

BREAKING THOSE MONTHLY GPLOT CHAINS

TERRY LESSMANN, MUTUAL OF OMAHA COMPANIES

PROBLEM: Your area maintains a rolling twelve-month graph. Each month the graph must be edited so that the latest twelve months appear in the correct order. Isn't there a way to have SAS® software do it automatically?

SOLUTION: Through the use of macros, SAS software can automate the execution of the graphics program. Three alternative solutions will be presented: two using SAS dates and one using non-SAS dates. Our objective will be to have SAS determine what the latest twelve months should be on the graph, and have them appear in the correct order. We will assume the data used in the graphics procedure will always be rolling twelve months. Sample data will be provided for each alternative.

ALTERNATIVE I:

The data set contains SAS dates (see sample 1). Using this data in a PROC GPLOT will produce a graph in the proper order, but the intervals do not represent months. An ORDER= option with BY MONTH in an AXIS statement must be used to produce correct monthly intervals. However, the ORDER= option requires monthly updates by the programmer. The goal is to have SAS determine the start and end date in the ORDER= option. The following code will show how macros can be used in conjunction with the SAS date variable REP_MO to accomplish this.

```

                                SAMPLE 1

DATA MO_DATA;
INPUT @01 CPU 3.
      @10 REP_MO DATE7.;

CARDS;
944      01 Sep 92
780      01 Oct 92
668      01 Nov 92
551      01 Dec 92
555      01 Jan 93
258      01 Feb 93
231      01 Mar 93
579      01 Apr 93
872      01 May 93
582      01 Jun 93
724      01 Jul 93
874      01 Aug 93
;
RUN;
```

```

PROC SORT DATA = MO_DATA;
  BY REP_MO;
RUN;

DATA NULL;
  SET MO_DATA END=EOF;
  IF _N_=1 THEN CALL SYMPUT ('FIRST', REP_MO);
  ELSE IF EOF THEN CALL SYMPUT('LAST', REP_MO);
RUN;

```

In the above code, the data set is sorted by the SAS date variable so the twelve months are properly ordered (in this example the data is already in sort order and would not need to be sorted). Next, a null data set is created where the value of REP_MO from the first record is put into the macro variable FIRST and the value of REP_MO from the last record is put into the macro variable LAST. The oldest (&FIRST) and the newest (&LAST) months are ready for use in the ORDER= option of the AXIS2 statement in the following PROC GPLOT.

```

PROC GPLOT      DATA = MO_DATA;
  AXIS1        LABEL = (F-SWISS)
               VALUE = (F-SWISS);
  AXIS2        LABEL = (F-SWISS)
               VALUE = (F-SWISS)
               ORDER = (&FIRST TO &LAST BY MONTH);

  PLOT CPU*REP_MO / VAXIS=AXIS1 HAXIS=AXIS2;
  SYMBOL1 I=JOIN C=CYAN V=—;
  FORMAT REP_MO MONYY5.;
  TITLE H=2 C=GREEN F=BRUSH 'CPU USAGE--MARKETING';
RUN;

```

&FIRST and &LAST resolve in the ORDER= option to produce the correct intervals for the graph. Note the FORMAT statement, which is necessary to convert the SAS dates from numbers into meaningful months and years on the output.

ALTERNATIVE II:

The data set in this alternative contains both a two-digit month and a two-digit year variable, rather than one SAS date field (see sample 2). In this scenario, the only way to get the months to appear in the proper order is to list each month in an ORDER= option. The ORDER= would have to be modified each month. The trick is to get the month and year variables into a SAS date variable. This is accomplished using the SAS software function MDY. The following code will create the necessary date variable.

