Using Macros to Produce Multiple Time Series Graphs

Kimberly S. Blair, United States Bureau of the Census, Washington, D.C.

ABSTRACT

The United States Census Bureau's Time Series Analytic Repository (TSAR) System was written using SAS® software and UNIX with extensive use of macros. The graphical user interface for TSAR was coded using SAS/AF® software and Screen Control Language. TSAR is a time series repository and a generalized analysis system that houses valuable economic indicator data. TSAR currently runs on a Digital Equipment workstation and a Silicon Graphics Challenge XL (SGIXL) Posix server at the Census Bureau.

The focus of this paper is the graphing of TSAR time series using SAS/GRAPH® software and macros. With macro code, my program can produce any number of graphs from TSAR time series for any time period. This paper is intended for intermediate SAS users. For historical reference, a related paper entitled "Design and Development of the Census Bureau's Time Series Analytic Repository" was presented at the North East SAS Users Group 1996 Conference by Danielle Ringstrom.

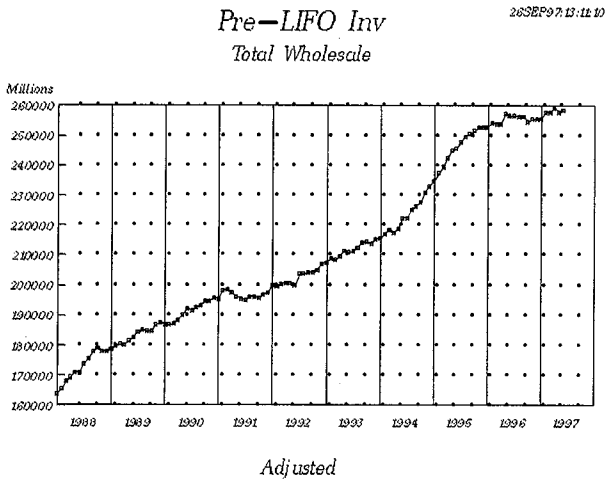
INTRODUCTION

The TSAR System is an integrated software and data system that stores and manipulates economic time series. Economic time series are vectors of data values where each data value represents some economic activity (such as sales) for a specific time period. This time period is typically months, quarters or years because economic surveys may be conducted monthly, quarterly or annually. In the TSAR System, each time series is stored as a separate data set containing just two fields, a data value and a date. For one economic survey, typically thousands of data sets are created to represent the various time series. One additional data set per survey exists containing all of the descriptive information needed to identify and understand each of the time series. This data set is referred to as the data descriptor file, and its format is shown in Appendix A. In TSAR, macros are used to perform code expansion and accomplish repetitive tasks such as cyclical updates, series arithmetic, seasonal adjustment, benchmarking and outputting series in various formats for printing.

TSAR also has a graphical user interface (GUI) system that was created using SAS/AF software and Screen Control Language. Assuming you have the access privileges for a certain survey, the TSAR GUI System gives you the ability to view secure time series or edit the data descriptor file. It also allows you to start batch executions to load new time series, start a processing cycle, run benchmarking, or release data to a public directory. All TSAR users, regardless of access privileges, may use the TSAR GUI System to view published time series, browse the data descriptor file, or access SAS/ASSIST® software to manipulate TSAR time series in their own work space.

PROBLEM

The main focus of this paper is the graphing of TSAR time series. Assume that you are given the task of producing similar graphs for multiple time series. Once you have the basic graph of one time series looking how you want, then you can add macros to expand the code to generate a graph for any TSAR time series. First, you use the GPLOT procedure to create a single graph that meets given specifications. Assume that these specifications come from an end user who works on a monthly survey. One of the problems you encounter involves the labels on the horizontal axis. The end user specifies that the graphs show the last ten years of data in a time series beginning with the December data month. For example, a graph for the March, 1997 data month began, at the origin, with December 1987 and shows all data points through March 1997. The labels along the horizontal axis denote the years (1987 1988 ... 1997). You define the December data months as major tick marks to obtain the horizontal reference lines. However, you find that it is impossible to center the year labels between the major tick marks. A very simple solution is to define the June data months as major tick marks, too. This allows you to easily center the year labels under June data months and in between two December data months as shown in the following graph.



TEMPORARY SOLUTION

Begin by writing the code to produce the graph by hard coding all values until the end user approves of the graph's appearance. Use a null DATA step to read the data descriptor file for textual information that the end user wants to see on the graphs. With this information, define macro variables, &title1, &title2, &title3 and &unit, for use in setting title and footnote definitions. Use the symbol statement to define the characteristics of the symbols which tell how the data are displayed.

