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.

Intelligenee 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 YMPLOT1 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 YMPLOT1.

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, 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, 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.

Notices

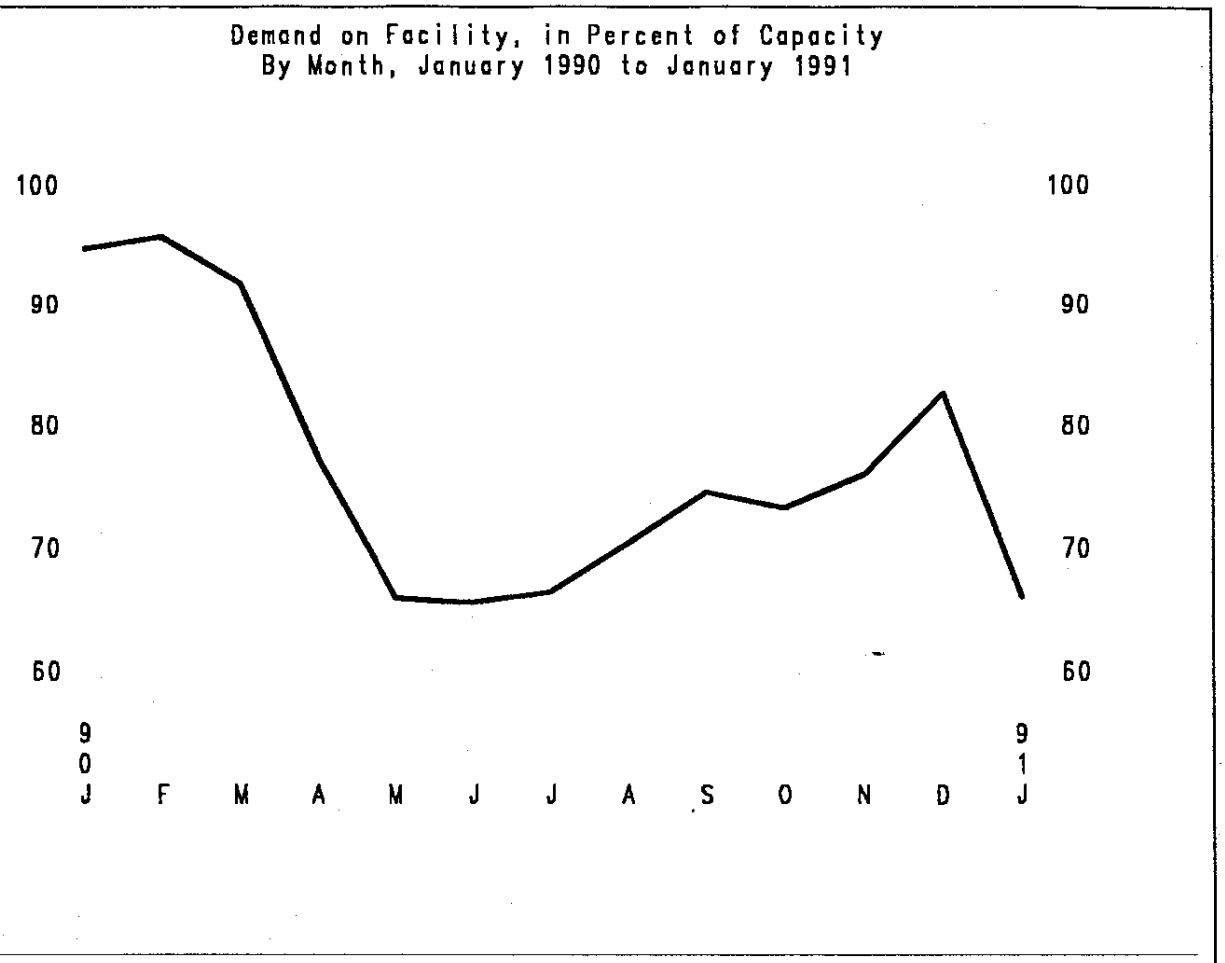
For more on design of graphs, tables, and reports, see the author's papers *Effective and Efficient Use of SAS/GRAPH Software* and *Effective and Efficient Information Delivery for Executive Management*, elsewhere in these *Proceedings*.

