Creating Scatterplot Matrices Using SAS/GRAPH® Software

Robert A. Vierkant, Marshfield Medical Research Foundation, Marshfield WI

ABSTRACT

The Pearson product-moment correlation coefficient measures the strength of the linear relationship between two variables. Correlation matrices are a good way to descriptively assess such relationships, but graphical approaches are often needed to tell the entire story. A scatterplot matrix is a graphical display of the bivariate relationships between a number of quantitative variables that is structured in a similar way as a correlation matrix. This paper presents a SAS® macro that produces scatterplot matrices using SAS/GRAPH software. It is intended for all SAS users, regardless of skill level. The macro is similar to another macro previously published, but with certain enhancements.

INTRODUCTION

It is often necessary in quantitative analyses to assess bivariate relationships between two or more variables. The most common way of measuring these relationships is with the Pearson product-moment correlation coefficient. Correlation matrices of quantitative variables can be produced with SAS procedure CORR (1990). These matrices are a good way to descriptively assess such relationships, but graphical approaches are often needed to find hidden associations or problems. A scatterplot matrix is a graphical display of the bivariate associations between a number of quantitative variables that has the same general layout as a correlation matrix. This paper presents a SAS macro that produces scatterplot matrices using SAS\GRAPH templates and procedures (1991a). An example is provided that shows output produced by the macro.

CORRELATION MATRICES

The Pearson product-moment correlation coefficient r is a standardized measure of the linear relationship between two quantitative variables. Values of the Pearson coefficient range from -1 to 1. A value of r near or equal to zero implies little or no linear relationship between variables x and y. The closer r is to -1 or 1, the stronger the linear relationship. Positive values of r imply that y increases as x increases; negative values imply that y decreases as x increases.

A common practice in the initial stages of a project is the generation of a correlation matrix of all quantitative variables as an exploratory way of examining the data. In such a matrix, a correlation coefficient is calculated between each pair of k variables, and these correlations are collectively represented in a square k x k matrix. Table 1 shows a correlation matrix of three hypothetical variables VAR1, VAR2, and VAR3 created with SAS procedure CORR. The entry in each cell is the correlation coefficient corresponding to the two variables. For example, the correlation coefficient between VAR1 and VAR2 is 0.737. The values on the diagonal are all 1 because a variable is always perfectly correlated with itself, and the matrix is symmetric because the correlation between VAR1 and VAR2 is the same as the correlation between VAR2 and VAR1.

Table 1: Correlation Matrix of Hypothetical Variables VAR1, VAR2, and VAR3

Correlation Analysis

3 'VAR' Variables: VAR1 VAR2 VAR3
Pearson Correlation Coefficients

	VAR1	VAR2	VAR3
VAR1	1.000	0.737	-0.014
VAR2	0.737	1.000	-0.072
VAR3	-0.014	-0.072	1.000

A correlation matrix is often a good way to descriptively assess the relationship between two variables, but sometimes this information is not enough. Figure 1 shows a perfect quadratic relationship between two variables that produces a correlation coefficient of zero. In contrast, Figure 2 shows a relationship where the correlation between two variables is 0.72. However, this coefficient is drastically affected by the one outlier in the data. Removal of that observation produces a new

