;

data _null_;
set &data nobs=HowMany;
call symput('NumberOfTickMarks',trim(left(HowMany)));
call symput('SecondLast',trim(left(HowMany - 1)));
stop;
run;

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

data _null_;
set &data;
if _N_ in (1 8)
then call symput('htickvalue' || trim(left(_N_)),trim(left(_N_)));
else
if _N_ eq 14
then call symput('htickvalue' || trim(left(_N_)), '1');
else

```

```

if _N_ eq 15
then call symput('htickvalue' || trim(left(_N_)), '5');
else
if _N_ in (21 22)
then call symput('htickvalue' || trim(left(_N_)), '2');
else
if _N_ eq %superq(SecondLast)
then call symput('htickvalue' || trim(left(_N_)),
                 trim(left(%substr(%superq(NumberOfTickMarks), 1, 1))));
else
if _N_ eq %superq(NumberOfTickMarks)
then call symput('htickvalue' || trim(left(_N_)),
                 trim(left(%substr(%superq(NumberOfTickMarks), 2, 1))));
else call symput('htickvalue' || trim(left(_N_)), '');
run;

%mend BetterDayOfMonHtickmarkValues;

```

## Code

See the Introduction for the source listing of the CommonPreliminaryCode macro used repeatedly in the remainder of the Appendix.

```
options mprint symbolgen;
```

```
%let ThePath = C:\Folder;
```

### Code for Figure 2

```

%CommonPreliminaryCode(PathToGraph=&ThePath, FigureNumber=2);

goptions htext=10pt ftext='Verdana';
    /* Specify height and font used for those parts of the graph
       for which you do not make an explicit assignment,
       or for which no direct controls are available in SAS/GRAPH. */

title font='Verdana'
      color=CX000000 /* RGB black */
      height=3 PCT ' ' /* Insert space at the top. */
      height=10pt
      justify=CENTER /* Force a new line. */
      "Billions of Units Sold"
      justify=CENTER /* Force a new line. */
      "By Brand";

footnote1 angle=+90 font=none height=1 ' ';
footnote2 angle=-90 font=none height=1 ' ';

pattern1 v=solid color= CX006600; /* Use an RGB Medium Dark Green for the bars. */

    /* Remove all axis paraphernalia: */
axis1 label=none major=none minor=none style=0;
axis2 label=none major=none minor=none style=0 value=none
      offset=(5 PCT, 5 PCT);
    /* Offset the horizontal axis to insert extra space
       between bar midpoint label and bar start,
       and between bar end and bar response value. */

proc gchart data=DataForSimpleCharts;
hbar Name /
      sumvar=Value
      sum
      sumlabel=' ' /* Suppress the Sum column heading. */

```

```

    descending
    maxis=axis1 raxis=axis2
    width=2.5 space=2.5; /* bar width and spacing */
run; quit;

```

### Code for Figure 3

```

%CommonPreliminaryCode(PathToGraph=&ThePath,FigureNumber=3);

goptions htext=10pt ftext='Verdana';
    /* Specify height and font used for those parts of the graph
       for which you do not make an explicit assignment,
       or for which no direct controls are available in SAS/GRAPH. */

title1 font='Verdana'
    color=CX000000 /* RGB black */
    height=3 PCT ' ' /* Insert space at the top. */
    height=10pt
    /* height=9pt smaller text for two-line title */
    justify=CENTER /* Force a new line. */
    "Billions of Units Sold"
    justify=CENTER /* Force a new line. */
    "and Market Shares";

footnote1 angle=+90 font=none height=1 ' ';
footnote2 angle=-90 font=none height=1 ' ';

pattern1 v=solid color= CX006600; /* Use an RGB Medium Dark Green for the bars. */