SAS 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.*

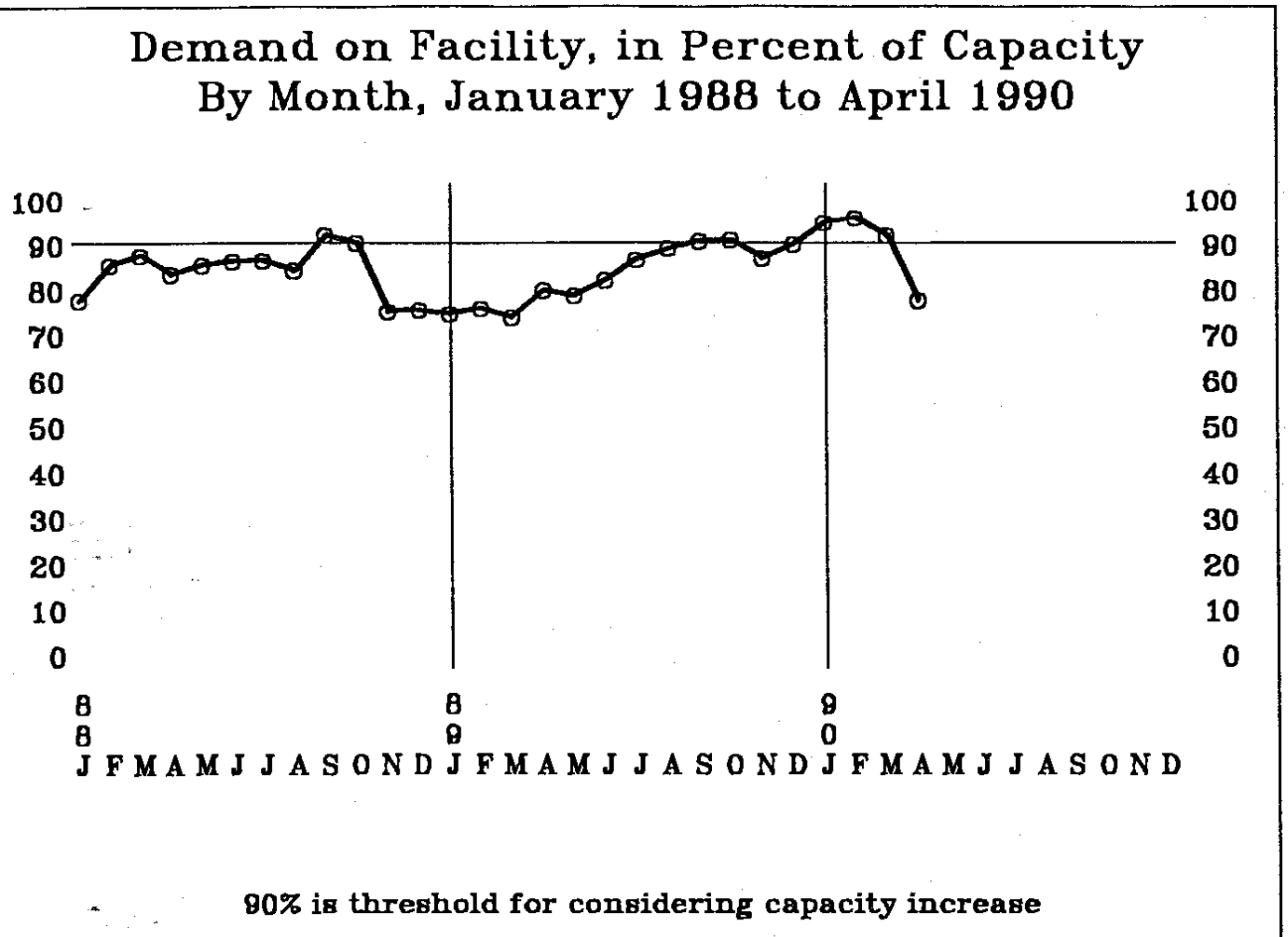
Author

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.
      @18 MEASURE   5.1;
RUN;
 /* put GOPTIONS here */
%YMPLOT1(TICKSTEP=10,
         DATA=TOSELECT,
         VAR=MEASURE);
%TTL(TTLNO=1,
 TEXT='Demand on Facility, in Percent of Capacity');
%TTLBYMO(TTLNO=2);
RUN;
```

Figure 1. Defaulted Custom Plot by Year-Month, and Program Listing: Using YMPLOT1 Macro, in February 1991.




```
OPTIONS DQUOTE;
DATA TOSELECT;
INFILE INDATA;
INPUT @1  YYMM      $4.
      @18 MEASURE   5.1;
RUN;
 /* put GOPTIONS here */
%YMPLOT1(TICKSTEP=10,
         DATA=TOSELECT,
         VAR=MEASURE,
         RDATFILE=RPTYYMM,  /* control card containing 9004 */
         YRSFILE=YRSBACK,   /* control card containing    2 */
         REFFILE=REFLINE,   /* control card containing   90 */
         SUMFILE=SUMMARY,   /* produce output summary file  */
         JANREFS=YES,
         MARKER=YES,
         DSTRTMM=1,
         TSTRTMM=1,
         TSTOPMM=12,
         VMIN=0,
         HTTL=1.5,
         FTTLFT=TRIPLEX,
         FTXT=TRIPLEX);
%TTL(TTLNO=1,
 TEXT='Demand on Facility, in Percent of Capacity');
%TTLBYMO(TTLNO=2);
%FOOTREF(FOOTNO=1,
 TEXT='% is threshold for adverse impact on service');
RUN;
```

Figure 2. Optioned Custom Plot by Year-Month, and Program Listing: Using YMPLOT1 Macro.

Appendix - Macros for Intelligent Graphic Reporting Applications

```
%MACRO DATRANG
  (DATA=,DSTRTMM=.,TSTRTMM=.,TSTOPMM=.);
