Intelligent Production Graphic Reporting Applications
LeRoy Bessler, Miller Brewing Company

Abstract and Introduction
Production graphic reporting applications are those run by a mainframe computer center on a pre-determined schedule, typically getting input data from sources that are computer-updated on a pre-determined schedule.

The formats of the graphs are pre-set. There is no opportunity for ad-hoc manual modification of SAS/GRAPH programs to update data-date-dependent textual content (e.g., data-date references in titles), custom axis definitions or custom tick-mark assignments which need to change from run to run in a data-sensitive manner (but for which SAS/GRAPH defaults are unacceptable), run-date-dependent selection of input, etc.

Custom macros, as well as standard SAS/GRAPH and SAS* features, can be used to build a self-contained, hands-off graphic reporting machine—which is date-sensitive and data-sensitive, and which respects application-specific graphic design standards.

This paper is offered as a tutorial, but one featuring the author’s design principles, demonstrated by practical implementation. Details of SAS and SAS/GRAPH use are “taught” by publication of complete example-supporting code for self-study by the reader, not by explicit instructional commentary.

Graphics Intelligence
It is accurate to say that there already is graphics intelligence in SAS/GRAPH default processing. Unfortunately, defaults by definition and design can not address all application-specific needs.

Intelligence can be built into a graphic program, using custom SAS macros, global variables, SAS functions, etc. Intelligence enables the program to change itself in response to changes in the run date, in the data date or data-date range, or in the data range. Whether and how the program changes are a function of application design.

Reusable Building Blocks
The reusable building blocks are SAS macros. There are three varieties (in order of increasing generality): (a) application-specific; (b) site-specific; and (c) general-purpose.

Fire Walls
Any changeable parameters (e.g., the location of a reference line) can be supplied only via control card files, NOT by modifying the hard-coded graphic program.

Data to be depicted is supplied via external files only.

There are no manual date changes. In programs they would compromise program integrity. In control cards they could be forgotten. Data-selection dates are computer-determined.

Years-back ranges (for monthly trend lines) are governed through control cards. Changes to the span of interest are unlikely and infrequent, so that the manual update won’t be forgotten if, in fact, it’s ever needed.

Powerful Packaging: Standard Templates
Site-standard and application-standard graphic templates (not to be confused with SAS/GRAPH’s TEMPLATEs for PROC GREPLAY) facilitate consistent presentation.

The presentation formats become familiar. There is no need for page-to-page “adjustment of the perception/interpretation mechanism”. The reader/viewer knows what to expect and where to find things quickly. As the use of the templates proliferates through ad-hoc or other production applications, adjustment becomes unnecessary even from report to report.

Among the benefits of use of standard templates is reduction of the time and effort required to develop new, and to enhance existing, graphic applications.

Data Currency and Integrity
By, after automatic extraction from the data, displaying the data date or the data-date range in a title, a means is provided for the reader/viewer to quickly ascertain the currency and completeness of the data.

Scheduled production jobs assure currency of some of the computer-updated data. Some data providers may be responsible for running personally-controlled jobs to provide other of the computer-updated data. Some data providers may manually key data. In every case, the data date is required on each record.
In an environment where every data record is dated, plan ahead. Even if only last month's data must be graphed as a bar chart, be sure to keep all prior months' data. Anticipate the almost inevitable future request for trend analysis.

The Perils of Pie Charts

Pie charts are extremely popular for presentations.

However, though presentation graphs can be created ad-hoc, "hands on", and iteratively, production graphs must get it right the first time, every time.

Consider the pie chart in Figure 1. It tries to show the relative size (and list the magnitude) of population of the twelve countries in the European Community.

If you have, say, 5 or 6 entities whose pie slices are approximately the same size, all the slices will be apparent, and SAS/GRAPH will have no problem putting down all the slice labels, with adequate white space between them. For Figure 1, this is clearly not the case.

For some applications, lumping small slices into an "Other" category is unacceptable. For a production "hands-off" application to have a slice alternately appear and disappear from month to month, depending on size, is irritating, if not unacceptable. Furthermore, if you put so many slices into "Other" that it becomes non-negligible, there will be questions as to its detailed content.

Finally, with the vagaries of SAS/GRAPH's pie-chart text handling, sizes and positions of slices may be such that some labels are poorly positioned (and very likely to have some adjacent pairs squashed together in an ugly manner if you request labels for PERCENT in addition to SLICE and VALUE), if they don't disappear completely.

So, Consider the Bread Chart

The solution—if you want a visual display of the relative size, the value, and the percent of whole, with reliably positioned and never suppressed text, and in a quickly digested format—is the Bread Chart. See Figure 2. The Bread Chart is like a very long (what we in America call) French bread cut into chunks of different lengths, which are then laid parallel, and largest to smallest, rather than end to end. (In view of my choice of solid, rather than empty, fill, it looks more like black bread than French bread.)

