RANKPLOT: A SAS MACRO FOR GENERATING LINE PRINTER SCATTERPLOTS OF POINTS AND POINT LABELS WITHOUT OVERPRINTING THE POINTS AND LABELS

Warren F. Kuhfeld, The University of North Carolina

ABSTRACT

Scatterplots are typically produced after a multidimensional scaling, principal components, or other exploratory multivariate data analysis for use as an aid to configuration interpretation. Standard line printer plots generally do not contain enough information to clearly identify all points. Moreover, some points may be hidden. A SAS macro that generates scatterplots where points are plotted by coordinate value ranks instead of original coordinate values will be discussed. This method has the advantage that long point descriptions can be printed with no overprinting or hiding of points and descriptions. There is a trade off; accurate distance information is sacrificed for complete point descriptions.

INTRODUCTION

Generating scatterplots of derived point configurations to aid the process of configuration interpretation is a standard procedure after a multidimensional scaling (MDS), principal components, or other exploratory multivariate data analysis. Scatterplots from PROC ALSCAL or PROC PLOT produce fairly accurate pictures of the overall shape of the configuration, but they do not provide the information that is most needed for configuration interpretation -- the names or descriptions of the individual points in the configuration. PROC PLOT allows for one character, variable plotting symbols, but one character is clearly not an adequate description of a point in all but the most trivial of cases. Another problem with these plots is that some of the points may not appear on the plot because more than one point may map to the same print position. Point descriptions could be put in by hand, and hidden points could be identified by hand, but this would be tedious if there is a large number of points or dimensions. A better approach would be to use a plotting routine that will print long point descriptions, without overprinting or hiding points. This paper describes a SAS macro RANKPLOT, that can plot up to 125 points, with point descriptions that are up to 60 characters long, with no overprinting. A plot produced by this macro will be compared to a plot from PROC PLOT with a variable plotting symbol, and to a plot produced by a macro written by Sarle (1983) that generates scatterplots of point descriptions.

THE RANK PLOT MODEL

One of the simplest methods of dimension interpretation is to sort the points on each of the r dimensions, and print r lists of point descriptions and their corresponding coordinate values. Inspection of the order of points on a dimension, particularly if the data were analyzed with the INDSCAL model, could lead to hypotheses concerning possible interpretations of the dimensions. This approach can be thought of as generating r unidimensional vertical plots where the location of a point on a plot is the rank of the coordinate of the point on the dimension. The rank plot model is an extension of this basic idea to two dimensions. The Y coordinate of a point in a rank plot is (an integer multiple of) the rank of the coordinate of the point on dimension i (i=1,...,r) and the X coordinate is (an integer multiple of) the rank of the coordinate of the point on dimension j (j=1,...,r and i NE j). The actual coordinate values of the point are printed on the ordinate and abscissa.

The rank plot can be used in the same way that scatterplots are used in interpreting MDS configurations. It is relatively easy to superimpose the results of a cluster, discriminant, or regression analysis on the plot as a further aid to interpretation.

AN EVALUATION OF THE RANK PLOT MODEL

The one major advantage of using rank plots is that long descriptions of points can be plotted without overprinting or hiding descriptions. Of course there are also disadvantages to this model. Plotting points by coordinate ranks involves a monotonic, order preserving transformation of the coordinates that destroys meaningful distance information. Large distances are shrunk and small distances are expanded. If the scaling was nonmetric, an order preserving transformation should not be a major concern. It is more of a concern after a metric analysis. In either case, the rank plot should not take the place of the standard scatterplot; it should be a supplement. The scatterplot provides a picture of the global structure of a configuration. The rankplot can be used to quickly see what points are close together or similar, and what points are far apart or dissimilar.

If a researcher decides that the disadvantages of a rank plot outweigh the advantages, and wants to fill in point descriptions on a scatterplot by hand, a rank plot can still be a valuable tool. It is easier to identify the points in a scatterplot with a rank plot as a guide, than with a list of coordinates and descriptions.