 /* %RPTDATE must precede invocation of DATRANG macro */
%GLOBAL DSTRTY;  /* data start year       */
%GLOBAL DSTRTM;  /* data start month      */
%GLOBAL DSTRTYM; /* data start year-month */
%GLOBAL DSTOPY;  /* data stop  year       */
%GLOBAL DSTOPM;  /* data stop  month      */
%GLOBAL DSTOPYM; /* data stop  year-month */
%GLOBAL TSTRTY;  /* tick start year       */
%GLOBAL TSTRTM;  /* tick start month      */
%GLOBAL TSTOPY;  /* tick stop  year       */
%GLOBAL TSTOPM;  /* tick stop  month      */
DATA _NULL_;
%IF &DATA ¬= ' ' %THEN %DO;
  INFILE &DATA;
  INPUT YRSBACK 2.;
  DSTRTY = &RPTYY - YRSBACK;
%END;
%ELSE %DO;
  DSTRTY = &RPTYY - 1;
%END;
CALL SYMPUT('DSTRTY',LEFT(DSTRTY));
CALL SYMPUT('TSTRTY',LEFT(DSTRTY));
IF &DSTRTMM ¬= . THEN
  DSTRTM = &DSTRTMM;
ELSE
  DSTRTM = &RPTMM;
CALL SYMPUT('DSTRTM',LEFT(DSTRTM));
IF &TSTRTMM ¬= . AND &TSTRTMM ¬> DSTRTM THEN
  CALL SYMPUT('TSTRTM',&TSTRTMM);
ELSE
  CALL SYMPUT('TSTRTM',LEFT(DSTRTM));
CALL SYMPUT('DSTRTYM',LEFT((DSTRTY * 100) + DSTRTM));
CALL SYMPUT('DSTOPY',&RPTYY);
CALL SYMPUT('DSTOPM',&RPTMM);
CALL SYMPUT('DSTOPYM',&RPTYYMM);
CALL SYMPUT('TSTOPY',&RPTYY);
IF &TSTOPMM ¬= . AND &TSTOPMM ¬< &RPTMM THEN
  CALL SYMPUT('TSTOPM',&TSTOPMM);
ELSE
  CALL SYMPUT('TSTOPM',&RPTMM);
RUN;
%MEND;
```

DATRANG: Determine start and stop values for year-month tick marks, and override start value, if any, for year-month input data selection. Assign those values to global variables. Use retrieved report-year-month global variables as appropriate.

