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 cannot 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.

With the vicissitudes of the PROC GCHART PIE function, one cannot know in advance that every slice will be labelled. An example of the problem, and the exposition of its solution, are provided in the author's paper Pie Charts and Bar Charts: Getting Their Best out of SAS/GRAPH Software, elsewhere in these Proceedings.

Trend Lines

Trend lines are a common application, and typically featured in periodic management reports. 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 YMPLLOT macro are best presented by simply referring you to the graphic examples in Figures 1 and 2 (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 program for Figure 2 with OPTIONS MPRINT MLOGIC SYMBOLGEN will list the SAS-generated run-time code in the SAS log.)

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

For the example in Figure 2, the data was cut off at April 1990 by specifying 9004 in the file with ddname RPTYYMM. Without a cut-off, YMPLLOT 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 23 months wide every month, ending at the current report-month, providing a constant tick-mark density.

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 minimize the paper's page count. At this size, horizontal grid lines could have made it easier to get an approximation of the exact value of the measurements, if there were a desire for such. Version 6 of SAS/GRAPH offers the less distracting option of dotted lines for grids. If detail look-up is really needed, a companion table is a better solution.

For more on design of graphs, tables, and reports, see the author's papers Effective and Efficient Use of SAS/GRAPH Software and Effective and Efficient Information Delivery for Executive Management, elsewhere in these Proceedings.

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

SAS System 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.

Author

Dr. LeRoy Bessler
Miller Brewing Company
P.O. Box 482
Milwaukee, Wisconsin 53201-0482, USA
Telephone 414-931-2773
OPTIONS DQUOTE;
DATA TOSELECT;
INFILE INDATA;
INPUT @1 YYMM $4.
@10 MEASURE $5.1;
RUN;
/* put GOPTIONS here */
%YMPLOT1(TICKSTEP=10,
DATA=TOSELECT,
VAR=MEASURE);
%TTL(TTNG=1,
TEXT='Demand on Facility, in Percent of Capacity');
%TTLBY(MO(TTNG=2);
RUN;

Figure 1. Defaulted Custom Plot by Year-Month, and Program Listing: Using YMPLOT1 Macro, in February 1991.
Demand on Facility, in Percent of Capacity
By Month, January 1988 to April 1990

90% is threshold for considering capacity increase

Figure 2. Optioned Custom Plot by Year-Month, and Program Listing: Using YMPLOT1 Macro.
## Appendix - Macros for Intelligent Graphic Reporting Applications

### DATRANG

**Description:** Determine start and stop values for year-month tick marks, and override start value, if any, for year-month data selection. Assign these values to global variables. Overwrite report-year-month global variables as appropriate.

#### DATRANG Macro

```bash
MACRO DATRANG
  (DATA, DSTRTM, DSTOPM, DSTRTM, DSTOPM, LSTRTM, DSTOPM)
  /-
  if DSTRTM = "", then
  DSTRTM = DSTRTM1;
  end;
  if DSTOPM = "", then
  DSTOPM = DSTOPM1;
  end;
  if DSTRTM = "", then
  DSTRTM = DSTRTM1;
  end;
  if DSTOPM = "", then
  DSTOPM = DSTOPM1;
  end;
  call SYMPUT('DSTRTV', LEFT(DSTRTV));
  call SYMPUT('TSTRTV', LEFT(OSTRTV));
  end;
  if &DATA = "", then
  call SYMPUT('DSTRTM', LEFT(DSTRTM));
  call SYMPUT('OSTRTM', LEFT(OSTRTM));
  else
  call SYMPUT('DSTRTM', &DATTMM);
  call SYMPUT('OSTRTM', &DATTMM);
  end;
  call SYMPUT('DSTRTV', &DATE);
  call SYMPUT('TSTRTV', &DATE);
  if LINT > 3 then
  call SYMPUT('FMT', COMMA, LEFT(LINT));
  end;
  end;
end DATRANG;
```

### GetREF

**Description:** Assign global variables for i-year-month values to be used for vertical axis tick marks.

#### GetREF Macro

```bash
MACRO GetREF
  (DATA=)
  if DATA = "", then
  DATA_NULL;
  else
  call SYMPUT('DSTRTV', LEFT(DSTRTV));
  call SYMPUT('TSTRTV', LEFT(TSTRTV));
  end;
end GetREF;
```

### GetRTKS

**Description:** Assign global variables to be used later for the VALUE parameter specifications for the response axis. (To control ticks, retrieve global variables for start, end, and increment of axis.)