Figure 1

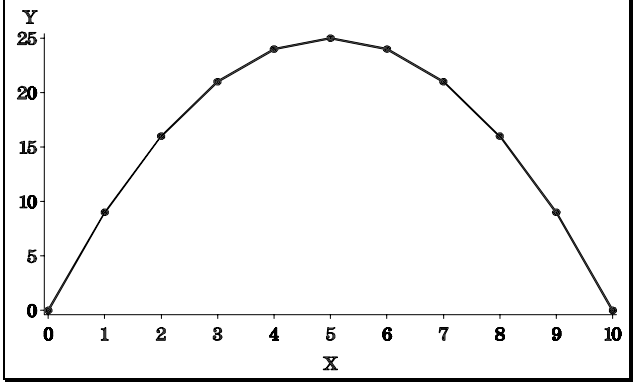
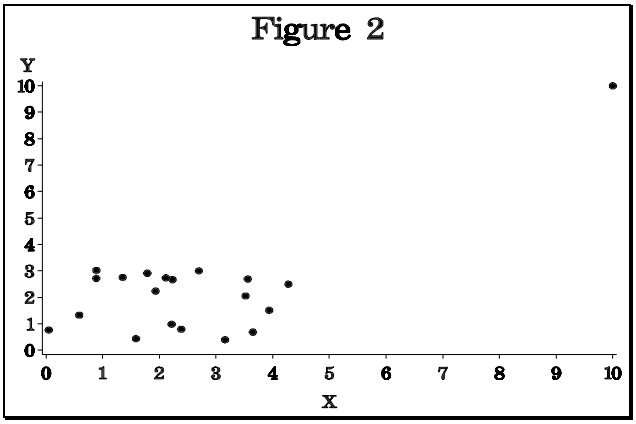


