

Producing Useful Survey Analysis Graphics with Efficient SAS® Code

Donald T. Gantz, George Mason University, Fairfax, VA
Yan Yuan, The Urban Institute, Washington, D.C.

ABSTRACT

Analyzing and reporting survey data typically requires you to generate many graphical presentations where each graphic reveals relationships between several variables. After you associate a good graphic vehicle with types of variables, you settle down to producing many individual graphs with each based on a subset of the variables. By cleverly designing the production process with flexible coding, you can create an efficient path from raw data to analysis and presentation graphics. Writing reusable modules of code and utilizing SAS macro language are a means to that end. This paper illustrates a particular application we have designed and implemented in our work supporting research on the database from a National Survey of Registered Nurses¹. The application is rooted in a very versatile core of graphics code that is easily modified to produce a varied menu of graphic structures via a front end program feeding it values for macro variables. In addition to demonstrating the utility of SAS macro language and SAS/GRAPH®, this paper makes available to you a graphic form that we have found quite useful.

INTRODUCTION

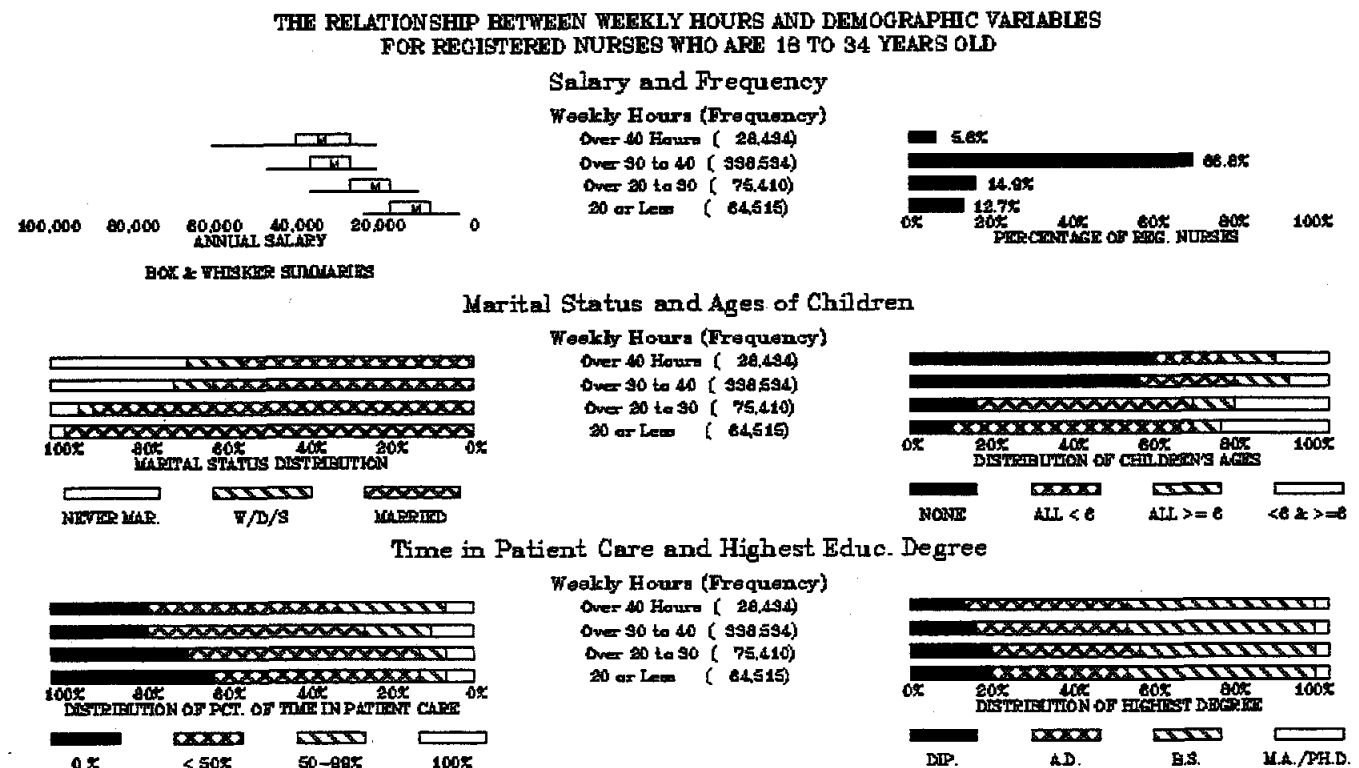
Surveys typically measure lots of information, and it is often overwhelming to analyze, understand and write narrative about survey data. Your narrative concerns patterns in the data, but how do you find patterns in data? The best vehicle for exploring data for patterns is graphics. Good graphics will suck information out of data and stimulate your narrative. This is a generally acknowledged fact that accounts for the current popularity of visualization methods in statistical data analysis. Our choice for a graphics tool is the SAS® System. Like many competing

graphics tools on the market, SAS software via its procedures provides a menu of graphic shells that can display patterns in a variable or relationships between multiple variables. However, in addition to the programming capabilities in SAS software allow us to write code to create custom graphics, and further, to manage that code to efficiently apply it to many survey variables. This paper illustrates that capability.

The survey that we analyze here is the National Sample Survey of Registered Nurses, March 1992. This is the fifth in a series of surveys sponsored by the U.S. Department of Health and Human Services to assess the size and characteristics of the national registered nurse population. The published report on that survey contains many tables but few graphics. We agreed to provide statistical analysis assistance to a George Mason University Nursing Ph.D. student whose thesis was based on this survey. After consultation with this student, we agreed that what she required was a comprehensive set of graphics that would reveal nurses' characteristics and drive her thesis research. This meant that we would generate a large quantity of graphics that hopefully would contain the necessary nuggets of gold for her research — revealing patterns of substantive characteristics.