SAMPLE 2

```
DATA MO_DATA;  
  INPUT CPU MONTH YEAR;  
  CARDS;
```

```
944    09 92  
780    10 92  
668    11 92  
551    12 92  
555    01 93  
258    02 93  
231    03 93  
579    04 93  
872    05 93  
582    06 93  
724    07 93  
874    08 93  
;  
RUN;
```

```
DATA MO_DATA (KEEP=CPU REP_MO);  
  SET MO_DATA;  
  REP_MO=MDY(MONTH, 1, YEAR);  
RUN;
```

The MDY function in the above code requires a month, day, and year value to create a SAS date. The data set contains MONTH and YEAR and a day value is hard coded in as "1" (which is arbitrary and could be any number that appears in every month of the year (1-28)). With day = 1, REP_MO will be a SAS date representing the first day of each month in the data set. With the SAS date created, the rest of the following program is similar to the first example.

```

PROC SORT;
  BY REP_MO;
RUN;

DATA _NULL_;
  SET MO_DATA END=EOF;
  IF _N_=1 THEN CALL SYMPUT('FIRST', REP_MO);
  ELSE IF EOF THEN CALL SYMPUT('LAST', REP_MO);
RUN;

GOPTIONS FTEXT=SWISS;

PROC GPLOT DATA=MO_DATA;
  AXIS1 ORDER=(&FIRST TO &LAST BY MONTH);

  PLOT CPU*REP_MO / HAXIS=AXIS1;
  SYMBOL1 I=JOIN C=CYAN V=;
  FORMAT REP_MO MONYY5.;
  TITLE H=2 C=GREEN F=BRUSH 'CPU USAGE--MARKETING';
RUN;

```

The macro variables are created in the null data step and resolved in the ORDER= option, releasing the programmer from monthly updates. Note that there is only one axis statement instead of two in Alternative I. This is possible by adding a GOPTIONS FTEXT=SWISS; statement, which makes SWISS the default font. The axis statements in Alternative I were used to define the font and set the order. The axis statement in Alternative II just defines the order.

ALTERNATIVE III:

The monthly data set does not have a SAS date variable or a month and year variable. Instead, the data set contains a variable MONTH with the values MTH1-MTH12 (see sample 3). Record 1 (MTH1) represents the oldest month while record 12 (MTH12) is the most recent. Next month, the CPU value in record two (258) will become the CPU value in record one. A PROC FORMAT, which must be updated monthly, is used to covert the values into meaningful months and years.

SAMPLE 1

```
DATA MO_DATA;  
  INPUT CPU MONTH $;  
CARDS;  
555 MTH1  
258 MTH2  
231 MTH3  
579 MTH4  
872 MTH5  
582 MTH6  
724 MTH7  
874 MTH8  
944 MTH9  
780 MTH10  
668 MTH11  
551 MTH12  
;  
RUN;
```

```
PROC FORMAT;  
  VALUE $CMON  
    'MTH1' = 'SEP92'  
    'MTH2' = 'OCT92'  
    'MTH3' = 'NOV92'  
    'MTH4' = 'DEC92'  
    'MTH5' = 'JAN93'  
    'MTH6' = 'FEB93'  
    'MTH7' = 'MAR93'  
    'MTH8' = 'APR93'  
    'MTH9' = 'MAY93'  
    'MTH10' = 'JUN93'  
    'MTH11' = 'JUL93'  
    'MTH12' = 'AUG93';
```

SAS macros can be used to build the format and eliminate the monthly maintenance.

```
%MACRO FMTMTH;
    DATA _NULL_;
    TODAY=DATE ();
    MO=MONTH (TODAY);
    YR=YEAR (TODAY)-1;

    %DO I=1 %TO 12;
        IF MO=13 THEN
            DO;
                MO=1;
                YR+1;
            END;
        CALL SYMPUT ("CMON&I", PUT (MDY (MO,1,YR), MONYY5));
        MO+1;
    %END;
RUN;
```