In a PROC step, use PROC GPLOT to create the graph. Use a WHERE clause in the PROC GPLOT statement to subset the input time series. Plot the data value on the vertical axis by the date on the horizontal axis. Use the HREF= option to place reference lines perpendicular to the horizontal axis at each December data month. Define the horizontal axis to have an offset of zero so the first data point resided on the vertical axis. Use the ORDER= option to define the major tick mark values as the December and June data months. Use the VALUE= option to define the text for the major tick marks so that the December data months are blank and the June data months are the corresponding data years. Define the label for the vertical axis using the macro variable &unit which you read in from the data descriptor file. The unit is typically either "Millions" or "Ratios". Next, use the AUTOVREF option to draw reference lines at all major tick marks on the vertical axis, and the LVREF option to make the vertical reference lines dashed instead of solid. Finally, use the FRAME option to draw a frame around the axis area. When the graph looks pleasing to the user, you can make the code more versatile with macros.

ADDITION OF MACROS

With macros, your program can produce graphs of any TSAR time series, for any time period. Turn the code already written into two macros called %drv_grph and %graphit, and replace all the hard coding with macro variables. Given a single macro variable containing the current data year, you can calculate other values needed by %graphit. In %drv_grph, use a null DATA step to calculate the prior nine data years and store them in macro variables for use as the labels on the horizontal axis. Also, store the last two digits of the current data year and the prior ten data years into macro variables for use in the ORDER= option for the horizontal axis, in the WHERE clause in the PROC GPLOT statement, and in the HREF= option in the PLOT statement. In another null DATA step, read the input file containing the names of the series to be graphed. Store the series names and the total number of series to be graphed into macro variables. For each series to be graphed, pass the series name to %graphit for execution.

The program is still not complete. The end user wants the ability to change the ranges and increment for the vertical axis on the fly rather than use the default values that PROC GPLOT derives from the data. So, add three new fields to the input file that used to contain just the series names. Then the user can include in the input file, the three values (*n* TO *n* BY *increment*) needed for the ORDER= option. Add the MISSOVER option to the INFILE statement so that the three additional fields are optional. Store the three new fields into macro array variables (along with the series names). Next, before passing the name of a series to be graphed to %graphit, define the macro variable, &order, to contain the text for the ORDER= option. For each series to be graphed, you must check the macro array variable defined as the first of the three new fields. If it is blank, define &order to be a string containing nothing. Otherwise, define &order by stringing together the three new fields in the proper syntax. Add &order to the vertical axis definition in %graphit. With this additional feature, the end user can easily change the graph shown in Figure 1 (with the default order for the vertical axis) to the graph shown in Figure 2 (with a customized order for the verical axis). Note: The graphs presented in this paper do not appear exactly as they do in actuality.

CONCLUSION

The final version of the code in macro form is shown in Appendix B. An example input file is show in Appendix C. The end user creates the input file so that it contains the names of all the series to be reviewed in this graphical format. Then, the graph program is included in the cyclical processing for the appropriate survey. Every cycle, the end user receives new printouts of the graphs. As more data are loaded into the series, additional data points are automatically added to the graphs.

APPENDIX A - DATA DESCRIPTOR FILE FORMAT

DATA ITEM	DATA TYPE	LENGTH
DSNAME	CHARACTER	8
REFNAME	CHARACTER	12
TITLE1	CHARACTER	80
TITLE2	CHARACTER	80
TITLE3	CHARACTER	80
TITLE4	CHARACTER	80
DESC	CHARACTER	200
SIC	CHARACTER	9
GEO	CHARACTER	15
UNIT	CHARACTER	2
INDEX	CHARACTER	4
SEASADJ	CHARACTER	1
TRADADJ	CHARACTER	1
HOLIDAY	CHARACTER	1
BENCHADJ	CHARACTER	1
FREQ	NUMERIC	4
NOTES	CHARACTER	160
PUBLSER	CHARACTER	1
PERMSER	CHARACTER	1
DERVSER	CHARACTER	1
STCKFLOW	CHARACTER	1
USRFLAG1	CHARACTER	1
USRFLAG2	CHARACTER	1
USRFLAG3	CHARACTER	1
USRFLAG4	CHARACTER	1
USRFLAG5	CHARACTER	1
USRFLAG6	CHARACTER	1
START	DATE	8
END	DATE	8

DATA DESCRIPTOR FILE ITEMS USED IN %GRAPHIT:

DSNAME - This data item is the data series name. It contains an eight (8) character name which is the name of the SAS data set which is used to store the actual time series vector. This data item can be composed of both numerics and characters; it usually is a combination of both types of characters. In addition, this data item is used as the index or link between the time series data file and the data descriptor file.