We selected some templates from a repertoire of graphics that had proven successful in prior work. Then we set out to rework our existing SAS code not only to apply to this survey, but also to be available to us as generic graphics applications. The process went well; the mounds of graphics provided the hoped for nuggets of gold, the student completed her Ph.D. and we have our generic applications. We present one of those applications here with the intent of both putting that application into your SAS toolbox and passing on our approach to application building.

Figure 1



RUNNING THE GRAPHICS APPLICATION

When we want to build a graphic that analyzes several variables jointly, we like to use what we refer to as wing style charts. These wing charts simultaneously display the distributions of two variables conditioned on levels of a third variable. Figure 1 presents an example of three wing charts produced as a composite. This example looks at the relationship between a nurse's weekly work hours and six demographic variables. The conditional distributions of five of the demographic variables are graphically presented by three separate wing charts while one full page of charts is prepared for each category of the sixth demographic variable which is age. Each wing chart has a central variable, a left wing variable and a right wing variable. A line in the chart consists of the distributions of the left and right variables for one category of the central variable. The order in which the central categories are stacked in each wing chart provides another dimension of visual information.

The second and third wing charts in Figure 1 use segmented bar charts to give the conditional distribution of the wing variable at a

given category of the central variable. Some variables like Marital Status and Highest Degree are truly categorical; other variables like Percent of Time Spent in Patient Care are initially categorized into three or four categories in a data step. The first wing chart in Figure 1 uses bars on the right side to represent the percentages of nurses in each of the central categories. On the left side, box & whisker diagrams are used to show how salary distributions vary across the categories of the central variable. Box & whisker diagrams are visual summaries of key percentiles of a quantitative variable. There are a variety of constructs for box & whisker diagrams. In our variant, the box portion spans from the 25th percentile value to the 75th percentile value. An 'M' is plotted within the box to mark the location of the 50th percentile (i.e. the median). The whiskers (bottom lines) extend out, in this case, to the 5th percentile value on the left and to the 95th percentile value on the right.

The programs that produce Figure 1 also created listings of all numerical summaries behind the graphics. For instance, the two listings in Table 1 below are associated with Figure 1

Table 1 Percentiles of SALARY within Work Hours Per Week

Work Hours Per Week	5th Percentile	10th Percentile	25th Percentile	Median	75th Percentile	90th Percentile	95th Percentile
20 or Less	3,300	5,000	8,643	13,750	18,229	23,000	26,000
Over 20 to 30	13,000	15,000	19,000	24,000	30,000	35,000	40,000
Over 30 to 40	22,000	24,000	28,000	32,795	38,000	44,000	48,492
Over 40 Hours	23,500	25,000	30,000	35,000	42,000	50,000	60,000

BREAKDOWN OF MARITAL STATUS DISTRIBUTION BY WORK HOURS PER WEEK
ROW PERCENTAGES ADD TO 100%

Work Hours Per Week	MARRIED	W/D/S	NEVER MAR.
20 or Less	95.2	1.1	3.7
Over 20 to 30	89.5	3.7	6.8
Over 30 to 40	63.3	8.6	28.1
Over 40 Hours	56.5	12.3	31.2

The following shell program produced Figure 1 by three times invoking the macro %ZITO and once invoking the macro %ZIT1.

```
*FIGURE1.SAS
;
*EXTERNAL FILE CONTAINING THE MACRO ZITO;
FILENAME BAR 'D:\WING01.SAS';
*EXTERNAL FILE CONTAINING THE MACRO ZIT1;
FILENAME GRAPH 'D:\GRAPH.SAS';
*DATA STEP FORMING THE DATA SET FOR ZITO;
FILENAME DATASTEP 'D:\DS01.SAS';

OPTIONS NODATE LS=80 PS=500 MLOGIC SYMBOLGEN;

%INCLUDE BAR;
%INCLUDE GRAPH;

%ZITO(
  ANNUAL SALARY,
  %STR(PERCENTAGE OF REG. NURSES),
  Work Hours Per Week,
  %STR(Weekly Hours (Frequency)),
  Salary and Frequency,
  %STR(CATEGORY=PUT(HOURS,HOURS.)),
  ,
  ,
  %STR(IF (18 LE AGE1 LE 34)),
  $HOURS1F,ANNOF19A,
  , , , ,
)
```

```
, , , ,
0,
ABCD,3,4,
1,100,
1,100000,
SALARY,
1,W
);

%ZITO(
  MARITAL STATUS DISTRIBUTION,
  %STR(DISTRIBUTION OF CHILDREN%'S AGES),
  Work Hours Per Week,
  %STR(Weekly Hours (Frequency)),
  Marital Status and Ages of Children,
  %STR(CATEGORY=PUT(HOURS,HOURS.)),
  %STR(VAR1=PUT(Q27,$Q27AF.)),
  %STR(VAR2=PUT(Q28,$Q28AF.)),
  %STR(IF (18 LE AGE1 LE 34)),
  $HOURS1F,ANNOF19B,
  MARRIED,W/D/S,NEVER MAR., ,
  NONE,ALL < 6,ALL >= 6,<6 & >=6,
  0,
  ABCD,3,4,
  2,100,
  2,100000,
  SALARY,
  1,W
);
```

```
%ZITO(
  %STR(DISTRIBUTION OF PCT. OF TIME IN PATIENT CARE),
  %STR(DISTRIBUTION OF HIGHEST DEGREE),
  Work Hours Per Week,
  %STR(Weekly Hours (Frequency)),
  Time in Patient Care and Highest Educ. Degree,
  %STR(CATEGORY=PUT(HOURS,HOURS.)),
  %STR(VAR1=PUT(TIME,Q13ACF.)),
  %STR(VAR2=PUT(PREPSUM,$PREPSUM.)),
  %STR(IF (18 LE AGE1 LE 34)),
  $HOURS1F,ANNOF19C,
  %STR(0 %%),%STR(< 50%%),
  %STR(50-99%%),%STR(100%%),
  DIP.,A.D.,B.S.,%STR(M.A./PH.D.),
  0,
  ABCD,4,4,
  2,100,
  2,100000,
  SALARY,
  1,W
);
```

```
%ZIT1(
  B.ANNOF19A,B.ANNOF19B,B.ANNOF19C,
  %STR(THE RELATIONSHIP BETWEEN WEEKLY HOURS
  AND DEMOGRAPHIC VARIABLES),
  %STR(FOR REGISTERED NURSES WHO ARE 18 TO 34
  YEARS OLD),
  50,100,61,32,0,.75
)
```