    /* Remove all axis paraphernalia: */
axis1 label=none major=none minor=none style=0 offset=(0,15 PCT);
    /* Without the OFFSET as a problem circumvention,
       the column labels would overlap the first row of values */
axis2 label=none major=none minor=none style=0 value=none
    offset=(5 PCT,5 PCT);
    /* Offset the horizontal axis to insert extra space
       between bar midpoint label and bar start,
       and between bar end and bar response value. */

proc gchart data=DataForSimpleCharts;
hbar Name /
    freq=Value
    freq
    freqlabel=' ' /* Suppress the Freq column heading. */
    percent
    percentlabel='Percent'
    descending
    maxis=axis1 raxis=axis2
    width=2 space=2.5; /* slightly reduce bar width */
run; quit;

```

### Code for Figure 5

```

%CommonPreliminaryCode(PathToGraph=&ThePath,FigureNumber=5);

    /* increase space at left-hand side */
footnote1 angle=-90 font=none height=1 ' ';

title1 font='Verdana' height=10pt "Top 12 Sizes in '000 sq km";

goptions htext=9pt ftext='Verdana';

    /* Remove all axis paraphernalia: */

```

```

axis1 label=none major=none minor=none style=0;
axis2 label=none major=none minor=none style=0 value=none;

pattern1 v=empty color=CX006600; /* RGB Medium Dark Green bars, outline only */

proc gchart data=CountriesbySizeTop12;
hbar NameAndValue /
  sumvar=Value
  descending
  maxis=axis1 raxis=axis2
  nostats /* no values at right margin */
  space=0; /* default width for bars, no space between them */
run; quit;

```

### Code for Figure 7

```

/** Prepare to Create the Custom Pie Chart */

proc means data=DataForSimpleCharts noprint sum;
var Value;
output out=PieTotal sum=TotalValue;
run;

/* concatenate Percent and Value with the original Slice Name */
data SliceNameWithPercentAndValue;
length NameWithPercentAndValue $ 15;
set DataForSimpleCharts;
if _N_ eq 1 then set PieTotal;
Percent = (Value / TotalValue) * 100;
NameWithPercentAndValue =
  trim(left(Name)) || ':' || trim(left(put(Value,2.))) || ':' ||
  trim(left(put(Percent,Z2.))) || '%';
run;

proc sort data=SliceNameWithPercentAndValue;
by descending Percent; /* sequence the slices from largest to smallest */
run;

data SliceNameWithPercentAndValue;
set SliceNameWithPercentAndValue;
pctseq = _N_; /* add a key to merge this data with the ColorList */
call symput('LegendEntry' || trim(left(_N_)), trim(left(NameWithPercentAndValue)));
/* Store the legend text entries in the symbol table. */
run;

data ColorList; /* Sequence the colors from darkest to lightest,
  if you want small slices to stand out. */
pctseq = 1; color='CX999999'; output;
pctseq = 2; color='CX9999FF'; output;
pctseq = 3; color='CX99FFFF'; output;
pctseq = 4; color='CX99FF99'; output;
pctseq = 5; color='CXFFFF99'; output;
pctseq = 6; color='CXFFFFFF'; output;
run;

data SliceWithColor;
merge SliceNameWithPercentAndValue ColorList;
by pctseq;
run;

proc sort data=SliceWithColor;
by NameWithPercentAndValue; /* because SAS/GRAPH will always apply
  your PATTERN statements in the sort order of the SLICE name text */

```

```

run;

data _null_;
set SliceWithColor;
call symput('SliceColor' || trim(left(_N_)), trim(left(color)));
      /* Store the slice colors in the symbol table. */
run;

  /*** Create the Custom Pie Chart ***/

%CommonPreliminaryCode(PathToGraph=&ThePath, FigureNumber=7,
  BackgroundColor=CXFFFFFF); /* RGB White background because color legend for pie. */

footnote1 angle=0 font=none height=1 ' '; /* white space at bottom */
footnote2 angle=-90 font=none height=1 ' '; /* push pie to the right */

title1 font='Verdana'
  height=9pt ' '; /* Insert space at the top. */
title2 font='Verdana'
  color=CX000000 /* RGB black */
  height=9pt
  'Units Sold By Brand'
  justify=CENTER /* Force a new line. */
  'Billions & Market Share';

goptions htext=9pt ftext='Verdana';

legend1 order=(%LegendEntries) label=none shape=bar(6 PCT,4 PCT)
  across=1 position=(middle right outside) offset=(-2 PCT,0);