All the macros used in the Bread Chart program are listed in the Appendix. The potential benefit of the TXTPRMS macro invoked by HBAR1D, and of the seemingly complicating macros for TITLES and FOOTNOTEs, is that they make it easy to define an application-default (not a SAS-default) font, height, etc. for text parts of graphs. Obviously, these text-oriented macros are not essential to the success of the Bread Chart concept. They only make it easier to achieve consistent application-specific customization of text from graph to graph.

There are two reasons for (through automation) building the percent-of-whole as part of the bar label, rather than asking the SAS/GRAPH HBAR function to compute and list the percent along the right margin (via options PERCENT and FREQ= response-variable-name): (1) SAS/GRAPH provides two decimal positions whether or not you want them; and (2) you have no control over height, font, and upper-case vs. mixed-case for the PERCENT column heading. One rarely needs two decimal positions on a percent to make a business inference or decision. All-upper-case is a hold-over from the bad old days of early data processing, and is not representative of standard business communication style. (As a matter of fact, paragraphs in all upper case type are harder to read.)

The Bread Chart answers all too common question, "How is the loaf divided up?"

Trend Lines

Trend lines are a common application. Here I show only the case of monthly graphs and a single trend line. Other periodicities are handled analogously. Multiple trend lines on a graph is a minor, straightforward extension, but not shown here.

The features and benefits of the YMPLOT1 macro are best presented by simply referring you to the graphic examples in Figures 3 and 4 (which also include the associated SAS/GRAPH programs), and the underlying macros listed in the Appendix. Many of the macros carry comments on the MACRO statement. Each macro is discussed in text at the bottom of its exhibit.

For explanation of macro syntax, global variables, SYMPUT, etc., I refer the reader to SAS-Institute-provided documentation (e.g., SAS Guide to Macro Processing). The principles and techniques used in my macros are by no means arcane. (Running the sample program for Figure 4 with OPTIONS MPRINT will list the SAS-generated final run-time code in the SAS log file.)

I will close with comments on some aspects of the design of YMPLOT1.

For the example in Figure 4, the data was cut off at April 1990 by specifying 9004 in the ddname RPTYYMM. Without a cut-off, YMPLOT1 will display data through the month previous to the month-of-run. In such a case—the typical situation of monthly reporting—specifying TSTRTMM = 1 and
TSTOPMM = 12 allowed retention of a January 1988
 to December 1990 time axis throughout 1990, as
 progressively more data became available. The
 horizontal axis of the graph remained constant month
 to month. Alternatively, if both TSTRTMM and
 TSTOPMM had been omitted, but the YRSFILE
 control-card value had been maintained at, as in this
 example, 2, the time axis would have been 25 months
 wide every month, ending at the current
 report-month.

 Fixing VMIN and VMAX likewise assured that the
 vertical axis of the graph remained constant month to
 month. Moreover, VMIN = 0 diminishes the
 likelihood of undue concern about what could, in
 fact, really be minor fluctuations month to month.
 In this example, where the units of the response
 variable are percent, fixing the axis maximum at the
 unequivocally maximum possible response is an
 eminently reasonable standardization.

 The benefit of the January reference lines coupled
 with the markers for the data points is obvious.

 The picture is actually compressed in the vertical
 direction only to enable the paper to meet the page
 count limit for publication. At this size, horizontal
 grid lines could have made it easier to get an
 approximation of the exact value of the
 measurements, if there was a desire for such. I look
 forward to using the less distracting option of dotted
 lines for grids in Version 6 of SAS/GRAPH.

 Notice
 *SAS/GRAPH and SAS are registered trademarks of
 SAS Institute Inc., Cary, NC, USA. DCF
 (Document Composition Facility) is a product of
 IBM Corp.

 Release 5.18 was used for this paper.

 The SAS code included in this paper was tested, and
 I believe it to be reliable. In any case, it can only be
 presented on an "as is" basis. Any code adopted by
 you should be tested by you, and you must assume
 responsibility for the consequences of its use. It must
 be tested, and might require modification, for
 compatibility with Version 6.