RUN;

The macro %ZITO contains the generic code that produces one wing chart. The shell program supplies values for local macro variables in %ZITO. These variables are:

```
%MACRO ZITO(
  TITLE1,TITLE2,TITLE3,TITLE4,
  SUBTITLE,
  CATEGORY,VAR1,VAR2,LOGIC,
  FMT,ANNO,
  LAB1_1,LAB2_1,LAB3_1,LAB4_1,
  LAB1_2,LAB2_2,LAB3_2,LAB4_2,
  SORT,
  STRING,GROUPS_1,GROUPS_2,
  RIGHTBAR,RIGHTMAX,
  LEFTBAR,LEFTMAX,
  BWLABEL,
  WEIGHT,W_VAR
);
```

```
*****;
*DEFINITIONS OF MACRO VARIABLES FROM THE SHELL PROGRAM: ;
* ;
*TITLE1 IS THE LABEL FOR THE LEFT VARIABLE (UPPER CASE) ;
*TITLE2 IS THE LABEL FOR THE RIGHT VARIABLE (UPPER CASE) ;
*TITLE3 IS THE LABEL FOR THE CENTER VARIABLE (Lower Case) ;
  THAT IS USED IN AUXILIARY LISTINGS ;
*TITLE4 IS THE LABEL FOR THE CENTER VARIABLE (Lower Case) ;
  THAT IS USED IN THE GRAPH ;
*SUBTITLE IDENTIFIES THE DATA SUBSET (IF ANY) THAT IS ;
  BEING SUMMARIZED ;
*CATEGORY IS THE DATA STEP CODE DEFINING THE CENTRAL ;
  VARIABLE ;
*VAR1 IS THE DATA STEP CODE DEFINING THE LEFT VARIABLE ;
*VAR2 IS THE DATA STEP CODE DEFINING THE RIGHT VARIABLE ;
*LOGIC IS THE DATA STEP CODE DEFINING A SUBSET OF DATA ;
  IF REQUIRED ;
*FMT IS THE FORMAT (FROM FORMAT1.SAS) FOR CATEGORY ;
*ANNO IS THE ANNOTATE DATA SET THAT IS BEING CREATED ;
*LAB1_1,...,LAB4_1 ARE LABELS FOR LEFT BAR SEGMENTS ;
*LAB1_2,...,LAB4_2 ARE LABELS FOR RIGHT BAR SEGMENTS ;
*SORT ACTIVATES THE SORT BY CENTRAL CATEGORY FREQUENCIES ;
  * OPTION (1=SORT 0=NO SORT) ;
*STRING ABCD ;
```

```
*GROUPS_1 THE NUMBER OF BAR SEGMENTS IN THE LEFT BARS ;
*GROUPS_2 THE NUMBER OF BAR SEGMENTS IN THE RIGHT BARS ;
*RIGHTBAR INDICATES THE STRUCTURE OF THE RIGHT BAR: ;
  * 1=FREQUENCY OF CENTER VARIABLE ;
  * 2=SEGMENTED BAR FOR A SEPARATE VARIABLE (VAR2) ;
*RIGHTMAX MAX VALUE FOR RIGHT BAR WHEN RIGHTBAR=1 ;
*LEFTBAR INDICATES THE STRUCTURE OF THE LEFT BAR ;
  * 1=BOX & WHISKER ARRAY ;
  * 2=SEGMENTED BAR FOR A SEPARATE VARIABLE (VAR1) ;
*BWLABEL IS A LABEL FOR THE BOX & WHISKER VARIABLE ;
*WEIGHT INDICATES THAT DATA VALUES ARE TO BE WEIGHTED ;
  * (1=WEIGHTED) ;
*W_VAR IS THE WEIGHT VARIABLE ;
*****;
```

Note that the information provided to %ZITO macro variables consists of selections for wing chart structural parameters, code to define central and wing variables, data set and format names, labels and titles. For Figure 1, some of the macro variable specifications for the three %ZITO invocations are identical since the central variable, Weekly Hours (in its natural order sort), and the cut to 18 to 34 year old nurses are the common theme of the page. Figure 1 represents but one of a vast number of alternatives available through %ZITO. In fact, the variety of graphic content in the three wing charts in Figure 1 demonstrate some of the options and flexibility available.

The second macro %ZIT1 combines the three annotate data sets created by the %ZITO invocations and uses the GSLIDE procedure to produce Figure 1. The titles for Figure 1 and the numbers that scale the annotate SIZE variable are fed to %ZIT1. The following code contains %ZIT1.

