FRAME IT! FRAME GENERATOR MACROS

Akos Felsovalyi, J.C. Penney Company, Inc.
Erzsebet Felsovalyi, Bloomfield College

ABSTRACT

The paper introduces a SAS macro that builds a frame around the graph by repeating a motif along the edges of the graph. The frame motif can be chosen from 101 predefined motifs available for the user! The user can also define his or her own favorite motif by setting up a simple annotate-like data set. Another macro enables the user to select sections of the frame and to move them to a new location. Utilizing the latter macro, nested multiple frames can be created, or smaller parts of the graph can be framed.

INTRODUCTION

The earlier SAS versions had not offered any automatic way to draw a frame around the graph. SAS Institute had recognized this lack and developed a new graphic option: BORDER. See page 4 of (3). It drew a simple rectangle bordering the display. However, the user community wanted more sophisticated and fancier borders. To fulfill that need, Kalinoski (1) designed some macros that were able to produce fancy borders. In his solution, there were four macros to produce the four edges of the frame using the .D drawing option in the NOTE statement to repeat the basic pattern. By the dawn of Version 5, his solution became a little inconvenient and now, the new annotate facility furnishes us with simpler and more powerful tools.

1. MACRO %FRAME, THE FRAME GENERATOR

Macro %FRAME is a simple, versatile frame generator running under SAS Version 5. It is able to produce almost any kind of frame, where a basic pattern makes up the frame by repeating itself. See Figure 6 for some examples. Macro %FRAME takes a motif and builds an annotate data set of the frame by repeating the motif along the four edges of the display area. When that data set is specified as the annotate option of a SAS/GRAPH procedure, the aimed frame is drawn.

The shaded part of Figure 1 shows the area occupied by the frame. The four corner areas may or may not belong to the frame depending upon the user. The user first has to define the motif. The motif is always one or more polygons defined by the 'motif data set'. Details about that data set are in paragraph 2. A simple motif is shown in Figure 2. That motif consists of two lines determined by three points: (0,1), (1,0) and (2,1). The macro encloses the motif into a rectangle and repeats it as building blocks as seen in Figure 3. The circumscribed rectangle is determined by the minimum and the maximum X and Y coordinates of the motif. The number of times the macro repeats the motif vertically (VNM) and horizontally (HNM) is controlled by the user. The user also controls the thickness of the frame (HSPACE and VSPACE values in Figure 1). Some frames require corner segments. The corner can be 'right angle'-type or 'diagonal'-type, or can be missing. Figure 4 pictures the same frame with the two types of corners. The type of corner is determined in the 'motif data set', see paragraph 2.

The output of macro %FRAME is an annotate data set. Its variables are the usual annotate variables: XSYS, YSYS, FUNCTION, X, Y, COLOR, SIZE, and LINE.
A simple motif and its creation

DATA A;
FUNCTION 'MOVE': X=0; Y=1; OUTPUT;
FUNCTION 'DRAW': X=1; Y=0; OUTPUT;
FUNCTION 'DRAW': X=2; Y=1; OUTPUT;

Figure 2

Frame based on motif of Figure 2

Observe
the building blocks

Figure 3

and one additional variable: EDGE. Variable EDGE is a character 3 variable and it is the name of the section to which the observation belongs. Its value can be T, L, B, R, ULC, URC, LRC, LLC marking the top, left, bottom, right, upper left corner, upper right corner, lower right corner and lower left corner of the frame respectively. By having this variable in the data set, we will be able to select specified parts of the frame. See macro %FRAMESL in paragraph 5. XSYS and YSYS have value of 3 for the 'absolute screen %' coordinate system, in other words all coordinates in the output data set are expressed in percentages between 0 and 100.

Parameters:

MOTIF= defines the motif of the frame. It can be an integer number or the name of a 'motif data set'. If it is a number, then a pre-defined motif is taken. See paragraph 3 about the predefined motifs. If it is a data set name, a user-defined data set supplies the motif. See paragraph 2 about setting up the 'motif data set'.

OUT= determines the name of the annotate data set created by this macro. This data set has to be the annotate option of a SAS/GGRAPH procedure in order to draw the frame.

VSPACE= determines the thickness of the top and the bottom of the frame. See Figure 1. It has to be a positive number not greater than 50, because the 'absolute screen %' coordinate system is used. Default is 0.

HSPACE= determines the thickness of the left and the right sides of the frame. See VSPACE above.

VNM= number of times the motif is repeated vertically. VNM has to be a positive integer number. Default value is 1.

HNM= number of times the motif is repeated horizontally. See VNM above.

ROTATE= defines the angle in degree by which the motif has to be rotated about the lower left corner of its circumscribing rectangle. This rotation takes place before the macro builds the frame. Default is no rotation: ROTATE=0.