 This material has been previously published. See the
 author's paper of the same title in: Proceedings of the
 First Annual Regional Conference of the MidWest
 SAS Users Group (Copyright 1990 by MWSUG,
 USA). See also the author's paper of the same title
 in: WISAS Proceedings, Volume 2, June Issue
 (Edited by LeRoy Bessler. Fox Point, Wis., USA:

 Author
 Dr. LeRoy Bessler
 Miller Brewing Company
 P.O. Box 482
 Milwaukee, Wisconsin 53201-0482, USA
 Telephone 414-931-2773
 FAX 414-931-3474
Population in the European Community, 1989
(in Millions)

Source: "The World Factbook 1989"

Figure 1. Unsuccessful Pie Chart, and SAS/GRAFH Program Listing: Missing three labels. Program is blameless.
Population in the European Community, 1989

<table>
<thead>
<tr>
<th>Country</th>
<th>Share</th>
<th>Millions</th>
</tr>
</thead>
<tbody>
<tr>
<td>West Germany</td>
<td>19%</td>
<td>61.0</td>
</tr>
<tr>
<td>Italy</td>
<td>18%</td>
<td>57.6</td>
</tr>
<tr>
<td>U.K.</td>
<td>18%</td>
<td>57.0</td>
</tr>
<tr>
<td>France</td>
<td>17%</td>
<td>56.0</td>
</tr>
<tr>
<td>Spain</td>
<td>12%</td>
<td>39.4</td>
</tr>
<tr>
<td>Netherlands</td>
<td>5%</td>
<td>14.8</td>
</tr>
<tr>
<td>Portugal</td>
<td>3%</td>
<td>10.5</td>
</tr>
<tr>
<td>Greece</td>
<td>3%</td>
<td>10.0</td>
</tr>
<tr>
<td>Belgium</td>
<td>3%</td>
<td>9.9</td>
</tr>
<tr>
<td>Denmark</td>
<td>2%</td>
<td>5.1</td>
</tr>
<tr>
<td>Ireland</td>
<td>1%</td>
<td>3.6</td>
</tr>
<tr>
<td>Luxembourg</td>
<td>0%</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Source: "The World Factbook 1989"

OPTIONS DQUOTE;
DATA TOPCT;
INFILE INDAT.A.;
INPUT @1 COUNTRY $12.  @18 POP $4.1;
RUN;
ZAPHDPCT(DATA=TOPCT,
  MIDPOINT=COUNTRY,
  RESPONSE=POP,
  OUT=TOCHART,
  OUTNAME=COUNTRY,
  OUTVALUE=POP);
/* put OPTIONS here */
ZHBAR3D(DATA=TOCHART,
  MIDPOINT=COUNTRY,
  RESPONSE=POP,
  MPTLABEL="Share",
  RSPLABEL="Millions",
  HTTL=1.5,
  FITTFT=TEPLEX);
XTTL(TITLOG=1,
  TEXT="Population in the European Community, 1989");
ZFOOT(FOOTNO=1,
  TEXT="Source: "The World Factbook 1989";
RUN;

Figure 2. Bread Chart, and SAS/GRAPH Program Listing: Shares ranked by size; names, percents, and values shown.
Demand on Facility, in Percent of Capacity
By Month, January 1990 to January 1991

Figure 3. Defaulted Custom Plot by Year-Month, and SAS/GRAPH Program Listing: Using YMPLLOT1 Macro.

OPTIONS DQUOTE;
DATA TOSELECT;
INFILE DATA;
INPUT @. YMM $4.
    MEASURE 5.1;
RUN;
/* put COPTIONS here */
ZYMPLOT1(TICKSTEP=18,
    DATA=TOSELECT,
    VAR=MEASURE);
XTLT(XTLM=1),
    TEXT='Demand on Facility, in Percent of Capacity');
XTLTBYMD(XTLM=2);
RUN;
Demand on Facility, in Percent of Capacity
By Month, January 1988 to April 1990

90% is threshold for adverse impact on service

Figure 4. Optioned Custom Plot by Year-Month, and SAS/GRAPH Program Listing: Using YMPLOT1 Macro.
Appendix - Macros for Intelligent Graphic Reporting Applications

**XMacro AINPUT**

```plaintext
/* input file */
/* input file */
/* return */
/* return */
/* output */
/* output response variable name */
/* value */
```

**DATA**

```plaintext
SET SELECT:
SET MACRO= DATA;
SET RESPONSE RESPONSE;
RUN:
PROC FREE DATA;APEND:
SET STANDARD:
SET RESPONSE RESPONSE;
RUN:
PROC FREE DATA;APEND:
```

**APENDPT:**

Determine preferred default for each variable, and append that default at the end of the inputlist description.

**XMacro DATRANG**

```plaintext
/* first 8 characters of DATRANG macro */
/* data start year */
/* data start year */
/* data data year */
/* data start year */
/* data stop year */
/* data start year */
/* data start year */
/* data stop year */
/* data start year */
```

**RANGE:**

```plaintext
GETRANGE:
DATA RANGE:
CALL SYMPUT('RANGE',RANGE);
RUN:
END;
```

**GETRANGE**

Assign global variables to January pre-month values to use for reference reference line.