*GRAPH.SAS;

```
%MACRO
ZIT1(ANNO01,ANNO02,ANNO03,TITLE01,TITLE02,
  X1,X2,Y1,Y2,Y3,S1);
```

LIBNAME B 'D:\GEOLOT\DATA\';

```
DATA ANNOS ;SET &ANNO01 (IN=INA) &ANNO02
(IN=INB)
  &ANNO03 (IN=INC);
```

```
IF INA THEN Y=&X1*(Y/&X2)+&Y1;
IF INB THEN Y=&X1*(Y/&X2)+&Y2;
IF INC THEN Y=&X1*(Y/&X2)+&Y3;
```

SIZE=&S1 * SIZE;

RUN;

*DO THE PLOTTING;

PROC GSLIDE ANNOTATE=ANNOS;

TITLE H=1 F=COMPLEX "&TITLE01"

TITLE2 H=1F=COMPLEX "&TITLE02"

RUN;

%MEND ZIT1;

The macro %ZIT1 is tailored for the combination of three wing charts. Spacing is set by values of the macro variables X1, X2, Y1-Y3 to scale the vertical graphing variable Y and S1 to scale the SIZE variable in the annotate data set. By varying just the %ZIT1 code, which manipulates saved annotate data sets, the number and locations of wing charts on a page can be tweaked as desired.

The technique of using a shell program to succinctly invoke %ZIT0 and %ZIT1, has several advantages:

- Sequences of minor modifications to the shell program will quickly build many useful graphics. For instance, changing only the central variable assignment and sort option builds a stack of graphics that relate many variables to the same set of demographic variables. Also, changing the option that subsets data in the data step provides additional views.
- The code that feeds the option selections into complex logic and detailed graphic plotting does not have to be accessed throughout the generation of many graphs.
- The macro variable values fed to the macro %ZIT1 can be tweaked to fine tune presentation graphics. This was the way that final thesis graphics were produced for the nursing Ph.D. student we were assisting.
- The generic application is available for use with other sets of data via appropriate shell programs.

We usually print a graph by clicking the print button in the graphics window. The graphics within this paper were exported from the graphics window via the EXPORT option on the FILE pull-down menu and subsequent selection of the *.TIF option. The exported TIF file was pulled into Word 97 by the PICTURE option on the INSERT pull-down menu. The size of the graph was then adjusted in Word 97.

OVERVIEW OF THE CORE GRAPHICS CODE

A core of code comprising the macro %ZIT0 is called by shell programs to produce customized graphics. The roots of this code go back over ten years. We initially produced a few wing charts like the top one in Figure 1; the code was specific to a particular survey. Over time for particular needs, we modified that code to produce wing charts that look like the other components of the figures. The nurses' survey we have cited was our incentive to roll all of the options into a single core of code. For that survey it was simply not possible in practical terms to associate each graph on our to-do list with a separate fully customized version of our existing programs. We set about unifying all wing chart formats into a single core of code by utilizing the proficiency with macro language we had gained by building other applications in the same spirit. In fact, we saved a lot of time by utilizing macro language constructs from those other applications. Whereas, when the macro language constructs were first developed, it was with the pain of trial, error and much reading of reference manuals, grabbing them from existing code was quick and smooth. One of our intended consequences for this paper is to provide you with macro language constructs and solutions that have worked for us and which will hopefully reduce your pain while developing applications.

Structurally a wing chart consists of central, left and right sections. The information used to build the left and right sections is conditioned on categories of the central section. In addition, frequencies for the central categories are included in the chart. Accordingly, we use the FREQUENCY procedure to form data sets with chart building frequencies and percentages. Macro language is used here to provide labels, titles and formats, and it also provides the logic capability to allow the shell program to exercise options for sorting, using a weight variable, and so forth. If the sort option is selected in calling %ZIT0, the central categories in the chart are stacked in ascending frequency order. The order of categories is maintained in a data set (named ORDER) which is merged onto the various summary data sets.

The program allows for bar structures with either three or four segments to bars. One chunk of code that generates summary data and builds annotate data sets for bars is looped through twice via macro language — once for left bars and once for right

bars. The shell program tells the code the bar variables, the number of segments, labels and titles.

There is an option for box and whisker diagrams² for the left side of the wing chart; this option requires the use of the UNIVARIATE procedure to calculate percentiles. A chunk of code is conditionally executed when the option is selected. Another option allows for the right side of the wing chart to be annotated bars with percentages for the central categories. Both of these options have been selected for the top wing chart in Figure 1.

All of the information used to build the wing chart is also printed to the output window as labeled and titled listings for inclusion in reports. Examples are shown in Table 1. Additionally, some information is written to the LOG with PUT statements and many of the work data sets are printed for diagnostic purposes. The TRANSPOSE procedure is used to change the observation and variable structure of the PROC FREQ output data sets for their subsequent use. For instance, note the transpositions listed in Table 2 which are for the left side of the middle wing chart in Figure 1. The macro call passed the value VAR1=PUT(Q27,\$Q27AF.) to the macro variable VAR1. Therefore the format

```
value $Q27AF '1'='A' '2'='B' '3'='C';
```

partitions Q27 into three categories with assigned values A, B and C.

**Table 2 DATA SET SET1_1
BAR SEGMENT FREQUENCIES BY CENTRAL CATEGORIES**

OBS	CATEGORY	VAR1	COUNT	PERCENT
1	1	A	61427.40	95.2138
2	1	B	698.79	1.0831
3	1	C	2389.06	3.7031
4	2	A	67517.71	89.5337
5	2	B	2790.36	3.7002
6	2	C	5102.29	6.7660
7	3	A	214276.00	63.2953
8	3	B	29229.54	8.6342
9	3	C	95028.40	28.0706
10	4	A	16070.62	56.5186
11	4	B	3504.65	12.3255
12	4	C	8858.94	31.1559

**DATA SET SET2_1
TRANSPOSED DATA SET SET2_1**

OBS	CATEGORY	_NAME_	A	B	C	ORDER
1	1	PERCENT	95.2138	1.0831	3.7031	1
2	2	PERCENT	89.5337	3.7002	6.7660	2
3	3	PERCENT	63.2953	8.6342	28.0706	3
4	4	PERCENT	56.5186	12.3255	31.1559	4

The transposed data set (SET2_1) is also listed above in Table 1 with formatting, labels and a title. It was easy to read and to understand the logical flow of our ten-year-old program, which was written to produce a wing chart like the top chart in Figure 1 in the context of a specific survey. Now that the program is a flexible application permeated with logic statements and macro language constructs, it is much more difficult to read. The elegance of flexibility has come at the price of clarity and digestibility. The code listed below includes comments that describe the function of each block of code. However, it is impossible to document and explain the many coding quirks that either finally got something to work or bulletproofed the code against some data anomaly. Some of the fixes we incorporated were solutions to stumbling blocks in other applications that we have retrieved from our code repertoire.