TITLE1 - This is a 80 character data item that contains the title of the time series. This title is purely text and is designed to be used in printouts to describe the time series. The owner of the time series provides the title.

TITLE2 - Additional title information, also a maximum of 80 characters.

TITLE3 - Additional title information, also a maximum of 80 characters.

UNIT - This data item contains the unit-of-measure for the data value for the time series. It defines what is being measured (quantified) by the series. The allowable integer data values are as follows:

1 - Dollars ($)
2 - Hundreds of Dollars ($100)
3 - Thousands of Dollars ($1,000)
4 - Millions of Dollars ($1,000,000)
5 - Combined Adjustment Factors (decimal)
6 - Seasonal Adjustment Factors (decimal)
7 - Trading Day Adjustment Factors (decimal)
8 - Holiday Adjustment Factors (decimal)
9 - Benchmark Adjustment Factors (decimal)
10 - Other Adjustment Factors
19 - Percent Change
11 - Ratios (decimal)
12 - Quantity (number) of outputs (integer)
13 - Quantity in thousands of units
14 - Quantity in millions of units
15 - Indexed data
16 - Standard Errors
17 - Coefficients of Variation
18 - Other

APPENDIX B - CODE

```
/*********************************************************************
MACRO:  drv_grph

 This macro is the driver program for the graphing of TSAR
 time series. This program reads processing control input
 files for graphing and graphs the specified series in the
 appropriate format.

  Required Parameter:
        grphpcif - Graphing Processing Control Input File
            ("/tsar-data/mwts/control/mwts-graph-series")
  Required Macro Variable (defined previously):
        datayear - Data Year (1997)

*********************************************************************/
%include "/tsar-soft/programs/graphit.mac";

%macro drv_grph (grphpcif) ;

 /*
  This DATA step defines macros variables gyy0 - gyy10
  and gyr1 - gyr9 which are used in defining the horizontal
  axis and reference lines for all graphs.
 */

 data _null_;
  call symput('gyy0',substr("&datayear",3,2));
  do j=1 to 9;
   putvar=&datayear-j;
   charyear=put(putvar,4.);
   charj=put(j,4.);
   trimj=trim(left(charj));
   call symput('gyy'||trimj,substr(charyear,3,2));
   call symput('gyr'||trimj,charyear);
  end;
  j=10;
  putvar=&datayear-j;
  charyear=put(putvar,4.);
  charj=put(j,4.);
  trimj=trim(left(charj));
  call symput('gyy'||trimj,substr(charyear,3,2));
 run;

 /*
  This DATA step reads the graphing processing control
  input file.  It defines macro variables for each of the series
  to be graphed and for the number of series to be graphed.
 */

 data _null_;
  infile &grphpcif end=lastone missover;
  input series $ gfrom $ gto $ gby $;
  trimn=trim(left(put(_n_,4.)));
  call symput('series'||trimn,series);
  call symput('gfrom'||trimn,trim(left(gfrom)));
  call symput('gto'||trimn,trim(left(gto)));
  call symput('gby'||trimn,trim(left(gby)));
  if lastone then call symput('numsers',trimn);
 run;