```
%MACRO FOOT(FOOTNO=,TEXT=,JFOOT=CENTER);
FOOTNOTE&FOOTNO
  H=&HFOOT F=&FTIFO J=&JFOOT &TEXT;
%MEND;
```

FOOT: Define FOOTNOTE number and text. Retrieve global variables for HEIGHT and FONT.

```
%MACRO FOOTREF(FOOTNO=,TEXT=);
FOOTNOTE&FOOTNO
  H=&HFOOT F=&FTIFO "&REFVAL"&TEXT;
%MEND;
```

FOOTREF: Define FOOTNOTE number and description of reference value. Retrieve global variables for HEIGHT, FONT, and reference value to be included in text.

```
%MACRO GETJREF(DOGET=);
%IF &DOGET=YES %THEN %DO;
DATA _NULL_;
%GLOBAL LASTJREF;
%GLOBAL %VARLIST(PREFIX=JREF,VARCOUNT=11); /* max is 10 yrs,
                                              Jan-Jan */
%LET BEGINYY = &TSTRTY + 1;
I = 0;
DO Y = &BEGINYY TO &TSTOPY;
  I = I + 1;
  CALL SYMPUT ('LASTJREF',I);
  CALL SYMPUT ('JREF'||LEFT(I),LEFT((Y * 100) + 1));
END;
RUN;
%END;
%MEND;
```

GETJREF: Assign global variables for January year-month values to use for vertical reference lines.

```
%MACRO GETREF(DATA=);
%IF &DATA ¬= ' ' %THEN %DO;
DATA _NULL_;
INFILE &DATA;
INPUT REFVAR;
%GLOBAL REFVAL;
CALL SYMPUT('REFVAL',LEFT(REFVAR));
RUN;
%END;
%MEND;
```

GETREF: Access file, if any, to assign global variable for value to use for response reference line.

```
%MACRO GETRTKS;
DATA _NULL_;
%GLOBAL FMTJ;
INTSTART = INT(ABS(&AXISTART));
INTEND   = INT(ABS(&AXISEND));
MAXINT   = MAX(INTSTART,INTEND,1);
LINT     = LENGTH(LEFT(RIGHT(MAXINT)))
   + INT((INT(LOG10(MAXINT)))/3); /* for comma(s), if any */
IF &AXISTART < 0
  THEN LINT = LINT + 1;
INTSTEP  = INT(&AXISSTEP);
LDEC     = LENGTH(LEFT(RIGHT(&AXISSTEP - INTSTEP))) - 2;
IF &AXISSTEP = INTSTEP THEN DO;
  IF LINT > 3 THEN
    CALL SYMPUT('FMTJ','COMMA'||LEFT(LINT)||'.');
  ELSE
    CALL SYMPUT('FMTJ',LEFT(LINT)||'.');
END;
ELSE
  CALL SYMPUT('FMTJ',LEFT(LINT + LDEC + 1)||'.'||LEFT(LDEC));
RUN;
DATA _NULL_;
%GLOBAL LASTRTK;
%GLOBAL %VARLIST(PREFIX=RTK,VARCOUNT=100) ;
I = 0;
DO J = &AXISTART TO &AXISEND BY &AXISSTEP;
  I = I + 1;
  CALL SYMPUT ('LASTRTK',I);
  CALL SYMPUT ('RTK'||LEFT(I),LEFT(PUT(J,&FMTJ)));
END;
RUN;
%MEND GETRTKS;
```

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