CONCLUSION

We often take on consulting projects requiring comprehensive analyses of surveys with many variables and a number of sub-

populations of interest. We can confidently enter an agreement to perform such analyses since we have at hand a rich repertoire of custom graphic applications. We know from the outset what we will be producing and are able to control the workload of generating analyses. We have shared with you one of the applications that has worked well for us. But beyond that, we have shared our approach to writing generic applications. We are in the trenches daily analyzing data with our SAS toolbox. Our mindset in writing code is from the very start to structure a program for future development and generalization. Any additional effort to generalize an application quickly pays off with efficiency gains in future work.

*WING01.SAS

```
;
%MACRO ZITO(
    TITLE1,TITLE2,TITLE3,TITLE4,
    SUBTITLE,
    CATEGORY,VAR1,VAR2,LOGIC,
    FMT,ANNO,
    LAB1_1,LAB2_1,LAB3_1,LAB4_1,
    LAB1_2,LAB2_2,LAB3_2,LAB4_2,
    SORT,
    STRING,GROUPS_1,GROUPS_2,
    RIGHTBAR,RIGHTMAX,
    LEFTBAR,LEFTMAX,
    BWLABEL,
    WEIGHT,W_VAR
);

*****
*DEFINITIONS OF MACRO VARIABLES FROM THE SHELL PROGRAM:
*
*TITLE1 IS THE LABEL FOR THE LEFT VARIABLE (UPPER CASE)
*TITLE2 IS THE LABEL FOR THE RIGHT VARIABLE (UPPER CASE)
*TITLE3 IS THE LABEL FOR THE CENTER VARIABLE (Lower Case)
        THAT IS USED IN AUXILIARY LISTINGS
*TITLE4 IS THE LABEL FOR THE CENTER VARIABLE (Lower Case)
        THAT IS USED IN THE GRAPH
*SUBTITLE IDENTIFIES THE DATA SUBSET (IF ANY) THAT IS
        BEING SUMMARIZED
*CATEGORY IS THE DATA STEP CODE DEFINING THE CENTRAL
        VARIABLE
*VAR1 IS THE DATA STEP CODE DEFINING THE LEFT VARIABLE
*VAR2 IS THE DATA STEP CODE DEFINING THE RIGHT VARIABLE
*LOGIC IS THE DATA STEP CODE DEFINING A SUBSET OF DATA
        IF REQUIRED
*FMT IS THE FORMAT (FROM FORMAT1.SAS) FOR CATEGORY
*ANNO IS THE ANNOTATE DATA SET THAT IS BEING CREATED
*LAB1_1,...,LAB4_1 ARE LABELS FOR LEFT BAR SEGMENTS
*LAB1_2,...,LAB4_2 ARE LABELS FOR RIGHT BAR SEGMENTS
*SORT ACTIVATES THE SORT BY CENTRAL CATEGORY FREQUENCIES;
        OPTION (1=NO SORT 0=NO SORT)
*STRING ABCD
*GROUPS_1 THE NUMBER OF BAR SEGMENTS IN THE LEFT BARS
*GROUPS_2 THE NUMBER OF BAR SEGMENTS IN THE RIGHT BARS
*RIGHTBAR INDICATES THE STRUCTURE OF THE RIGHT BAR:
        1=FREQUENCY OF CENTER VARIABLE
        2=SEGMENTED BAR FOR A SEPARATE VARIABLE (VAR2)
*RIGHTMAX MAX VALUE FOR RIGHT BAR WHEN RIGHTBAR=1
*LEFTBAR INDICATES THE STRUCTURE OF THE LEFT BAR
        1=BOX & WHISKER ARRAY
        2=SEGMENTED BAR FOR A SEPARATE VARIABLE (VAR1)
*BWLABEL IS A LABEL FOR THE BOX & WHISKER VARIABLE
*WEIGHT INDICATES THAT DATA VALUES ARE TO BE WEIGHTED
        (1=WEIGHTED)
*W_VAR IS THE WEIGHT VARIABLE
*****

*****
*INCLUDE THE DATA SET CODE FOR THE DATA TO BE ANALYZED.
*THE PHYSICAL FILE FOR THE LOGICAL NAME DATASET IS DEFINED
*IN THE SHELL PROGRAM.
*MACRO VARIABLES DEFINE CATEGORY, VAR1, VAR2 AND LOGIC
*CODE (IF ANY) FOR EXCLUSIONS.
*****
```

```
%INCLUDE DATASET;
&CATEGORY;
&VAR1;
&VAR2;
&LOGIC;

RUN;

*****
*GET SUMMARIES FOR CENTER VARIABLE CATEGORIES.
*GET PRINTED OUTPUT FROM PROCs FOR DIAGNOSTICS.
*****

PROC FREQ DATA=CENTER1;
    TABLES CATEGORY
        / OUT=CENTER2;
%IF &WEIGHT=1 %THEN %DO;
    WEIGHT &W_VAR;
%END;
    LABEL
    CATEGORY='&TITLE1'
    ;
    TITLE
    '&TITLE3 Categories'
    ;

*RENAME THE CENTER FREQUENCY VARIABLE;

DATA CENTER2;SET CENTER2;
    RENAME COUNT=SUM;

*****
*SORT THE CENTER CATEGORIES IN INCREASING FREQUENCY
    ORDER FOR GRAPHING IF THE SORT OPTION IS SELECTED.
*****
%IF &SORT %THEN %DO;
    PROC SORT DATA=CENTER2; BY SUM;
%END;

*CREATE THE CENTER GRAPHING ORDER VARIABLE;
DATA CENTER2;SET CENTER2;ORDER=_N_;

*SAVE THE CENTER GRAPHING VARIABLE IN DATA SET ORDER;
DATA ORDER;SET CENTER2;
    KEEP CATEGORY ORDER;

*SORT THE DATA SET ORDER FOR LATER MERGING;
PROC SORT DATA=ORDER;BY CATEGORY;

*****
*PRINT THE DATA SET ORDER FOR DIAGNOSTICS.
*****
PROC PRINT DATA=ORDER;

*****
*GET SUMMARIES FOR LEFT AND RIGHT BAR SEGMENTS.
*ONE ITERATION OF THE MACRO LANGUAGE LOOP FOR EACH.
*****
%IF &LEFTBAR=2 %THEN %DO;

%DO I=1 %TO &RIGHTBAR;

    PROC FREQ DATA=SET1_&I;
    %IF &WEIGHT=1 %THEN %DO;
        WEIGHT &W_VAR;
    %END;
    TABLES CATEGORY*VAR&I / NOPRINT
        OUT=SET1_&I
    ;

    PROC FREQ DATA=SET1_&I;BY CATEGORY;
    TABLES VAR&I / NOPRINT
        OUT=SET2_&I
    ;

    WEIGHT COUNT;
```