LIMITATIONS

No more than 125 points may be plotted. There are 132 print positions on most line printers, and 7 positions are needed for the boundary and for labeling the ordinate. This is not a serious limitation for MDS configurations since most MDS analyzes are done on much smaller datasets. Also, while there is no built in limit on the number of points in PROC ALSCAL, running it with more than 125 points would be expensive or impossible due to its time and memory requirements.

The RANKPLOT macro does not have any built in limit on the length of point descriptions. A sixty character description will always fit on a page (though it may extend beyond the right edge of the plot). If the point happens to fall in one of the first few print positions, descriptions of over 100 characters are possible, but very long descriptions will tend to make the plot more difficult to work with. A conservative upper bound on the length of descriptions that can fit in a plot without extending into the margin can be computed as follows:

```
EXPAND = 1;
IF (# POINTS) < 63
THEN EXPAND = ROUND(100/(# POINTS),1);
SIZE = (# POINTS)*EXPAND;
UPPER LIMIT = FLOOR(SIZE/2)-1;
```

If the description of a point is longer than this limit, and if the point happens to fall in the middle of a line, then the description will not be truncated, but it may run over the right edge of the plot. Descriptions that are too long to fit on the page will be truncated.

USING THE RANKPLOT MACRO

Four keyword parameters may be specified when invoking the macro.

_X The name of the variable that contains the X coordinates (default: DIM1).

_Y The name of the variable that contains the Y coordinates (default: DIM2).

_NAME The name of a character variable that contains the point descriptions (default: NAME).

DATA The name of the SAS dataset that contains the coordinates and descriptions (default: _LAST_).

NOTES ON THE ALGORITHM

Ranks are assigned by sorting the dataset and then assigning ranks in a data step with _N_. PROC RANK would be more efficient, but it does not have an option for arbitrary assignment of ranks within ties. Since the datasets will be small the inefficiency of using PROC SORT and data steps should not be a problem.

The plot is generated in a data step using the report generation facilities with a full page output buffer. An asterisk is printed where the point is and the description is printed on the right if there is room. If there is no room on the right, the description is printed on the left if there is room. If there is on either side, the description is printed on the right and it will extend past the right boundary of the plot.

AN EXAMPLE

Newberry and Kuhfeld (unpublished) gave subjects 100 cards describing common city scenes and asked them to lay out the cards on a grid in the way that the "ideal" city would be planned. Distances between points were computed and the data were analyzed with PROC ALSCAL.

Figure 1 contains the scatterplot of the 2-dimensional ordinal solution produced by PROC PLOT with the following code:

```
OPTIONS PS=64 LS=80;
PROC PLOT;
     PLOT DIM2*DIM1=NAMES;
```

The OPTIONS statement was used to produce a plot that was close to square since the range of DIM1 and DIM2 is about the same. Figure 2 contains the RANKPLOT version produced by the following macro invocation:

```
%RANKPLOT(_X=DIM1,_Y=DIM2,_NAME=NAMES);
```

Figure 3 contains a plot produced by a macro written by Sarle. All plots were printed at 8 lines per inch, and the two macro produced plots spanned two pages of computer paper.

The rank plot gives the mistaken impression that the space is rectangular. (Of course PROC PLOT, left to its own devices does the same thing.) It also does not show (unless you look at the values on the ordinate and abscissa) how far the pornographic movies and testing laboratory are from the other points. In addition, the barber looks more isolated than it

really is. These facts again point out the need for the use of a scatterplot as a supplement to a rank plot.

On the favorable side, the rank plot immediately shows, without a lot of manual work the clusters in the space. Most of the nicer, more expensive places are in the top right. Sleazy places are in the bottom left. Most family oriented places are in the lower right quadrant. Commercial, government and office places are near the center.

The plot produced by Sarle's macro looks a lot like the rank plot. It is better than the rank plot on the edges where the density of points is low because complete descriptions are printed and distances are shown more accurately. In the center where the density is high, some points and parts of many descriptions are hidden or broken into more than one piece. On the line right below 0 for example, the office tower, county building, and newstand are completely hidden by the emergency service headquarters and federal building with the federal building breaking up the emergency service headquarters into two parts. The rank plot macro printed every description in its entirety.