```
%MACRO MAKLIST(ITEM=);
%DO I = 1 %TO &&LAST&ITEM;
  "&&&ITEM&I"
%END;
%MEND;
```

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

```
%MACRO MARKER(MARKER=);
DATA _NULL_;
%GLOBAL V;
%IF &MARKER=YES %THEN %DO;
  CALL SYMPUT('V','-');
%END;
%ELSE %DO;
  CALL SYMPUT('V','NONE');
%END;
RUN;
%MEND;
```

MARKER: If a plot-point marker was asked for, assign the standard marker to a global variable. Else, assign no marker as the default.

```
%MACRO PLTILIN(DATA=,PLOTVVAR=,PLOTHVAR=,VAXISNO=,HAXISNO=);
SYMBOL1 I=JOIN V=&V       L=1 W=4 C=BLACK;
SYMBOL2         V=NONE            C=BLACK;
PROC GPLOT DATA=&DATA;
PLOT &PLOTVVAR*&PLOTHVAR /
     HAXIS=AXIS&HAXISNO
     VAXIS=AXIS&VAXISNO
%IF &PLOTHVAR = YYMM AND &JANREFS = YES %THEN %DO;
     HREF = %MAKLIST(ITEM=JREF)
     LHREF = 1
%END;
%IF &REFFILE ¬= ' ' %THEN %DO;
    VREF  = &REFVAL
    LVREF = 1
%END;
    ;
PLOT2 &PLOTVVAR*&PLOTHVAR=2 / VAXIS=AXIS&VAXISNO;
%MEND;
```

PLTILIN: 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 the midpoint variable is year-month, provide reference lines at each January, if requested.

```
%MACRO RAXIS(RAXISNO=);
AXIS&RAXISNO LABEL = NONE
      MAJOR = NONE
      MINOR = NONE
      STYLE = 0
      ORDER = &AXISTART TO &AXISEND BY &AXISSTEP
      VALUE = ( H=&HTEX F=&FTEX %MAKLIST(ITEM=RTK) );
%MEND;
```

RAXIS: Retrieve global variables to provide the response-axis definition.

```
%MACRO RPTDATE(DATA=);
%GLOBAL RPTYYMM RPTYY RPTMM RPTYYYY;
%IF &DATA ¬= ' ' %THEN %DO;
  DATA _NULL_;
  INFILE &DATA;
  INPUT @01 RPTYYMM $4.
        @01 RPTYY   $2.
        @03 RPTMM   $2.;
  CALL SYMPUT('RPTYYMM',RPTYYMM);
  CALL SYMPUT('RPTYY',RPTYY);
  CALL SYMPUT('RPTMM',RPTMM);
  FORMAT RPTYYYY $4.;
  RPTYYYY = 1900 + RPTYY;
  CALL SYMPUT('RPTYYYY',RPTYYYY);
  RUN;
%END;
%ELSE %DO;
  DATA _NULL_;
  YR=YEAR(TODAY());
  MO=MONTH(TODAY());
  IF MO > 1 THEN DO;
    SELECTMM = MO - 1;
    SELECTYY = YR - 1900;
  END;
  ELSE DO;
    SELECTMM = 12;
    SELECTYY = YR - 1901;
  END;
  SLCTYYMM = (SELECTYY * 100) + SELECTMM;
  CALL SYMPUT('RPTYYMM',SLCTYYMM);
  CALL SYMPUT('RPTYY',SELECTYY);
  CALL SYMPUT('RPTMM',SELECTMM);
  FORMAT RPTYYYY $4.;
  RPTYYYY = 1900 + SELECTYY;
  CALL SYMPUT('RPTYYYY',RPTYYYY);
  RUN;
%END;
%MEND;
```

RPTDATE: Assign global variables for the report date--as year-month, year, and month. Use input file, if any. Else, assume month previous to current month is desired.