Figure 2

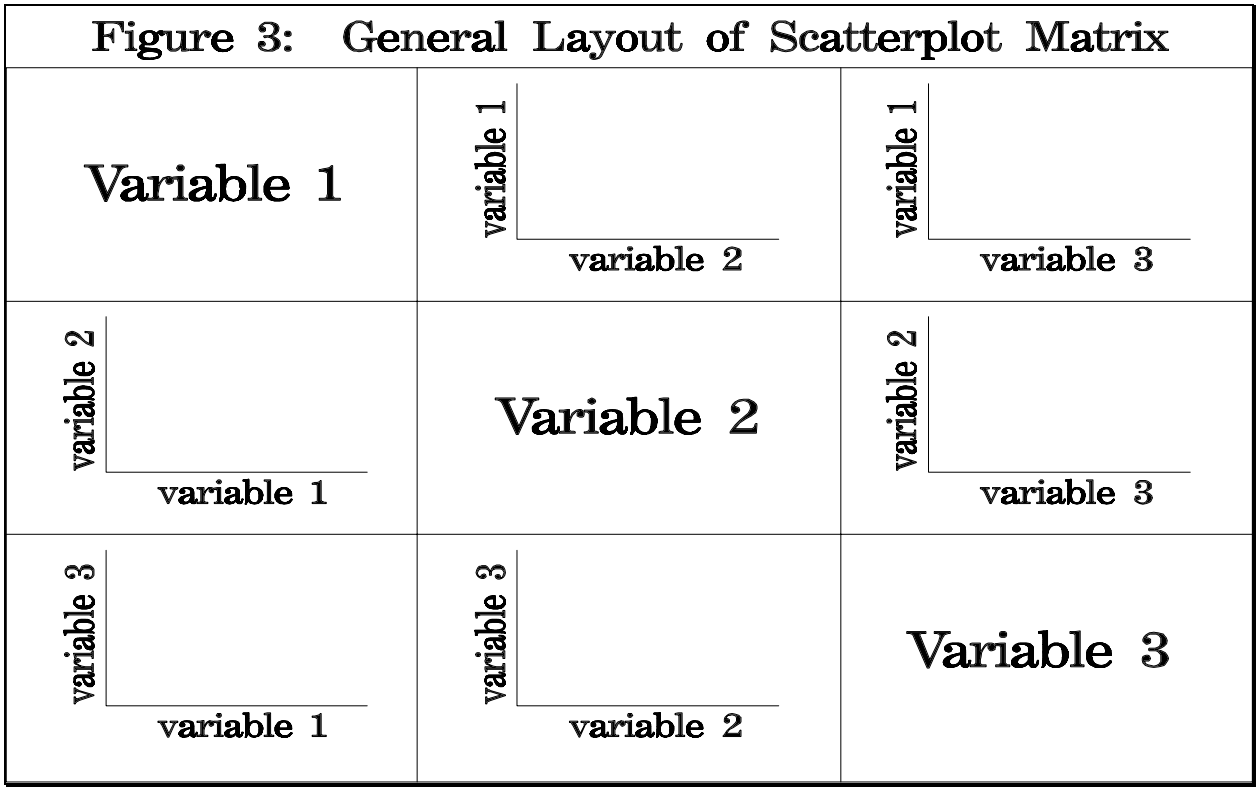


correlation of 0.01. These two examples indicate that it is important to graphically consider the relationship between two variables as well as descriptively. An efficient way of doing this is with a scatterplot matrix.

SCATTERPLOT MATRICES

The layout of a scatterplot matrix is similar to that of a correlation matrix. A scatterplot is produced for each pair of k variables, and these plots are collectively represented in a square k x k matrix. Figure 3 shows a general layout of a scatterplot matrix for three variables. Names and mean values of the variables are often placed on the marginals or in the diagonal elements of the matrix. Each off-diagonal cell corresponds to a scatterplot of two of the variables and has the following format: the vertical axis of the plot is the variable named in diagonal element falling in the same row as the plot, and the horizontal axis is the variable named in the diagonal element falling in the same column as the plot. For example, the plot in row 1, column 2 of Figure 3 is that of variable 1 (vertical axis) and variable 2 (horizontal axis). Notice that the upper right triangle of the matrix is similar to the lower left triangle, except that the horizontal and vertical axes are transposed.

Figure 3: General Layout of Scatterplot Matrix



SAS MACRO

The SAS macro PLOTMAT (appendix) contains code to generate a scatterplot matrix of up to ten variables using SAS/GRAPH templates and procedures GPLOT, GSLIDE, and GREPLAY. The names and mean values of each variable are placed in the diagonal elements of the matrix, and the scatterplots are placed in the off-diagonal elements. Options are provided that 1) allow the user to include the correlation between the two variables along with each scatterplot and 2) allow the user to specify an overall title for the matrix. Keyword parameters are required to specify the data set (DS), the number of variables to be displayed in the scatterplot matrix (NUMVARS), and the name of each variable to be included in the matrix (VAR1--VAR10). Ten parameters are required for variable names, and each is defaulted to a null value. If the number of variables to be displayed in the scatterplot is less than ten, then the user needs only to specify variable names for the first few parameters and leave the remaining parameters at the null value. Keyword parameters also specify the optional title for the matrix (the default is no title), and the option to include the correlation coefficient with each scatterplot (the default is no correlations).

This macro produces output that is similar to that produced in Friendly's macro SCATMAT (SAS Institute Inc., 1991b). Some enhancements included in the macro PLOTMAT that are not in SCATMAT are the options to print both the individual correlations with each plot and an overall title for the matrix.

The macro PLOTMAT uses the call symput command to read variable means and correlations into macro parameters. An annotate data set is used to display correlations in each plot instead of an individual title. This allows all plot statements to fall within the same GPLOT procedure. A template is created based on the number of variables in the matrix and whether an overall title for the graph is specified. Finally, the display option is turned off when each individual plot is created, but is turned back on when the final scatterplot matrix is to be displayed.

EXAMPLE

Table 2 contains a correlation matrix for data set NAION that includes 51 cases in a case-control study of nonarteritic anterior ischemic optic neuropathy, an eye disorder resulting from increased blood flow resistance in the optic nerve head (Jacobson et al, 1997). Variables included in the correlation matrix are height (HEIGHT), weight (WEIGHT), hematocrit level (HEMAT), white blood cell count (WBC), cholesterol level (CHOL), and creatinine level (CREAT).