REFERENCES NOTES

Newberry, B. H. & Kuhfeld, W. F. An unpublished study of the layout of the "ideal" city.

Sarle, W. A SAS macro for generating scatterplots. SAS Views: Exploratory Multivariate Data Analysis. 1983.

FIGURE 1

PROC PLOT OUTPUT

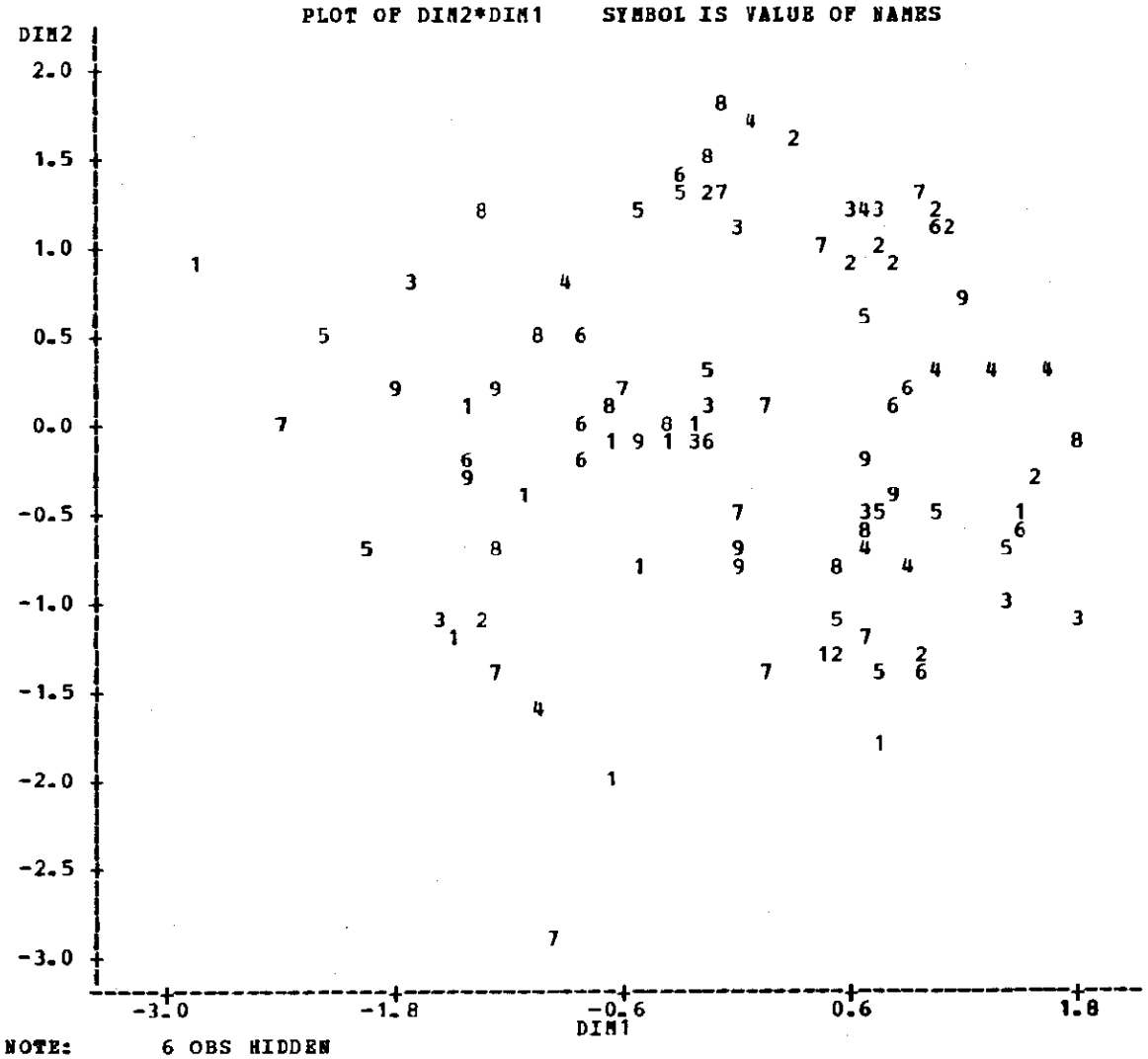
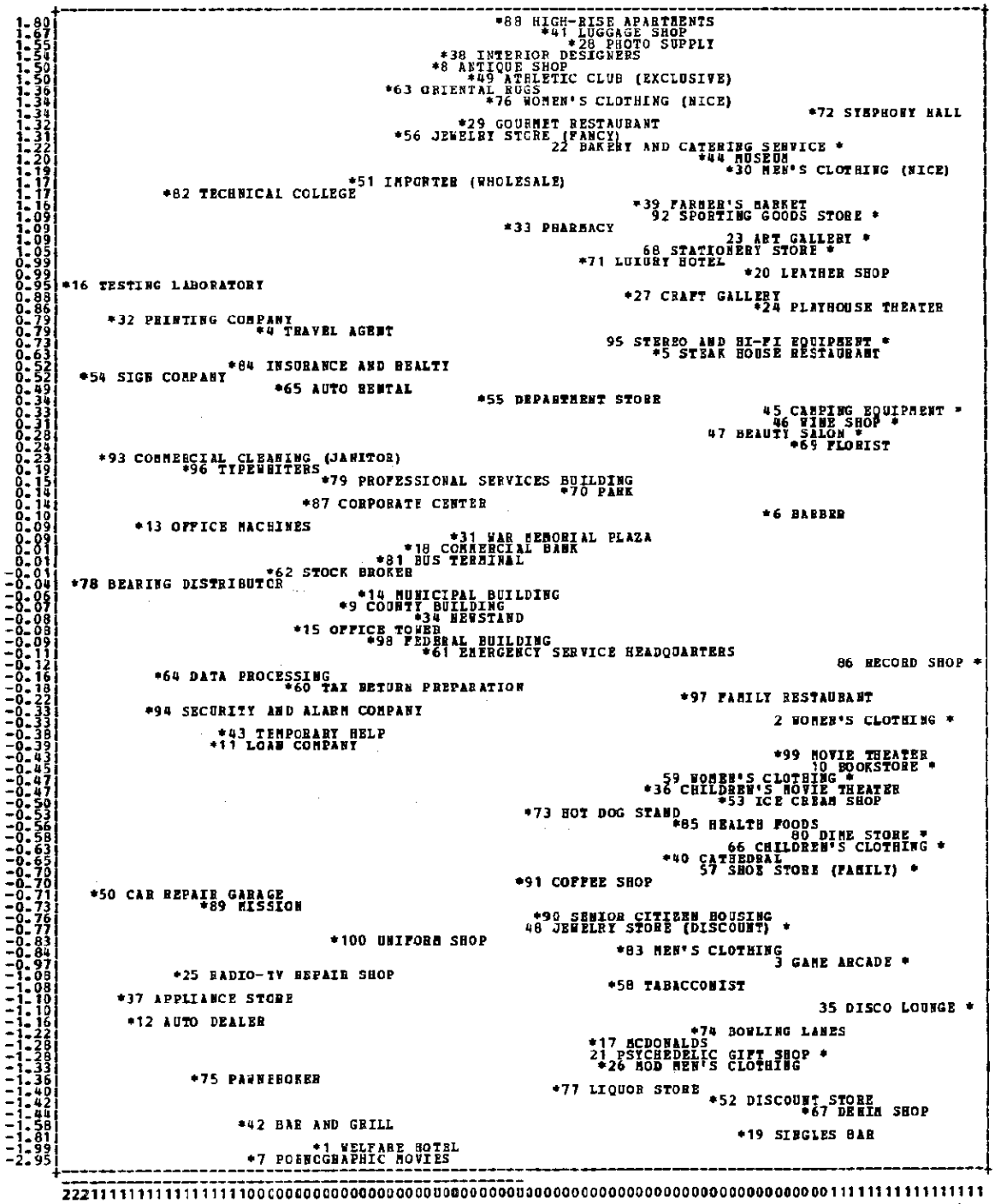