INFORMATION VISUALIZATION

```

*PRINT SET2_&I FOR DIAGNOSTICS;
PROC PRINT DATA=SET2_&I;
TITLE "BAR SEGMENT FREQUENCIES BY CENTRAL CATEGORIES";

*KEEP REQUIRED VARIABLES IN SET2_&I;
DATA SET2_&I;SET SET2_&I;
KEEP CATEGORY VAR&I PERCENT;

*****;
*RESTRUCTURE SET2_&I BY TRANSPOSING BY CATEGORY. ;
*****;
PROC TRANSPOSE DATA=SET2_&I OUT=SET3_&I;BY CATEGORY;
ID VAR&I;

*****;
*MERGE SET3_&I WITH THE DATA SET ORDER SO THAT VAR&I
OBSERVATIONS CAN BE LINED UP IN THE SAME ORDER AS
CENTER OBSERVATIONS. ;
*****;

DATA SET3_&I;MERGE SET3_&I ORDER; BY CATEGORY;

*****;
*SORT SET3_&I TO PUT OBSERVATIONS IN ORDER FOR PRINTING. ;
*****;
PROC SORT DATA=SET3_&I;BY ORDER;

*****;
*PRINT THE DATA FOR INCLUSION AS A REPORT TABLE. ;
*****;
*PUT BAR SEGMENT LABELS INTO MACRO VARIABLES;
%LET TEMP1=LAB1_&I;
%LET TEMP2=LAB2_&I;
%LET TEMP3=LAB3_&I;
%LET TEMP4=LAB4_&I;
%LET LABEL1=%SUPERQ(&TEMP1);
%LET LABEL2=%SUPERQ(&TEMP2);
%LET LABEL3=%SUPERQ(&TEMP3);
%LET LABEL4=%SUPERQ(&TEMP4);

*****;
*SET UP INTERMEDIATE MACRO VARIABLES. ;
*****;
%LET TEMP=GROUPS_&I;
%LET TEMPT=TITLE&I;

*PRINT THE DATA;
PROC PRINT LABEL SPLIT=' ' DATA=SET3_&I;
VAR
%DO J=1 %TO %SUPERQ(&TEMP);
  %SUBSTR(&STRING,&J,1)
%END;
;ID CATEGORY;
FORMAT
%DO J=1 %TO %SUPERQ(&TEMP);
  %SUBSTR(&STRING,&J,1)
%END;
  5.1 CATEGORY &FMT...;
LABEL
CATEGORY="&TITLE3"
A="&LABEL1"
B="&LABEL2"
C="&LABEL3"
D="&LABEL4"
;
TITLE
"BREAKDOWN OF %SUPERQ(&TEMPT) BY %UPCASE(&TITLE3)";
TITLE2
"ROW PERCENTAGES ADD TO 100%";

*****;
*FORM AND SAVE THE SAS/GRAPH ANNOTATE DATA SET FOR SIDE &I.;
*****;
DATA ANNO&I; SET SET3_&I;

KEEP FUNCTION TEXT STYLE X Y XSYS YSYS POSITION SIZE LINE;