```
%MACRO RXSPRMS(DATA=,VAR=,STEP=,AXISMIN=.,AXISMAX=.);
%GLOBAL AXISTART AXISEND AXISSTEP;
DATA _NULL_;
%GLOBAL FMTAEND;
INTSTEP  = INT(&STEP);
LDEC     = LENGTH(LEFT(RIGHT(&STEP - INTSTEP))) - 2;
IF &STEP = INTSTEP THEN
  CALL SYMPUT('FMTAEND','10.');
ELSE
  CALL SYMPUT('FMTAEND',LEFT(10||'.'||LEFT(LDEC)));
RUN;
PROC MEANS DATA=&DATA NOPRINT MIN MAX;
VAR &VAR;
OUTPUT OUT=MINMAX MIN=MINVAR MAX=MAXVAR;
RUN;
DATA _NULL_;
SET MINMAX;
CALL SYMPUT('AXISSTEP',&STEP);
FORMAT AXISTART 10.;
AXISTART = MINVAR - MOD(MINVAR,&STEP);
IF (&AXISMIN ¬= . AND AXISTART >= &AXISMIN)
  THEN AXISTART = &AXISMIN;
CALL SYMPUT('AXISTART',AXISTART);
MAXREMAI = MOD(MAXVAR,&STEP);
FORMAT AXISEND &FMTAEND;
IF MAXREMAI ¬= 0
  THEN AXISEND = MAXVAR + &STEP - MAXREMAI;
ELSE
      AXISEND = MAXVAR;
IF (&AXISMAX ¬= . AND AXISEND <= &AXISMAX)
  THEN AXISEND = &AXISMAX;
CALL SYMPUT('AXISEND',AXISEND);
RUN;
%MEND RXSPRMS;
```

RXSPRMS: Assign global variables for start, end, and increment of response axis. Invoker must supply increment. Invoker may specify start and/or end. Else, axis start or end is determined based on actual data range, and is placed at closest point that increment permits.

```
%MACRO SAVLAST(DATA=,VAR=,OUT=);
%IF &OUT ¬= ' ' %THEN %DO;
DATA _NULL_;
SET &DATA END=LAST;
RETAIN VALUE 'N. A.';
IF YYMM = &RPTYYMM THEN VALUE = &VAR;
IF LAST;
FILE &OUT;
PUT @1 VALUE;
RUN;
%END;
%MEND;
```

SAVLAST: Not for graph. Used to save number for all-graphs summary of report month elsewhere.

```
%MACRO TTL(TTLNO=,TEXT=);
TITLE&TTLNO H=&HTITLE F=&FTIFO &TEXT;
%MEND;
```

TTL: Define TITLE number and text. Retrieve global variables for HEIGHT and FONT.

```
%MACRO TTLBYMO(TTLNO=);
TITLE&TTLNO H=&HTITLE F=&FTIFO
 "By Month, &FIRSTMO &FIRSTYR to &LASTMO &LASTYR";
%MEND;
```

TTLBYMO: Define TITLE number. Retrieve global variables for HEIGHT, FONT, and text (dates).

```
%MACRO TTLDAT$(DATA=);
%GLOBAL FIRSTMO FIRSTYR LASTMO LASTYR;
PROC SORT DATA=&DATA OUT=SORTED; BY YYMM;
DATA _NULL_;
SET SORTED;
IF _N_ = 1;
FORMAT FIRSTMM 2.;
FIRSTMM = SUBSTR(YYMM,3,2);
FIRSTMO = PUT(FIRSTMM,MONTHUL.);
CALL SYMPUT('FIRSTMO',TRIM(FIRSTMO));
FIRSTYR = SUBSTR(YYMM,1,2) + 1900;
CALL SYMPUT('FIRSTYR',LEFT(FIRSTYR));
RUN;
PROC SORT DATA=&DATA OUT=SORTED; BY DESCENDING YYMM;
DATA _NULL_;
SET SORTED;
IF _N_ = 1;
FORMAT LASTMM 2.;
LASTMM = SUBSTR(YYMM,3,2);
LASTMO = PUT(LASTMM,MONTHUL.);
CALL SYMPUT('LASTMO',TRIM(LASTMO));
LASTYR = SUBSTR(YYMM,1,2) + 1900;
CALL SYMPUT('LASTYR',LEFT(LASTYR));
RUN;
%MEND;
```

TTLDAT$: Assign global variables for two month-years to define the data-date range for use in a title. Prepare the month names with a site-specific mixed-case format.