FIGURE 2

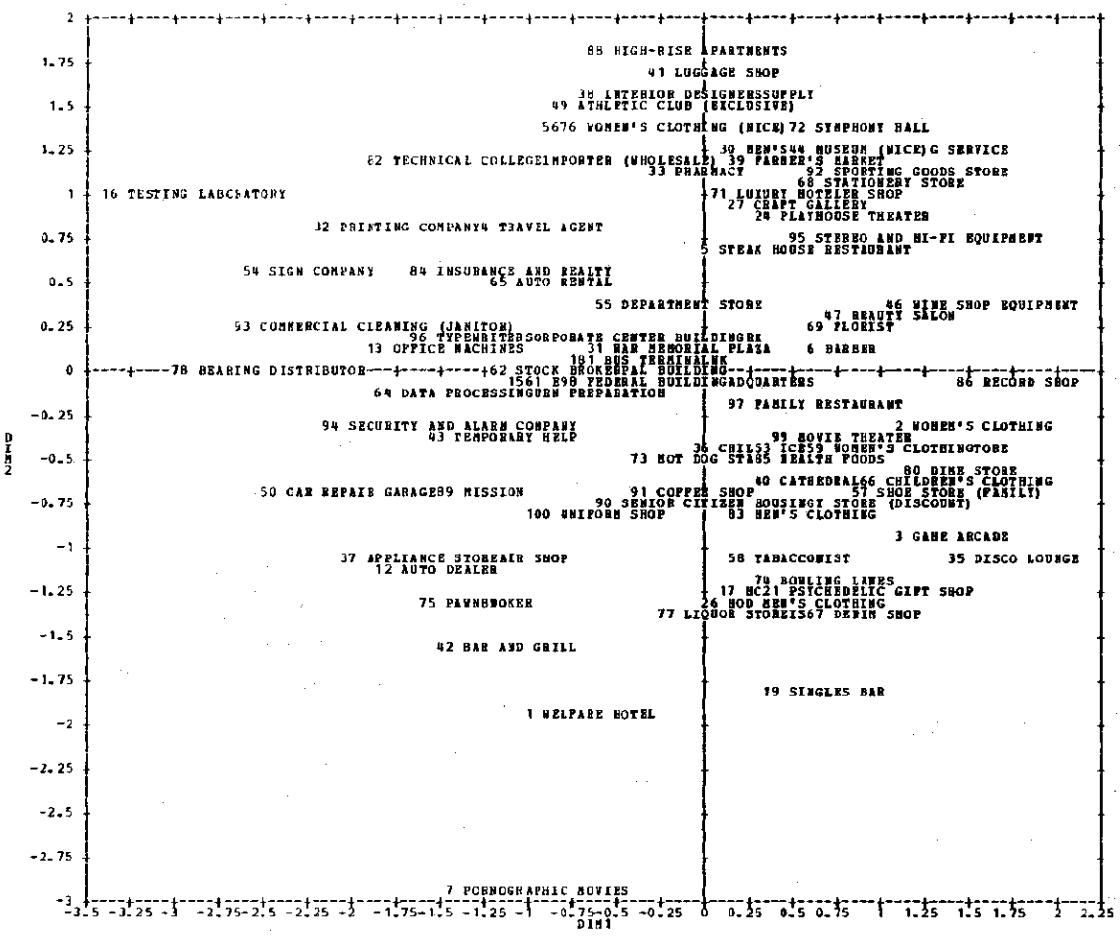
RANKPLOT OUTPUT



PLOT OF DIM2 (ON THE ORDINATE) AND DIM1 (ON THE ABSCISSA)