```

```

LENGTH TEXT $50 STYLE $8 POSITION $1 FUNCTION $8
ZIP2 $3 ZIP1 $3 ;

RETAIN XSYS '3' YSYS '3' Y 10 SIDE 0;

*ASSIGN VALUES TO HORIZONTAL MARGIN VARIABLES;
H0=50; H1=34; H2=66;
H3=3; H4=97;

*ASSIGN VALUES TO VERTICAL STARTING LOCATIONS;
VSTART1=8;
VSTART2=18;
VSTART3=22;

*SCALE IS THE LENGTH OF THE BAR INTERVAL;
SCALE=31;

*MAX IS THE FULL BAR LENGTH -- 100%;
MAX=100;

*DEFINE AN ARRAY FOR PERCENTAGE VARIABLES A, B, ETC.;
ARRAY L{%SUPERQ(&TEMP)}
%DO J=1 %TO %SUPERQ(&TEMP);
  %SUBSTR(&STRING,&J,1)
%END;
;

*SET MISSING PERCENTAGES TO 0;
DO J=1 TO %SUPERQ(&TEMP);
  IF L{J}=. THEN L{J}=0;
END;

*CREATE VARIABLES INDICATING SIDE & NUMBER OF BAR SEGMENTS;
SIDE=&I;
GROUPS=%SUPERQ(&TEMP);

*SET LEFT MARGIN OF BARS;
IF SIDE=1 THEN H=H3;
ELSE IF SIDE=2 THEN H=H2;
IF SIDE=1 THEN DO;
  IF GROUPS=3 THEN X1=4;
  IF GROUPS=4 THEN X1=H;
END;
ELSE IF SIDE=2 THEN DO;
  IF GROUPS=3 THEN X1=H + 1;
  IF GROUPS=4 THEN X1=H;
END;

*PRINT AXES, LABELS, ETC. WHEN _N_=1;
IF _N_=1 THEN DO;

*****;
*CREATE KEY BLOCKS FOR BAR SEGMENT FILL PATTERNS. ;
*****;
IF GROUPS=3 THEN DO;
  *THREE GROUPS;
  X=X1;Y=VSTART1;FUNCTION="MOVE";OUTPUT;
  X=X+7;Y=Y+3;LINE=0;STYLE="E"; FUNCTION="BAR";OUTPUT;
  X=X+4;Y=Y-3;FUNCTION="MOVE";OUTPUT;
  X=X+7;Y=Y+3;STYLE="L3X"; FUNCTION="BAR";OUTPUT;
  X=X+4;Y=Y-3;FUNCTION="MOVE";OUTPUT;
  X=X+7;Y=Y+3;STYLE="X3X"; FUNCTION="BAR";OUTPUT;
END;

ELSE IF GROUPS=4 THEN DO;
  *FOUR GROUPS;
  X=X1;Y=VSTART1;FUNCTION="MOVE";OUTPUT;
  X=X+5;Y=Y+3;LINE=0;STYLE="S"; FUNCTION="BAR";OUTPUT;
  X=X+4;Y=Y-3;FUNCTION="MOVE";OUTPUT;
  X=X+5;Y=Y+3;STYLE="X3X"; FUNCTION="BAR";OUTPUT;
  X=X+4;Y=Y-3;FUNCTION="MOVE";OUTPUT;
  X=X+5;Y=Y+3;STYLE="L3X"; FUNCTION="BAR";OUTPUT;
  X=X+4;Y=Y-3;FUNCTION="MOVE";OUTPUT;
  X=X+5;Y=Y+3;STYLE="E"; FUNCTION="BAR";OUTPUT;
END;