```
%MACRO TXTPRMS
      (HTTL=1, /* height for all %ttl____ macros            */
 FTTLFT=NONE, /* font for all %ttl____ & %foot___ macros    */
 HTXT=1,      /* height for other text if H can be specified*/
 FTXT=NONE,   /* font for other text if F can be specified  */
 HFT=1);      /* height for all %foot___ macros             */
DATA _NULL_;
%GLOBAL HTITLE;
CALL SYMPUT('HTITLE',LEFT(&HTTL));
%GLOBAL FTIFO;
CALL SYMPUT('FTIFO',"&FTTLFT");
%GLOBAL HTEX;
CALL SYMPUT('HTEX',LEFT(&HTXT));
%GLOBAL FTEX;
CALL SYMPUT('FTEX',"&FTXT");
%GLOBAL HFOOT;
CALL SYMPUT('HFOOT',LEFT(&HFT));
RUN;
%MEND;
```

TXTPRMS: Options are explained in comments on MACRO statement.

```
%MACRO VARLIST(PREFIX=,VARCOUNT=);
%DO I = 1 %TO &VARCOUNT;
  &PREFIX&I
%END;
%MEND;
```

VARLIST: Assign global variables for a list of values to be used later in specification of a parameter.

```
%MACRO YMHAXIS(YMHAXSNO=);
AXIS&YMHAXSNO LABEL = NONE
      MAJOR = NONE
      MINOR = NONE
      STYLE = 0
      ORDER = %MAKLIST(ITEM=YYMM)
      VALUE = (A=&TANGLE R=&TROTATE H=&HTEX F=&FTEX
               %MAKLIST(ITEM=HTK));
%MEND;
```

YMHAXIS: Retrieve global variables to provide the year-month axis definition. The axis must be horizontal--though axis definition is incapable of forcing that--for the tick marks to be useful, given their design (see an example for results).

```
%MACRO YMHTKS(TKFMT=YYM);
DATA _NULL_;
%GLOBAL TANGLE;
%GLOBAL TROTATE;
IF "&TKFMT" = 'YYM' THEN DO;
  CALL SYMPUT ('TANGLE' ,-90);
  CALL SYMPUT ('TROTATE',+90);
END;
ELSE DO;
  CALL SYMPUT ('TANGLE' ,0);
  CALL SYMPUT ('TROTATE',0);
END;
%GLOBAL LASTHTK;
%GLOBAL LASTYYMM;
%GLOBAL %VARLIST(PREFIX=HTK,VARCOUNT=121) ; /* max is 10 yrs,
                                              Jan-Jan */
%GLOBAL %VARLIST(PREFIX=YYMM,VARCOUNT=121); /* max is 10 yrs,
                                              Jan-Jan */
I = 0;
IF &TSTRTY < &TSTOPY THEN
  DO Y = &TSTRTY TO &TSTOPY;
   IF Y = &TSTRTY THEN
    DO M = &TSTRTM TO 12;
     %YMHTMAK(FMT=&TKFMT);
    END;
   ELSE
   IF Y = &TSTOPY THEN
    DO M = 1 TO &TSTOPM;
     %YMHTMAK(FMT=&TKFMT);
    END;
   ELSE
    DO M = 1 TO 12;
     %YMHTMAK(FMT=&TKFMT);
    END;
  END;
ELSE
  DO;
   Y = &TSTRTY;
   DO M = &TSTRTM TO &TSTOPM;
     %YMHTMAK(FMT=&TKFMT);
   END;
  END;
RUN;
%MEND YMHTKS;
```

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