The previous code will create twelve macro variables &CMON1-&CMON12. The SAS software system date is used as a starting point in generating the twelve macro variables. Based on today's date, the variable MO is created from the MONTH function and contains the numeric value of the current month. The variable YR contains the current year minus one from the YEAR function. In the macro DO loop, the variable MO is incremented twelve times. SAS dates are generated using the MDY function, and these dates are formatted and put into the macro variable with the CALL SYMPUT statement. The DAY value in the MDY function is hard coded in as 1, but can be any value the programmer chooses from 1-28. When MO=13 (the start of a new year), MO is reassigned the value of 1 and the YR variable is incremented by one. In this example, the CALL SYMPUT created &CMON1 would evaluate to 'SEP92'. With the twelve macro variables built containing the proper month-year, the following PROC FORMAT is ready to build.

```
PROC FORMAT;
    VALUE $CMON
        %DO I=1 %TO 12;
            "MTH&I" = "&&CMON&I"
        %END;
    ;
RUN;
%MEND FMTMTH;
```

The above format is built using a %DO loop, which creates the range and values of the format. For example, when I=1, the range resolves to "MTH1". On the value side of the format, the macro processor resolves the && into &, scans the letters CMON (which it can't resolve), and resolves &I into 1. The macro processor always rescans an item resulting from a resolution, so it scans &CMON1 and resolves it ("SEP92" in our example). The loop continues through eleven more iterations. The loop builds the format and the value statement is closed with a semicolon after the loop is completed. The macro is ended and ready to call.

The macro FMTMTH is called, creating the format \$CMON. \$CMON is used in the following PROC GPLOT.

```
%FMTMTH;

PROC GPLOT DATA=MO_DATA;
  AXIS1 ORDER='MTH1' 'MTH2' 'MTH3' 'MTH4' 'MTH5' 'MTH6'
'MTH7'
           'MTH8' 'MTH9' 'MTH10' 'MTH11' 'MTH12';
  PLOT CPU*MONTH / CT=GREEN CA=GREEN
        HREF='MTH3' 'MTH6' 'MTH9' 'MTH12'
        LH=2 HAXIS=AXIS1;

  SYMBOL1 I=JOIN C=BLUE W=4;
  FORMAT MONTH $CMON.;
  TITLE H=2 C=GREEN F=BRUSH 'CPU USAGE--MARKETING';
  RUN;
```

AXIS1 uses an ORDER= option that remains constant each month (the CPU values and \$CMON change from month to month). By having SAS software create the format based on today's date, monthly maintenance is eliminated.

CONCLUSION: Regardless of the data format, SAS software should be used to automatically generate the correct X axis. Combining SAS software system features, functions, and macros will allow the programmer to eliminate monthly maintenance and break those monthly GPLOT chains.

Questions or comments regarding this paper should be directed to:

Terry Lessmann
Mutual of Omaha
Mutual of Omaha Plaza
Omaha, NE 68175
(402) 342-7600

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

*** The complete programs can be found on the following pages. ***

```

/*****
/**      ALTERNATIVE I      **/
/** PROGRAM TO PUT SAS DATES INTO **/
/** MACRO VARIABLES.      **/
*****/

GOPTIONS DEVICE=GDDMP CG BORDER;
OPTIONS MPRINT SYMBOLGEN;

DATA NO DATA;
  INPUT CPU 206 REP_MO DATE7.;
CARDS;
944 01SEP92
780 01OCT92
668 01NOV92
551 01DEC92
555 01JAN93
258 01FEB93
231 01MAR93
579 01APR93
872 01MAY93
582 01JUN93
724 01JUL93
874 01AUG93
;
RUN;

PROC SORT;
  BY REP_MO;
RUN;

DATA _NULL_;
  SET NO DATA END=EOF;
  IF _N_ = 1 THEN CALL SYMPUT('FIRST', REP_MO);
  ELSE IF EOF THEN CALL SYMPUT('LAST', REP_MO);
RUN;

PROC GPLOT DATA=NO DATA;
  AXIS1 LABEL=(F=SWISS)
  VALUE=(F=SWISS);
  AXIS2 LABEL=(F=SWISS)
  VALUE=(F=SWISS)
  ORDER=(&FIRST TO &LAST BY MONTH);

  PLOT CPU*REP_MO / VAXIS=AXIS1 HAXIS=AXIS2;
  SYMBOL1 I=JOIN C=CYAN V==;
  FORMAT REP_MO NONYYS.;
  TITLE H=2 C=GREEN F=BRUSH 'CPU USAGE -- MARKETING';
RUN;