FIGURE 3

SARLE'S MACRO OUTPUT



MACRO RANKPLOT CODE

```
%MACRO RANKPLOT(_X=    DIM1,   /* X COORDINATES       */
                _Y=    DIM2,   /* Y COORDINATES       */
                _NAME=NAME,    /* POINT DESCRIPTIONS  */
                _DATA=_LAST_) /* SAS DATASET         */
                ;
*-------------------------------------------------------------------+
| MACRO NAME:  RANKPLOT                                             |
| PURPOSE:     PRODUCE A TWO DIMENSIONAL PLOT OF &_X VERSUS &_Y     |
|              BASED ON THE RANKS OF &_X AND &_Y SUCH THAT THE      |
|              NO POINT ON THE PLOT IS PLOTTED IN THE SAME LINE     |
|              AND COLUMN AS ANY OTHER POINT                        |
| LIMITATIONS: UP TO 125 POINTS MAY BE PLOTTED AND THE              |
|              DESCRIPTION OF THE POINT MAY BE UP TO 60             |
|              CHARACTERS LONG.  THIS LIMIT OF 60 CHARACTERS        |
|              ASSUMES THAT IT IS ALL RIGHT TO HAVE A DESCRIPTION   |
|              OF A POINT CONTINUE OUTSIDE THE BOUNDS OF THE        |
|              PLOT.  IF THIS IS NOT DESIRABLE THEN THE LENGTH      |
|              OF THE NAME SHOULD BE SHORTER.  A CONSERVATIVE       |
|              UPPER BOUND ON POINT NAME LENGTH MAY BE              |
|              COMPUTED WITH THE FOLLOWING EQUATIONS:               |
|              EXPAND = 1                                           |
|              IF (# POINTS) < 63                                   |
|                 THEN EXPAND = ROUND(100/(# POINTS),1)             |
|              SIZE = (# POINTS) * EXPAND                           |
|              CONSERVATIVE UPPER LIMIT = FLOOR(SIZE/2) -1          |
| PARAMETERS:  &_X     -- THE NAME OF THE HORIZONTAL AXIS           |
|                         VARIABLE                                  |
|              &_Y     -- THE NAME OF THE VERTICAL AXIS             |
|                         VARIABLE                                  |
|              &_NAME -- THE NAME OF THE CHARACTER VARIABLE         |
|                         CONTAINING THE POINT NAMES                |
|              &_DATA -- THE NAME OF THE SAS DATASET                |
| DESCRIPTION  THE LOCATION OF A POINT ON THE PLOT IS DETERMINED    |
| OF THE       BY (AN INTEGER MULTIPLE OF) THE RANKS OF THE &_X     |
| OUTPUT PLOT: AND &_Y VARIABLES.  AN ASTERISK (*) IS PRINTED IN    |
|              COLUMN:  CONSTANT * RANK(&_X)                        |
|              AND ROW: CONSTANT * RANK(&_Y).                       |
|              THE DESCRIPTION IS PRINTED AFTER THE * OR            |
|              RIGHT BEFORE THE * (DEPENDING ON THE NUMBER OF       |
|              AVAILABLE PRINT POSITIONS), BUT IN ALL CASES THE     |
|              DESCRIPTIONS APPEAR ENTIRELY ON THE SAME LINE AS     |
|              THE * AND THEY ARE NOT OVERPRINTED.                  |
| WRITTEN BY:  WARREN F. KUHFELD                                    |
|              L. L. THURSTONE PSYCHOMETRIC LABORATORY              |