**XMacro GETREF**

```plaintext
/* input data */
/* input year */
/* input reference */
/* input reference */
/* input reference */
/* input reference */
```

**GETREF**

Assign global variables to be used later for the VALUE parameter specifications for the response axis. (To control data, return global variables for start, end, and increment of response axis.)

**XMacro FOUT**

```plaintext
/* output file */
/* output file */
/* output response variable name */
/* value */
```

**FOOT:**

Determine FOOTNOTE number and text. Barrow global variables for HEIGHT and FONT.

**XMacro FOOTNOTEP**

```plaintext
/* output file */
/* output response variable name */
/* value */
```

**FOOTTOP:**

Determine FOOTNOTE number and description of reference value. Barrow global variables for HEIGHT, FONT, and reference value to be included in text.
MACRO HBARID (DATA=); /* input file */  
HBARID();  /* same as HBARID */  
RESPONSE();  /* name of response variable */  
MIDPOINT();  /* name of midpoint variable */  
RESPLABEL();  /* 16-char label of RESPONSE */  
FILL();  /* fill for bars - EMPTY | SOLID */  
HTEXT1();  /* height for all HTEXT */  
HTEXT2();  /* font for all TEXT */  
HTEXT3();  /* height for all TEXT */  
{MACRO HBARID}END;  
PROC GCHART DATA=DATA;  
XIF FILL = SOLID THEN HB;  
PATTERN VISDOL C=BLACK;  
XEND;  
ELSE HB;  
PATTERN V=EMPTY C=BLACK;  
XEND;  
HBAR HBPOINT /  
SUM SVAR = RESPONSE  
MID = MIDPOINT  
BAXIS = HBAXIS  
DESCENT =  
AXIS1 MAJOR = NONE  
MINOR = NONE  
STYLE = 8  
AXIS2 LABEL = NONE  
VALUE = NONE  
STYLE = 6  
LABEL BRESHOLE = RESPLABEL;  
LABEL BRESPONSE = RESPLABEL;  
{MACRO HBARID}END;  
MACRO MAXLIST(ITEM=);  
XDO 1 = 1  
{MACRO MAXLIST}END;  
MAXLIST:  
Retrieve the global variables to define the list of values to be assigned to a parameter.  
MACRO MARKER(MARKER=);  
SOLUTION V;  
XIF MARKER = "YES"  
THEN XDO;  
CALL SYMPUT('V','"NO"');  
XEND;  
MARKER:  
If a plot-point marker is asked for, assign the standard marker to a global variable. Else, assign no marker as the default.  
MACRO PLOTLIN(ITEM=DATA,FLOTVVAR,PLOTVAR,VARI=VARISMD);  
SYM1 = JOIN V=F  
L=1 M=4 C=BLACK;  
PROC SPLIT DATA=DATA;  
PLOT EPLTIVVAR=PLOTVAR /  
VARI=VARISMD  
XIF (EPLTIVVAR = V) AND (JNAMES = YES) THEN XDO;  
HEIGHT = MAXLIST(ITEM=);  
LHEIGHT = 1;  
XEND;  
PLOT EPLTIVVAR=PLOTVAR /  
VARI=VARISMD;  
{MACRO PLOTLIN}END;  
PLOTLIN:  
Do a single-line plot, using line width 4. Provide tick marks at the right side. Include a reference line for the response variable, if a reference value is provided. If no midpoint variable is year-month, provide reference lines at each January, if requested.  
MACRO RAXIS(BAXIS=VARISMD);  
AXISVARISMD LABEL = NONE  
MAJOR = NONE  
MINOR = NONE  
STYLE = 1  
BAIXIS TO BAXISVARISMD BY BAXISSTEP  
VALUE = 1  
HTEXT1=HTEXT F=HTEXT MPLTXY LABEL(ITEM=TERM=))  
{MACRO RAXIS}END;  
RAXIS:  
Retrieve global variables to provide the response-axis definition.  
MACRO RPDATE(DATA=);  
SOLUTION RPVVAR RPRTY RPRTMP;  
RPV VAR = "" THEN XDO;  
DATA NULL.;  
INFILE DATA;  
INPUT RPV 54.;  
RPRTY 12.;  
RPRTMP 22.;  
CALL SYMPUT('RPVVAR',RPVVAR);  
CALL SYMPUT('RPRTY',RPRTY);  
CALL SYMPUT('RPRTMP',RPRTMP);  
RUN;  
{MACRO RPDATE}END;  
RPDATE:  
Assi...
DATA _NULL_;