#### GetRTKS Macro

```bash
MACRO GetRTKS
  (DATA=
  if DATA = "", then
  DATA_NULL;
  else
  call SYMPUT('RTKFILE', LEFT(REFVAR));
  end;
end GetRTKS;
```

### GetTKS

**Description:** Define FOOTNOTE number and text. Retrieve global variables for HEIGHT, FONT, and vertical referent lines.

#### GetTKS Macro

```bash
MACRO GetTKS
  (FOOTNO=, TEXT=)
  if FOOTNO = "", then
  FOOTNO = FOOTNO1;
  if TEXT = "", then
  TEXT = TEXT1;
  end;
  call SYMPUT('FMT', COMMA, LEFT(LINT));
  end;
end GetTKS;
```

### MakLIST

**Description:** Retrieve the global variables to develop the list of values to be assigned to a parameter.

#### MakLIST Macro

```bash
MACRO MakLIST
  (ITEM1)
  i = 1 + 1 * 100 * ITEM1;
  call SYMPUT('RTKFILE', LEFT(REFVAR));
  end;
end MakLIST;
```

### MarkR

**Description:** Assign global variables for i-year-month values to be used for vertical axis tick marks.

#### MarkR Macro

```bash
MACRO MarkR
  (MARKER=)
  if MARKER = "", then
  MARKER = MARKER1;
  end;
end MarkR;
```

### PlotLINE

**Description:** Do a single-line plot, using line width 4. Provide tick marks at right side. Include a reference line for the response variable, if a reference value is provided. If the referenced variable in year-month, provide reference lines at each January, if requested.

#### PlotLINE Macro

```bash
MACRO PlotLINE
  (DATA=, PLOTVVAR, PLOTHVAR, XAXISNO, YAXISNO=, VARCOUNT=)
  if XAXISNO = "", then
  XAXISNO = XAXISNO1;
  if YAXISNO = "", then
  YAXISNO = YAXISNO1;
  end;
  call SYMPUT('V', LEFT(VAXIS));
  call SYMPUT('H', LEFT(HAXIS));
  end;
end PlotLINE;
```
Macros and Statements for Creating Datasets and Manipulating Data

**Macro RAXIS: Define Global Variables for Axis Definitions**

- **RAXIS**
  - **RAXISRTDATE**: Assign global variables for a report date as year-month, year, month.
  - **RAXISRTVMM**: Assign global variables for the report date as year-month, year.
  - **RAXISRTMM**: Assign global variables for the report date as month.
  - **RAXISRTVVVV**: Assign global variables for the report date as day, month, year.

- **RAXISRTDATE**
  - **RAXISRTVMM**
  - **RAXISRTMM**
  - **RAXISRTVVVV**

- **RAXISRTMM**
  - **RAXISRTVMM**
  - **RAXISRTMM**
  - **RAXISRTVVVV**

- **RAXISRTMM**
  - **RAXISRTVMM**
  - **RAXISRTMM**
  - **RAXISRTVVVV**

**Macro RXSPRMS: Assign Global Variables for Start, End, and Increment**

- **RXSPRMS**
  - **RXSPRMS**: Assign global variables for start, end, and increment.
  - **RXSPRMS**: Assign global variables for start, end, and increment.
  - **RXSPRMS**: Assign global variables for start, end, and increment.
  - **RXSPRMS**: Assign global variables for start, end, and increment.

**Macro SAVLAST: Save Last Value for Use Later**

- **SAVLAST**
  - **SAVLAST**: Save last value for use later.
  - **SAVLAST**: Save last value for use later.
  - **SAVLAST**: Save last value for use later.
  - **SAVLAST**: Save last value for use later.

**Macro TXTFRMS: Assign Global Variables for Text Formatting**

- **TXTFRMS**
  - **TXTFRMS**: Assign global variables for text formatting.
  - **TXTFRMS**: Assign global variables for text formatting.
  - **TXTFRMS**: Assign global variables for text formatting.
  - **TXTFRMS**: Assign global variables for text formatting.

**Macro VARLIST: Assign Global Variables for List Specification**

- **VARLIST**
  - **VARLIST**: Assign global variables for list specification.
  - **VARLIST**: Assign global variables for list specification.
  - **VARLIST**: Assign global variables for list specification.
  - **VARLIST**: Assign global variables for list specification.
YMHAXIS: Retrieve global variables to provide the year-month axis definition. The axis must be horizontal, though this definition is in a format that, for tick marks to be used, give their design (see an example for details).

YMHAXIS: Assign global variables to be used for the VALUE and ORDER parameter specifications for the year-month axis. (Retrieves global variables for tick-mark start and stop to set bounds.)

YMHAXIS: Use by the YMHITXS macro. Uses a site-specific format for initial character of month-name.