%PatternStatements;

proc gchart data=SliceNameWithPercentAndValue;
pie NameWithPercentAndValue / sumvar=Value
  noheading coutline=CX000000 descending
  legend=legend1
  slice=none value=none percent=none; /* turn off all pie labels */
run; quit;

```

### Code for Figure 11

```

%CommonPreliminaryCode(PathToGraph=&ThePath, FigureNumber=11);

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. */
title1 font=none height=1 ' ';

goptions htext=9pt ftext='Verdana';

* 1990 and 1991 had 262 and 263 trading days.          *;
* On a two-line plot, use 263 invisible tick marks. *;
%CreateTradeDayHtickmarkValues(data=DOW19901991, filter=%str(Year eq 1991));
axis1 label=none major=none minor=none style=0
  order=1 to &NumberOfTickMarks by 1
  value=(%GetHtickmarkValues);

%IdentifyYaxisCriticalPoints(data=DOW19901991, yVar=Dow, yVarFormat=4.);
axis2 label=none major=none minor=none style=0
  order=&yAxisStartTickMarkValue to &yAxisEndTickMarkValue by &yAxisIncrement
  value=("&yAxisStartTickMarkDisplay" "&yAxisEndTickMarkDisplay");

```

```

title2 font='Verdana'
justify=CENTER
height=9pt
"Dow Index 1990-1991"
justify=CENTER
"Start=&yVarStart, End=&yVarEnd";

%IdentifyXdatesForCriticalPoints(data=DOW19901991,yVar=DOW,xDateVar=date);
title3 font='Verdana' height=9pt
  "&LocateMinAndOrMaxWhenUnique";

symbol1 color=CXFF0000 interpol=join v=none w=1;
symbol2 color=CX0000FF interpol=join v=none w=1;

legend1 label=none shape=symbol(15,1);

proc gplot data=DOW19901991;
plot Dow*TradingDay=Year / haxis=axis1 vaxis=axis2 legend=legend1
  vref=&yAxisStartTickMarkValue &yAxisEndTickMarkValue cvref=CX333333 lvref=35;
  /* dark gray dotted reference lines at top and bottom of vaxis range */
run; quit;

```

#### Code for Figure 14

```

%CommonPreliminaryCode(PathToGraph=&ThePath,FigureNumber=14);
goptions htext=9pt ftext='Verdana';

footnote1 angle=+90 font=none height=1 ' ';
footnote2 angle=-90 font=none height=1 ' ';
footnote3 angle=0 font=none height=1 ' ';
title1 font=none height=1 ' ';

%IdentifyYaxisCriticalPoints(data=BEX2007April,yVar=BEX,yVarFormat=2.);
title2 font='Verdana'
justify=CENTER
height=9pt
"BEX Index in April 2007"
justify=CENTER
"Start=&yVarStart, End=&yVarEnd";
%IdentifyXvalsForCriticalPoints(data=BEX2007April,yVar=BEX);
title3 font='Verdana' height=9pt
  "High=&yAxisEndTickMarkDisplay"
  "&AtXvalueWhenMaxYisUnique"
  justify=CENTER
  "Low=&yAxisStartTickMarkDisplay"
  "&AtXvalueWhenMinYisUnique";

* In situations where SAS/GRAPH would insist on
  rendering multi-digit tick mark values vertically,
  this macro delivers them horizontally.
  Ambiguity as to location of the tick mark vs. the point
  can be eliminated with reference lines. *;
%BetterDayOfMonHtickmarkValues(data=BEX2007April);
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 value=none;
symbol1 color=CX000000 interpol=join v=none;
proc gplot data=BEX2007April;
plot BEX*Day / haxis=axis1 vaxis=axis2
  href=1 8 15 22 30 chref=CX333333 lhref=35;
  /* dark gray dotted reference lines at one-week intervals of haxis range */
run; quit;

```