```
%MACRO YMHTMAK(FMT=);
I = I + 1;
CALL SYMPUT ('LASTYYMM',I);
YYMM = (Y * 100) + M;
CALL SYMPUT ('YYMM'||LEFT(I),LEFT(YYMM));
CALL SYMPUT ('LASTHTK',I);
IF "&FMT" = 'YYM' THEN DO;
  MO = PUT(M,MOINITL.);
  IF M = 1 THEN YYM = PUT((LEFT(Y) || MO),$CHAR3.);
  ELSE YYM = PUT((' ' || MO),$CHAR3.);
  CALL SYMPUT ('HTK'||LEFT(I),YYM);
END;
ELSE
  CALL SYMPUT ('HTK'||LEFT(I),PUT(M,MMM.));
%MEND YMHTMAK;
```

YMHTMAK: Used by the YMHTKS macro. Uses a site-specific format for initial character of month-name.

```
%MACRO YMPLOT1(
TICKSTEP=,     /* increment between vertical tick marks      */
DATA=,         /* input data                                 */
VAR=,          /* variable to be plotted vs. year-month      */
RDATFILE=' ', /* ddname of file to specify last year-month
                 to plot; default is month prior to today's */
YRSFILE=' ',   /* ddname of file to specify number of years
                 back to plot; default is 1                 */
REFFILE=' ',   /* ddname of file to specify reference line   */
SUMFILE=' ',   /* ddname of file to output value for last
                 year-month of plot as summary information  */
JANREFS=NO,    /* reference lines at each January? YES/NO    */
MARKER=NO,     /* circle with dot at each data point? YES/NO */
DSTRTMM=.,     /* override for month number of first year-month
                 plotted; default is month number same as last
                 year-month plotted                         */
TSTRTMM=.,     /* override to extend year-month axis
                 prior to first year-month plotted          */
TSTOPMM=.,     /* override to extend year-month axis
                 past last year-month plotted               */
VMIN=.,        /* override to start vertical axis below
                 minimum needed for actual data             */
VMAX=.,        /* override to end vertical axis above
                 maximum needed for actual data             */
HTTL=1,        /* height for all %ttl____ macros             */
FTTLFT=NONE,   /* font for all %ttl____ & %foot___ macros    */
HTXT=1,        /* height for other text if H can be specified*/
FTXT=NONE,     /* font for other text if F can be specified  */
HFT=1);        /* height for all %foot___ macros             */
%RPTDATE(DATA=&RDATFILE);
%DATRANG
  (DATA=&YRSFILE,
   DSTRTMM=&DSTRTMM,TSTRTMM=&TSTRTMM,TSTOPMM=&TSTOPMM);
%GETJREF(DOGET=&JANREFS);
%YMHTKS;
DATA TOPLOT;
SET &DATA;
IF YYMM >= &DSTRTYM;
IF YYMM <= &DSTOPYM;
RUN;
%SAVLAST(DATA=TOPLOT,VAR=&VAR,OUT=&SUMFILE);
%RXSPRMS(DATA=TOPLOT,VAR=&VAR,
        AXISMIN=&VMIN,AXISMAX=&VMAX,STEP=&TICKSTEP);
%GETRTKS;
%TTLDATS(DATA=TOPLOT);
%GETREF(DATA=&REFFILE);
%MARKER(MARKER=&MARKER);
%TXTPRMS(HTTL=&HTTL,FTTLFT=&FTTLFT,
         HTXT=&HTXT,FTXT=&FTXT,HFT=&HFT);
%PLTILIN(DATA=TOPLOT,PLOTVVAR=&VAR,PLOTHVAR=YYMM,
         VAXISNO=1,HAXISNO=2);
%RAXIS(RAXISNO=1);
%YMHAXIS(YMHAXSNO=2);
%MEND;
```

YMPLOT1: Plot one variable vs. year-month. Options are explained in comments on MACRO statement.