Let us draw a frame based on the motif of Figure 2. The solution is the following:

DATA A:
FUNCTION 'MOVE': X=0; Y=1; OUTPUT;
FUNCTION 'DRAW': X=1; Y=0; OUTPUT;
FUNCTION 'DRAW': X=2; Y=1; OUTPUT;
%FRAME(MOTIF=A, HSPACE=10, VSPACE=15, VNM=4, HNM=2, ROTATE=0)
PROC GSLIDE ANNOTATE=B;

In this example the motif is designed by the 'motif data set' A. Data set B is the annotate data set produced by the macro and used as the annotate data set by procedure GSLIDE.

Frame with 'right angle' corner

Frame with 'diagonal' corner

Figure 4
2. THE 'MOTIF DATA SET'

The motif is always one or more polygons, even if arcs are part of the motif, because arcs can be rectified by polygons. E.g. the motif of Figure 6/b is also a polygon.

By designing the motif, the user has to specify the coordinates of the points of the polygon(s).

The 'motif data set' is an annotate-like data set with the following variables: FUNCTION, COLOR, SIZE, LINE, X, Y. Their descriptions are the same as those of the corresponding variables of FUNCTION='DRAW' on page 123 of (2). Only two functions are valid in the 'motif data set': MOVE and DRAW. To define the coordinates of the points in the motif, the user can choose any coordinate system.

Figure 2 is the definition of a simple motif. That motif is one polygon made up by two lines. The 'motif data set' has three observations for the three points. In that example, the points are (0,1), (1,0), (2,1). However, any other coordinate system can be used and the following three-three points would result the same motif:

(300,400) (400,300) (500,400),
(13,21) (19,15) (25,21),
(-10,40) (0,30) (10,40).

Besides the real annotate variables, the 'motif data set' has one extra variable: END. Variable END is a character 1 variable and defines the corners of the frame. When two segments of the frame meet, a corner is formed. E.g. Figure 5 sketches out the corner area of the top and the left side. Our intention is to keep the frame continuous through the corners. If the corner is requested, the first point of the polygon in the first block of the top is connected to the corresponding point of the last block of the left side. The points can be connected with a right angle or a diagonal line. If the motif is so designed that the first and the last points of a polygon have the same Y coordinates, the polygon (thus the frame) remains continuous through the corner.

To request the corner, an END value has to be assigned to the first and only to the first point of a polygon. END can have two values: R for right angle-type corner and D for diagonal-type corner. If END has blank value, the corner is not formed. Each first point of each polygon may have END value.

When defining a motif, keep in mind that the motif can be as complex as desired. Complex motifs just require more points to be defined and to be drawn. In addition, do not forget that the lines in the motif may have different colors, because the COLOR variable is part of the 'motif data set'.

3. PREDEFINED MOTIFS

Paragraph 2 explains how to define a motif and set up the 'motif data set', but macro %FRAME offers 101 predefined motifs to choose from! It is very easy to choose and use a predefined motif. When macro %FRAME is invoked, supply the motif number instead of the name of the 'motif data set'. E.g. if the user wants to use motif no. 1, then %FRAME(MOTIF=1) has to be defined. See the description of the MOTIF parameter in paragraph 1.

The 101 predefined motifs are shown in Figure 11. A motif similar to the one in Figure 2 is among the predefined motifs. It is motif 29. Using that predefined motif, the outside frame of Figure 4 can be drawn with only two statements:

%FRAME(MOTIF=29,HSPACE=3,VSPACE=5,HNM=12,VNM=8,OUT=B,ROTATE=180);
PROC GSLIDE ANNOTATE=B;

Notice that the macro rotates the motif by 180 degrees in order to obtain the exact motif of Figure 2.

4. EXAMPLES

Figure 6 shows some examples for setting up 'motif data sets' and generating their frames by calling macro %FRAME. The frames are drawn by the GSLIDE procedure. Notice that the motifs of the frames 'b' and 'e' are also polygons, though those frames are built up by arcs. Examples 'c', 'e' and 'f' have one polygon, the other motifs have two. Look at the END variable. E.g. the first polygon of motif 'a' has END='R' value, the second polygon has no END value.
Examples

Figure 6
Five motifs from these six (except motif 'f') are predefined motifs, the user does not have to design them and set up their 'motif data sets'. Their frames could be generated very easily. E.g. let us create frame 'd'. Its predefined motif number is 96 (check Figure 11). The following two commands would generate the frame of Figure 6/d:

```
%FRAME(MOTIF=96,HSPACE=3,VSPACE=5,HNM=9,VNM=6,OUT=B)
PROC GSLIDE ANNOTATE=B;
```

5. FRAME SELECTION: MACRO %FRAMESL

Macro %FRAMESL selects certain parts of the frame and moves them to a new location. The frame may consist of eight parts. The eight parts designated by the letters T, R, B, L, ULC, URC, LRC, LLC. See the description of variable EDGE in paragraph 1.