Figure 4 shows the scatterplot matrix for these variables created by macro PLOTMAT using the following macro call:

```
%plotmat(ds=naion, numvars=6, var1=height,
var2=weight, var3=hemat, var4=wbc, var5=chol,
var6=creat, corr=Y, title='Figure 4: Scatterplot Matrix
of Continuous Variables in Data Set NAION')
```

Notice the one extreme value of creatinine depicted in Figure 4. This value merits additional attention and would not have been detected with a correlation matrix.

Table 2: Correlation Matrix of Variables in Data Set NAION

Correlation Analysis
6 'VAR' Variables: HEIGHT WEIGHT HEMAT WBC CHOL CREAT

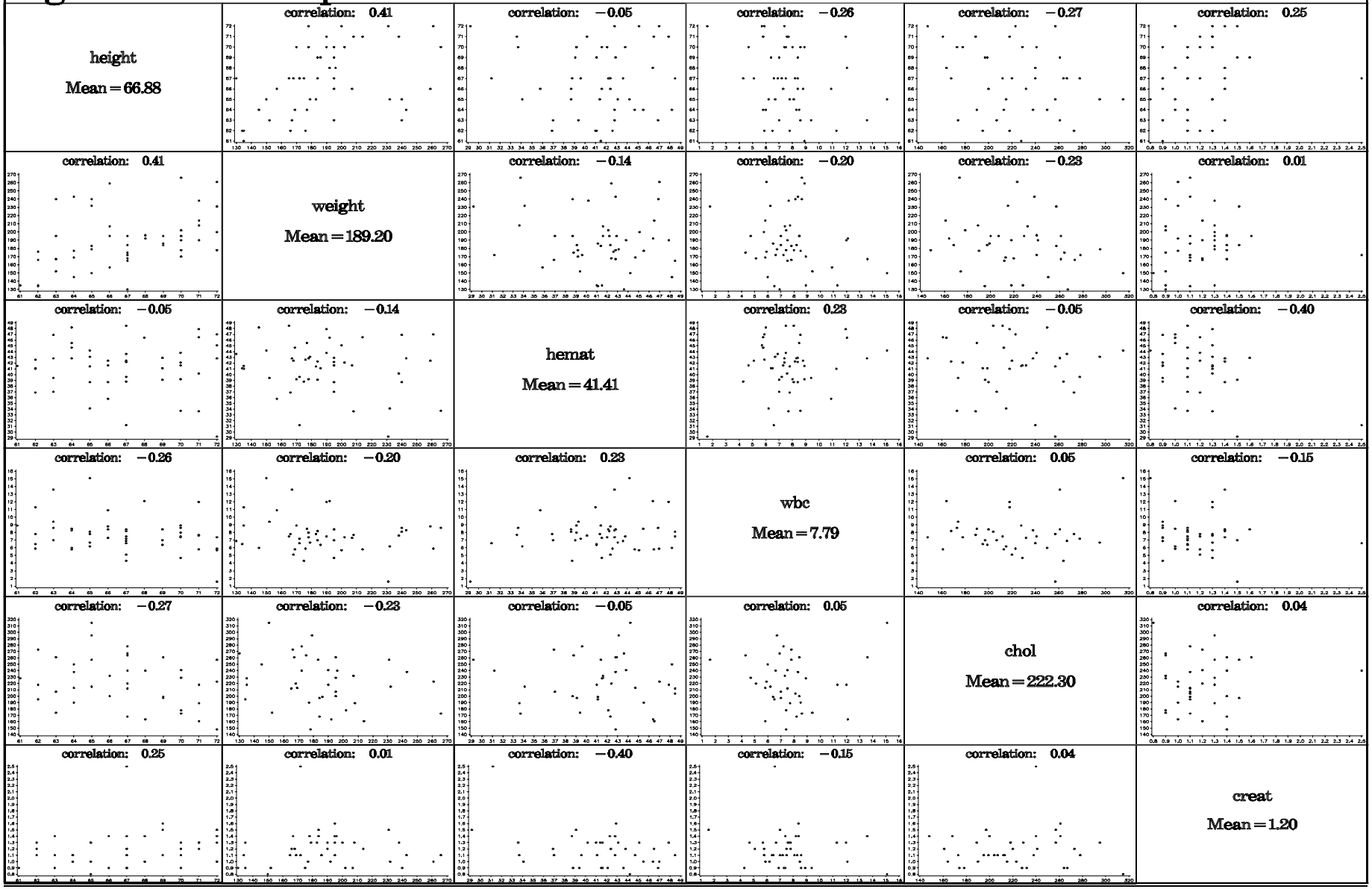
Pearson Correlation Coefficients

	HEIGHT	WEIGHT	HEMAT	WBC	CHOL	CREAT
HEIGHT	1.000	0.413	-0.050	-0.263	-0.266	0.246
WEIGHT	0.413	1.000	-0.141	-0.204	-0.231	0.013
HEMAT	-0.050	-0.141	1.000	0.232	-0.054	-0.398
WBC	-0.263	-0.204	0.232	1.000	0.045	-0.151
CHOL	-0.266	-0.231	-0.054	0.045	1.000	0.040
CREAT	0.246	0.013	-0.398	-0.151	0.040	1.000

CONCLUSION

Scatterplot matrices are valuable tools in assessing bivariate relationships between continuous variables, and can often detect associations or problems that simple correlation matrices cannot. SAS macro PLOTMAT presents an easy and effective way to create scatterplot matrices.

Figure 4: Scatterplot Matrix of Continuous Variables in Data Set NAION



REFERENCES

Jacobson DM, Vierkant RA, and Belongia EA (1997), "Nonarteritic Anterior Ischemic Optic Neuropathy: A Case-Control Study of Potential Risk Factors," *Archives of Ophthalmology*, 115, 1403-1407.

SAS Institute Inc. (1991a), *SAS/GRAPH Software: Usage, Version 6, First Edition*, Cary, NC: SAS Institute, Inc.

SAS Institute Inc. (1991b), *SAS System for Statistical Graphics, First Edition*, Cary, NC: SAS Institute Inc, pp. 576-581.

SAS Institute Inc. (1990), *SAS Procedures Guide, Version 6, Third Edition*, Cary, NC: SAS Institute, Inc.

CONTACT INFORMATION

Robert A. Vierkant, MAS
Marshfield Medical Research Foundation
1000 North Oak Avenue, ML2
Marshfield, WI 54449
(715) 389-3536
Email: vierkanr@mfldclin.edu