 /***  Run the graph macro once for each series.  ***/

 %do I = 1 %to &numsers;
  %if "&&gfrom&I" = " " %then
    %let order=%STR();
  %else
     %let order=%STR(order=(&&gfrom&I to &&gto&I by
                                    &&gby&I));
  %graphit(&&series&I);
 %end;

%mend drv_grph;
/*****************  End of MACRO drv_grph  *****************/
```

```
/*********************************************************************
MACRO:  graphit

 This macro will create and print a graph for a single series.

 The graph will show the last ten years in the series and will
 always start with the December data month.

  Required Parameter:
        series - Series Name (S0BZ5000)

  Required Macro Variables (defined previously):
        survey - Survey Name (mwts)
        datayear - Data Year (1997)
        prdevice - Print Device (LJ5PS)

  Required Librefs:
        &survey - Directory (/tsar-data/&survey/series)
        library - Formats Library (/tsar-soft/formats)

*********************************************************************/
%macro graphit(series);

 filename fileto "/tsar-data/&survey/listings/&survey..graph";

 goptions reset=all reset=global;
 goptions device=&prdevice gsfname=fileto;
 goptions gsfmode=append rotate=landscape;
 goptions ftext=zapfi htext=1;

 data _null_;

  set &survey..descript
   (where=(upcase(dsname)=upcase("&series")));

  call symput('title1',trim(left(title1)));
  call symput('title2',trim(left(title2)));
  call symput('title3',trim(left(title3)));
  call symput('unit',trim(left(put(input(unit,2.),units.))));

  gdt=datetime();
  call symput('gdt',put(gdt,datetime.));

 run;

 symbol1 i=join v=square h=0.4;

 axis1 offset=(0) minor=none label=none major=none

     order="01dec&gyy10"d "01jun&gyy9"d
           "01dec&gyy9"d   "01jun&gyy8"d
           "01dec&gyy8"d   "01jun&gyy7"d
           "01dec&gyy7"d   "01jun&gyy6"d
           "01dec&gyy6"d   "01jun&gyy5"d
           "01dec&gyy5"d   "01jun&gyy4"d
           "01dec&gyy4"d   "01jun&gyy3"d
           "01dec&gyy3"d   "01jun&gyy2"d
           "01dec&gyy2"d   "01jun&gyy1"d
           "01dec&gyy1"d   "01jun&gyy0"d
           "01dec&gyy0"d

     value=(t=1   "" t=2   "&gyr9"
            t=3   "" t=4   "&gyr8"
            t=5   "" t=6   "&gyr7"
            t=7   "" t=8   "&gyr6"
            t=9   "" t=10 "&gyr5"
            t=11 "" t=12 "&gyr4"
            t=13 "" t=14 "&gyr3"
            t=15 "" t=16 "&gyr2"
            t=17 "" t=18 "&gyr1"
            t=19 "" t=20 "&datayear"
            t=21 "");

 title1 h=2 f=zapfi "&title2" h=1 j=r "&gdt";
 title2 h=1.5 f=zapfi "&title1";

 footnote1 h=1.5 f=zapfi "&title3";

 proc gplot data=&survey..&series
  (where=(date>="01dec&gyy10"d));

  plot &series*date
   / href="01dec&gyy9"d "01dec&gyy8"d "01dec&gyy7"d
          "01dec&gyy6"d "01dec&gyy5"d "01dec&gyy4"d
          "01dec&gyy3"d "01dec&gyy2"d "01dec&gyy1"d
     haxis=axis1 vaxis=axis2 autovref lvref=35 frame;

 run;
 quit;

%mend graphit;
/******************  End of MACRO graphit  ******************/
```