|              DAVIE - 013A, UNC CHAPEL HILL, NC 27514              |
|              (919) 962-7643                                       |
+-------------------------------------------------------------------+
*-----COMPUTE THE RANK FOR &_X-----;
PROC SORT DATA=&_DATA OUT=__TEMP;
     BY &_X;

DATA __TEMP NOBS(KEEP=XRANK);
     SET __TEMP END=EOF;
     XRANK = _N_;
     OUTPUT __TEMP;
     IF EOF THEN OUTPUT NOBS;  /* NUMBER OF OBSERVATIONS */

*-----SORT ON &_Y FOR COMPUTING RANK OF &_Y-----;
PROC SORT DATA = __TEMP OUT = __TEMP;
     BY DESCENDING &_Y;

OPTIONS PS=135;
DATA _NULL_;
     *-----DETERMINE THE SIZE OF THE PLOT-----;
     SET NOBS(RENAME=(XRANK=N_OBS));
     IF N_OBS < 63 THEN BLOWUP = ROUND(100/N_OBS,1);
                   ELSE BLOWUP = 1;
     LEFTLINE = 6;
     TOPLINE  = 1;
     RIGHTLINE = LEFTLINE + 1 + BLOWUP*N_OBS;
     BOTTLINE = TOPLINE  + 1 + BLOWUP*N_OBS;
     LENGTH XLABEL $ 5 X1-X5 $ 1;
     FILE PRINT NOTITLES N=PS;
     *-----DRAW THE OUTLINE OF THE PLOT-----;
     DO I = 1 TO BLOWUP*N_OBS;
        LPOINT = LEFTLINE + I;
        TPOINT = TOPLINE  + I;
        PUT #TOPLINE    @XPOINT     '-'
            #BOTTLINE   @XPOINT     '-'
            #TPOINT     @LEFTLINE   '|'
            #TPOINT     @RIGHTLINE  '|';
        END;
     PUT #TOPLINE    @LEFTLINE   '+'
         #TOPLINE    @RIGHTLINE  '+'
         #BOTTLINE   @LEFTLINE   '+'
         #BOTTLINE   @RIGHTLINE  '+';
     *-----THE START OF THE PLOTTING ROUTINE-----;
     YRANK = 0;
     EOF   = 0;
     DO UNTIL(EOF);
        SET __TEMP END = EOF;
        *-----DETERMINE THE LOCATION OF THE '*'-----;
        XPOINT = XRANK*BLOWUP + LEFTLINE;
        YRANK  = YRANK + 1;
        YPOINT = YRANK*BLOWUP + TOPLINE;
        *-----PRINT &_Y ON THE VERTICAL AXIS AND PRINT THE '*'-----;
        PUT #YPOINT @1 &_Y 5.2
                    @XPOINT '*';
        *-----SHOULD &_NAME BE ON THE LEFT OR RIGHT?-----;
        COLLEFT = RIGHTLINE - XPOINT;
        LENGTH  = LENGTH(&_NAME);
        LABPOINT = XPOINT + 1;
        IF (LENGTH > COLLEFT) AND ((XPOINT - LEFTLINE) > LENGTH)
           THEN LABPOINT = (XPOINT - LENGTH) - 1;
        *-----MAKE SURE &_NAME WILL FIT ON THE PAGE-----;
        IF (LABPOINT > XPOINT) AND (LENGTH > (132-XPOINT))
           THEN &_NAME = SUBSTR(&_NAME,1,(132-XPOINT));
        *-----PRINT &_NAME-----;
        PUT #YPOINT @LABPOINT &_NAME;
        *-----PRINT &_X ON HORIZONTAL AXIS-----;
        XLABEL = PUT(&_X,5.2);
        ARRAY XL X1-X5;
        DO OVER XL;
           XL = SUBSTR(XLABEL,_I_,1);
           END;
        YPOINT = BOTTLINE +2;
        PUT #YPOINT (X1-X5) (@XPOINT $1. /);
        END; /* DO UNTIL(EOF) */
     *-----GENERATE THE LEGEND-----;
     __Y = '          ';
     __X = '          ';
     CALL VNAME(&_Y,__Y);
     CALL VNAME(&_X,__X);
     YPOINT = BOTTLINE+8;
     PUT #YPOINT @7 'PLOT OF ' __Y '(ON THE ORDINATE)'
                    ' AND '    __X '(ON THE ABSCISSA)';
     PUT _PAGE_;
     STOP;
OPTIONS PS=60;
%MEND RANKPLOT;
```