APPENDIX: SAS MACRO PLOTMAT

```
**************************************;
**         SAS MACRO PLOTMAT        **;
**     PARAMETERS ARE AS FOLLOWS    **;
**                                  **;
** 1) ds=data set                   **;
** 2) numvars=number of variables to **;
**       be in matrix (2 to 10)     **;
** 3) var1--var10=names of variables **;
**      in the matrix.  If have less **;
**      than 10, then leave values  **;
**      of remaining variables null **;
** 4) title=title of scatterplot    **;
**      matrix.  Default is null    **;
** 5) corr=option to print correla- **;
**      tions with scatterplots.    **;
**      Options are YES or Y, and   **;
**      NO or N                     **;
**                                  **;
**************************************;


%macro plotmat(ds=,numvars=,var1=,var2=,
               var3=,var4=,var5=,var6=,
               var7=,var8=,var9=,var10=,
               title=,corr=N);

****generate means and correlations;
proc corr data=&ds out=tmp noprint;
   var &var1 &var2 &var3 &var4 &var5
       &var6 &var7 &var8 &var9 &var10;
run;
data tmp; set tmp;

  ****create macro variables for all means;
  if _TYPE_='MEAN' then do;
    %do i=1 %to &numvars;
      call symput("m&i",trim(left
                 (put(&&var&i,10.2))));
    %end;
  end;

  ****create macro variables for all
      correlations;
  %do i=1 %to &numvars;
    if _NAME_=upcase("&&var&i") then do;
    %do j=1 %to &numvars;
      %let k=%eval((&i-1)*&numvars+&j);
      call symput("c&k",trim(left
                 (put(&&var&j,10.2))));
    %end;
    end;
  %end;
run;

****create annotate data sets used to place
    correlation on scatterplot;
%if %upcase(&corr)=Y or %upcase(&corr)=YES
%then %do;
  %do i=1 %to &numvars;
    %do j=1 %to &numvars;
      %let k=%eval((&i-1)*&numvars+&j);
      data annot&k; function='label';
         xsys='3'; ysys='3'; y=96; x=50;
         hsys='3'; size=8; style='centx';
         text="correlation:  &&c&k"; output;
      run;
    %end;
  %end;
%end;

****graphic options;
goptions reset=global device=win nodisplay
         gunit=pct border rotate=landscape;

****scatterplots for the off-diagonal;
symbol1 h=2 value=dot;
axis1 label=none minor=none
      value=(h=3 f=simplex);
proc gplot data=&ds gout=plotmat;

   ****title if correlation=yes is specified;
   title;
   %if %upcase(&corr)=Y or %upcase(&corr)=YES
   %then %do;
      title h=8 f=centx ' ';
   %end;
   %do i=1 %to &numvars;
     %do j=1 %to &numvars;
       %let k=%eval((&i-1)*&numvars+&j);
       plot &&var&i*&&var&j / vaxis=axis1
                              haxis=axis1
                              name="g&i._&j"
       %if %upcase(&corr)=Y
       or %upcase(&corr)=YES %then %do;
          anno=annot&k
       %end;
       ;
     %end;
   %end;
run; quit;


****variable names and means for
    diagonal elements;
%do l=1 %to &numvars;
  proc gslide gout=plotmat name="m&l";
    title1 h=10 f=centx lspace=30
           "&&var&l";
    title2 h=10 f=centx lspace=8
           "Mean=&&m&l";
  run; quit;
%end;
```

```
****graph for the title;
proc gslide gout=plotmat name='title';
   title h=4 f=centx &title;
run; quit;

****create template;
goptions display;
proc greplay igout=plotmat tc=tempcat nofs;

   ****assign the x and y coordinates
       within the template for each graph
       that is to be represented;
   tdef m&numvars
   %let num=%eval(&numvars-1);
   %if &title= %then %let totpct=100;
   %else %let totpct=95;

   %do i=0 %to &num;
      %do j=1 %to &numvars;
         %let t=%eval(&i*&numvars+&j);
         %let lx=%eval(100*(&j-1)
                  /&numvars);
         %let ly=%eval(&totpct*
                  (&numvars-&i-1)/&numvars);
         %let uy=%eval(&totpct*
                  (&numvars-&i)/&numvars);
         %let rx=%eval(100*&j/&numvars);

         %let x=&t. / llx=&lx. lly=&ly.
                ulx=&lx. uly=&uy. urx=&rx.
                ury=&uy. lrx=&rx. lry=&ly;
         &x
      %end;
   %end;
   %if title^= %then %do;
      %let t=%eval(&t+1);
      %let x=&t. / llx=0 lly=0 ulx=0 uly=100
             urx=100 ury=100 lrx=100 lry=0;
      &x
   %end;
   ;
   template m&numvars;

   ****place graphs in the boxes created
       for template defined above;
   treplay
   %do i=1 %to &numvars;
      %do j=1 %to &numvars;
         %let t=%eval((&i-1)*&numvars+&j;
         &t:
         %if &i=&j %then %do;
            m&i
         %end;
         %else %do;
            g&i._&j
         %end;
      %end;
   %end;
   %if title^= %then %do;
      %let t=%eval(&t+1);
      &t:title
   %end;
   ;
run; quit;

****delete graphs from temporary catalogs;
proc catalog c=plotmat kill; run; quit;
%mend plotmat;
```