APPENDIX C - EXAMPLE INPUT FILE

```
S0BZ5000
S0BZ5002
P0BZ5000
P0BZ5002
RPBZ5000 1.1 1.5 .05
RPBZ5002 1.3 2.0 .05
```

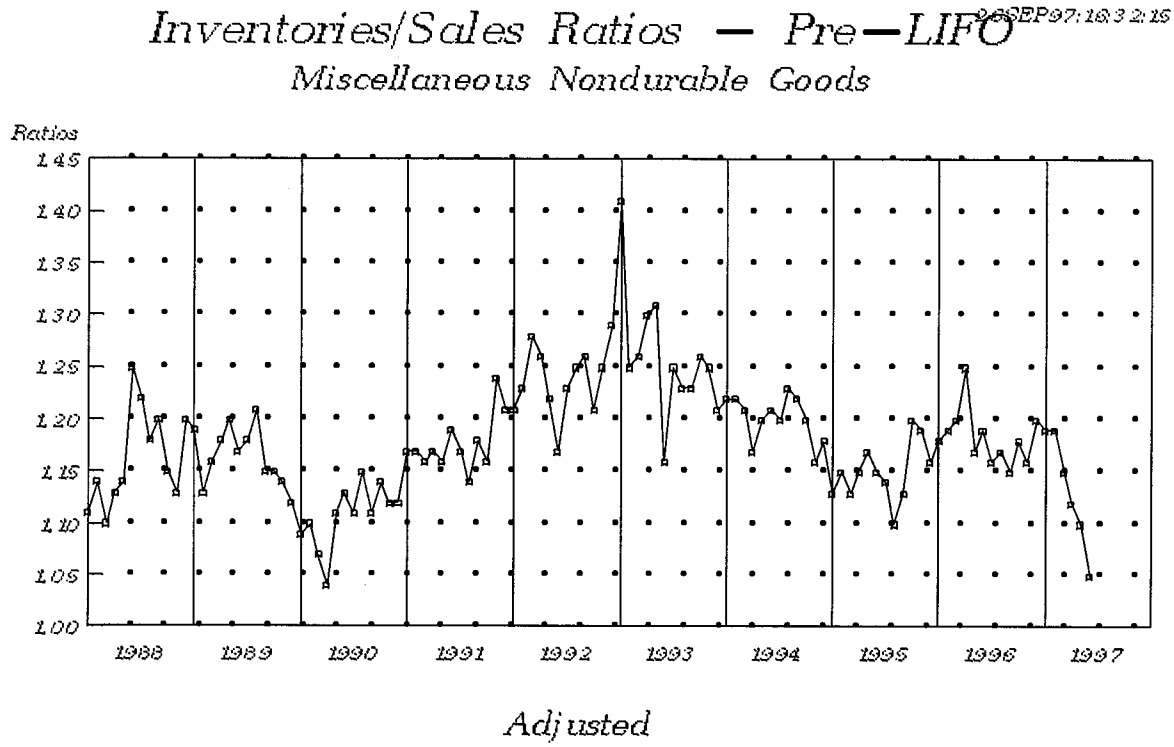


Figure 1

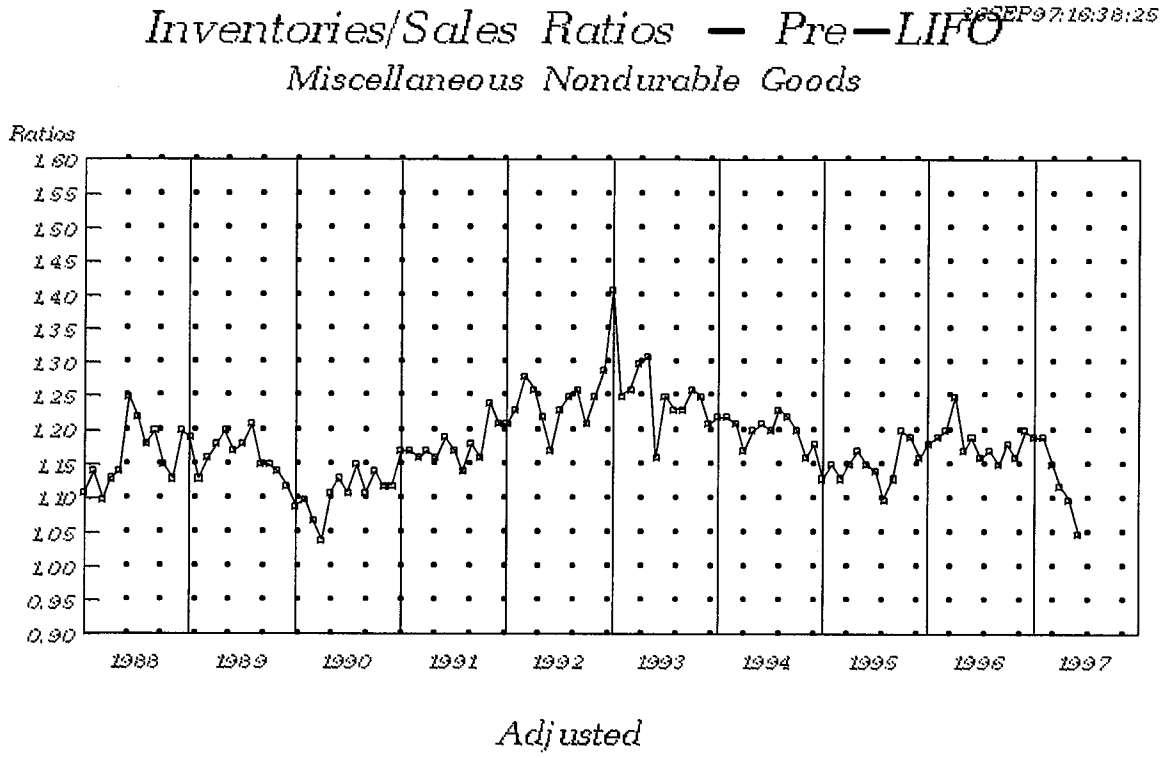


Figure 2