```

```

/*****
/**      ALTERNATIVE II      **/
/** PROGRAM TO CREATE SAS DATES AND **/
/** PUT THEN INTO MACRO VARIABLES. **/
*****/

```

```

GOPTIONS DEVICE=GDDMPCG BORDER;
OPTIONS MPRINT SYMBOLGEN;

```

```

DATA NO_DATA;
  INPUT CPU MONTH YEAR;

```

```

CARDS;
944 09 92
780 10 92
668 11 92
551 12 92
555 01 93
258 02 93
231 03 93
579 04 93
872 05 93
582 06 93
724 07 93
874 08 93
;

```

```

RUN;

```

```

DATA NO_DATA (KEEP=CPU REP_MO);

```

```

  SET NO_DATA;
  REP_MO = MDY(MONTH,1,YEAR);
RUN;

```

```

PROC SORT;
  BY REP_MO;
RUN;

```

```

DATA NULL;
  SET NO_DATA END=EOF;
  IF _N_ = 1 THEN CALL SYMPUT('FIRST',REP_MO);
  ELSE IF EOF THEN CALL SYMPUT('LAST',REP_MO);
RUN;

```

```

GOPTIONS FTEXT=SWISS;

```

```

PROC Gplot DATA=NO_DATA;
  AXIS1 ORDER=(&FIRST TO &LAST BY MONTH);
  PLOT CPU*REP_MO / HAXIS=AXIS1;

```

```

  SYMBOL1 I=JOIN C=CYAN V=;
  FORMAT REP_MO MONYY5.;

```

```

TITLE H=2 C=GREEN F=BRUSH 'CPU USAGE -- MARKETING';
RUN;

```

```

/*****
**      ALTERNATIVE III      **
**  PROGRAM TO PUT MACRO VARIABLES  **
**  INTO PROC FORMAT.      **
*****/

GOPTIONS DEVICE=GDDMPG BORDER FTEXT=SWISS;
OPTIONS CENTER SYMBOLGEN NPRINT;

DATA NEW;
INPUT CPU MONTH $;
CARDS;
555 MTH1
258 MTH2
231 MTH3
579 MTH4
872 MTH5
582 MTH6
724 MTH7
874 MTH8
944 MTH9
780 MTH10
668 MTH11
551 MTH12
;

%MACRO FMTMTH;

DATA NULL;
TODAY=DATE();
* TODAY='01AUG93'D;
MO=MONTH(TODAY);
YR=YEAR(TODAY)-1;
XDO I = 1 XTO 12;
IF MO =13 THEN
DO;
MO= 1;
YR+ 1;
END;
CALL SYMPUT("CMON&I",PUT(MDY(MO,1,YR),MONYY5.));
MO+1;
XEND;
RUN;

PROC FORMAT;
VALUE $CMON
XDO I = 1 XTO 12;
"MTN&I" = "&2CMON&I"
XEND;
;
RUN;
%MEND FMTMTH;

%FMTMTH;

PROC GPLOT DATA=NEW;
  AXIS1 ORDER=
    'MTH1' 'MTH2' 'MTH3' 'MTH4' 'MTH5' 'MTH6' 'MTH7' 'MTH8'
    'MTH9' 'MTH10' 'MTH11' 'MTH12';
  PLOT CPU*MONTH / CT=GREEN CA=GREEN
    HREF='MTH3' 'MTH6' 'MTH9' 'MTH12' LH=2
    HAXIS=AXIS1;
  SYMBOL1 I=JOIN C=BLUE W=4;
  FORMAT MONTH $CMON.;

  TITLE H=2 C=GREEN F=BRUSH 'CPU USAGE -- MARKETING';
RUN;

```