Any part(s) of the frame can be selected using the SELECT parameter. The selected part(s) can be moved anywhere on the display area by specifying the new location. To specify the new location, the user has to determine four points as seen in Figure 7. The four points are P1, P2, P3, P4. Their coordinates are given by the P1X, P1Y, P2X, P2Y, P3X, P3Y, P4X, P4Y macro parameters respectively.

Moving the frame

```
Figure 7
```

All P1X, P1Y, ..., P4Y parameters do not have to be specified all the time, only when it is necessary. For example, if the user selects the top of the frame and wants to move it, then only P2X, P3X, P1Y and P2Y parameters have to be specified. These four coordinates determine the new location of the top unambiguously.

Parameters:

**DATA** - the name of the input data set which contains the frame created by macro %FRAME. Default value is _LAST_.

**OUT** - the output data set which contains the selected part(s) of the frame at the new location. If this data set is defined as the annotate data set of a graphic procedure, it yields the selected part(s) drawn. Default value is OUT.

**SELECT** - list of the part(s) of the frame to be selected. The names of the parts are T, R, B, L, ULC, URC, LRC, LLC. If more than one part has to be selected, the names of those parts have to follow one another having at least one space between them. E.g. to select the top, upper right corner and the bottom of the frame, specify SELECT=T URC B. If the entire frame is going to be selected, the ALL option can be used instead of spelling out the names of all parts: SELECT=ALL. Default value is ALL.

**P1X**

**P1Y**

**P2X**

**P2Y**

**P3X**

**P3Y**

**P4X**

**P4Y**

the coordinates of the points determining the new location. Default values are the coordinates of the points determining the original location of the frame.

6. FRAME FANTASY

Using the two macros, %FRAME and %FRAMESL, the user has an almost limitless frame-creating capability. The only limitation is the user's fantasy. So, let our fantasy work and let us enjoy some results in Figures 8-10.

```
Combination of different frames
```

```
DATA ANNOTATE:
SET TOP LEFT BOTTOM RIGHT:
PROC GSLIDE ANNOTATE=ANNOTATE;
```

```
Figure 8
```

520
Figure 8 is a combination of four frames. They are based on the predefined motifs 86-89. In the figure, the top of motif 86, the right side of motif 87, the bottom of motif 88 and the left side of motif 89 are combined into one frame-ory. The selection of the previous frame parts are done by macro %FRAMESL. The four motifs have the same corners, that makes their combination possible. The four corners are selected from the first frame, see the parameter SELECT=T ULC URC LLC LRC.

Figure 9

Figure 9 is another frame-ory of a multiple frame, but this time the different frames are nested. The process is similar to that of Figure 8, but the PIX, PLY,...,P4X coordinates are specified here and they are different for every frame. The motifs incorporated in this figure are 41, 54, 97 and 1.

Figure 10

Figure 10 could be the emblem of our SUGI 12 Conference. In spite of the look, it is not nested, it is not even a multiple frame. The motif is very simple, it is ten parallel lines with increasing thickness. Thus, this motif has ten polygons. The END value, specified for each line, plays a very important role, since the corners cover larger area of the frame than the other parts. The program drawing Figure 10 follows:

DATA A;
DO Y=1 TO 10;
  SIZE=Y;
  FUNCTION='MOVE'; X=0; END='B'; OUTPUT;
  FUNCTION='DRAW'; X=1; END=' '; OUTPUT;
END;
GOPTION NOTEXT82;
%FRAME(MOTIF=A,VSPACE=35,HSPACE=35)
PROC GSLIDE ANNOTATE=OUT;
NOTE; NOTE; NOTE; NOTE; NOTE; NOTE;
NOTE; NOTE; NOTE; NOTE; NOTE; NOTE;
NOTE; NOTE J=C H=3 F=XSWISS 'SUGI 12';

CONCLUSION

Macros %FRAME and %FRAMESL give the user a very simple and powerful tool to create unusual and fancy frames. Use them to make your graph more impressive and spectacular!

REFERENCE


For more information please contact:

Akos Felsovalyi
J.C. Penney, 39th Floor
1301 Ave. of the Americas
New York, NY 10019
(212) 957-6696

SAS, SAS/GRAPH are registered trademarks of SAS Institute, Inc., Cary, NC, USA
101 predefined motifs

Figure 11