```

```
*****;
*CREATE LABELS FOR THE BAR SEGMENT FILL PATTERNS. ;
*****;
POSITION="5";STYLE="COMPLEX";SIZE=1;FUNCTION="LABEL";

IF GROUPS=3 THEN DO;
X=H+1+(7/2);Y=Y-7;TEXT="&LABEL3";OUTPUT;
X=H+1+7+4+(7/2);TEXT="&LABEL2";OUTPUT;
X=H+1+(2*(7+4))+(7/2);TEXT="&LABEL1";OUTPUT;
END;

ELSE IF GROUPS=4 THEN DO;
X=H+(5/2);Y=Y-7;TEXT="&LABEL1";OUTPUT;
X=H+5+4+(5/2);TEXT="&LABEL2";OUTPUT;
X=H+(2*(5+4))+(5/2);TEXT="&LABEL3";OUTPUT;
X=H+(3*(5+4))+(5/2);TEXT="&LABEL4";OUTPUT;
END;

*****;
*CREATE THE LABEL FOR THIS SIDE. ;
*****;
X=(H+(SCALE/2));Y=VSTART2;TEXT="%SUPERQ(&TEMP)";OUTPUT;

*****;
*CREATE SCALE LABELS FOR THE BARS. ;
*****;
Y=VSTART3;

IF SIDE=1 THEN DO;
*LEFT AXES;
DO I=H1 TO (H3+1) BY -6;
ZIP=((H1-I)/6)*20;
X=I;
SIZE=1;ZIP1=PUT(ZIP,3.);TEXT=ZIP1||"%";OUTPUT;
END;
END;

ELSE DO;
*RIGHT AXES;
DO I=H2 TO (H4+1) BY 6;
ZIP=((I-H2)/6)*20;
X=I;
SIZE=1;ZIP2=PUT(ZIP,3.);TEXT=ZIP2||"%";OUTPUT;
END;
END;

END;
*****;
*INITIALIZATIONS WITH _N_=1 ARE NOW COMPLETE. ;
*****;

*WRITE DIAGNOSTIC INFORMATION TO THE PROGRAM LOG FILE;
PUT _N_ = SIDE= ;

*****;
*DEFINE LEFT AND RIGHT MARGIN VARIABLES FOR BARS. ;
*****;
IF SIDE=1 THEN H=H3;ELSE IF SIDE=2 THEN H=H2;
IF SIDE=1 THEN HH=H1; ELSE IF SIDE=2 THEN HH=H4;

*****;
*STORE BAR SEGMENT LENGTH VARIABLE NAMES AS MACRO VALUES. ;
*****;

*CASE FOR BARS WITH THREE SEGMENTS;
%IF %SUPERQ(&TEMP)=3 %THEN %DO;
%IF &I=1 %THEN %DO;
%LET V1=C;
%LET V2=B;
%LET V3=0;
%END;
%ELSE
%IF &I=2 %THEN %DO;
%LET V1=A;
%LET V2=B;
%LET V3=0;
```

```
%END;
%END;

*CASE FOR BARS WITH FOUR SEGMENTS;
%IF %SUPERQ(&TEMP)=4 %THEN %DO;
%IF &I=1 %THEN %DO;
%LET V1=D;
%LET V2=C;
%LET V3=B;
%END;
%ELSE
%IF &I=2 %THEN %DO;
%LET V1=A;
%LET V2=B;
%LET V3=C;
%END;
%END;

*****;
*DRAW THE BARS BY SEGMENTS. ;
*****;

*CASE OF BARS WITH THREE SEGMENTS;
IF GROUPS=3 THEN DO;

*SCALE THE PERCENTAGE VALUES TO FIT THE BAR INTERVAL;
DO J=1 TO 3;
L{J}=ROUND(SCALE*L{J}/MAX);
END;

*SET THE VERTICAL LOCATION OF THE MIDDLE OF THE FIRST BAR;
IF ORDER=1 THEN Y=VSTART2;

*DRAW THE BAR SEGMENTS;
X=H;Y=Y+6;FUNCTION="MOVE";OUTPUT;
X=X+&V1;Y=Y+3;LINE=0;STYLE="E"; FUNCTION="BAR";OUTPUT;
Y=Y-3;FUNCTION="MOVE";OUTPUT;
X=X+&V2;Y=Y+3;STYLE="L3X"; FUNCTION="BAR";OUTPUT;
Y=Y-3;FUNCTION="MOVE";OUTPUT;
X=HH;Y=Y+3;STYLE="X3X"; FUNCTION="BAR";OUTPUT;
Y=Y-3;FUNCTION="MOVE";OUTPUT;
END;

*CASE OF BARS WITH FOUR SEGMENTS;
IF GROUPS=4 THEN DO;

*SCALE THE PERCENTAGE VALUES TO FIT THE BAR INTERVAL;
DO J=1 TO 4;
L{J}=ROUND(SCALE*L{J}/MAX);
* IF L{J} GT SCALE THEN L{J}=SCALE;
END;

*SET THE VERTICAL LOCATION OF THE MIDDLE OF THE FIRST BAR;
IF ORDER=1 THEN Y=VSTART2;

*DRAW THE BAR SEGMENTS;
X=H;Y=Y+6;FUNCTION="MOVE";OUTPUT;
X=X+&V1;Y=Y+3;LINE=0;STYLE="S"; FUNCTION="BAR";OUTPUT;
Y=Y-3;FUNCTION="MOVE";OUTPUT;
X=X+&V2;Y=Y+3;STYLE="X3X"; FUNCTION="BAR";OUTPUT;
Y=Y-3;FUNCTION="MOVE";OUTPUT;
IF (X+&V3)<HH THEN DO;
X=X+&V3;Y=Y+3;STYLE="L3X"; FUNCTION="BAR";OUTPUT;
Y=Y-3;FUNCTION="MOVE";OUTPUT;
X=HH;Y=Y+3;STYLE="E"; FUNCTION="BAR";OUTPUT;
Y=Y-3;FUNCTION="MOVE";OUTPUT;
END;
ELSE DO;
X=HH;Y=Y+3;STYLE="L3X"; FUNCTION="BAR";OUTPUT;
Y=Y-3;FUNCTION="MOVE";OUTPUT;
END;
END;

%END;
```

INFORMATION VISUALIZATION

```

*****;
*END OF THE MACRO LANGUAGE LOOP.
*****;

%END;

*****;
*ANNOTATE DATA SET OF LABELS FOR CENTER VARIABLE CATEGORIES;
*****;

DATA ANNO3;SET CENTER2 END=EOF;
KEEP FUNCTION TEXT STYLE
  X Y XSYS YSYS POSITION
  WHEN SIZE LINE;
LENGTH TEXT $50 STYLE $8 POSITION $1
  FUNCTION $8 WHEN $1
  ZIP2 $3 ZIP1 $3 ;
RETAIN XSYS '3' YSYS '3' Y 10;

*ASSIGN VALUES TO HORIZONTAL MARGIN VARIABLES;
H0=50; H1=34; H2=66;
H3=3; H4=97;

*ASSIGN VALUES TO VERTICAL STARTING LOCATIONS;
VSTART1=8;
VSTART2=18;
VSTART3=22;

*SCALE IS THE LENGTH OF THE BAR INTERVAL;
SCALE=31;

*SET VERTICAL LOCATION FOR FIRST LABEL;
IF ORDER=1 THEN Y=VSTART2;

*DEFINE THE CATEGORY LABEL;
X=50; Y=Y+6; FUNCTION="MOVE";OUTPUT;
FUNCTION="LABEL";POSITION="B";SIZE=1;STYLE="COMPLEX";
TEXT=PUT(CATEGORY,&FMT.)||" ("||PUT(SUM,COMMA9.)||")";
WHEN="A";OUTPUT;

%IF &RIGHTBAR=1 %THEN %DO;
PCT=ROUND(SCALE*PERCENT/&RIGHTMAX);
X=H2;FUNCTION="MOVE";OUTPUT;
X=X+PCT;Y=Y+3;LINE=0;STYLE="S"; FUNCTION="BAR";OUTPUT;
Y=Y-3;FUNCTION="LABEL";TEXT=PUT(PERCENT,6.1)||"%";
POSITION="C";STYLE="COMPLEX";OUTPUT;
%END;

IF EOF THEN DO;

*SUBTITLE FOR CENTER CATEGORY VARIABLE;
X=50;Y=Y+9;SIZE=1.25;TEXT="&TITLE4";POSITION="5";OUTPUT;

*SUBTITLE FOR GRAPH;
X=50;Y=Y+9;SIZE=1.5;TEXT="&SUBTITLE";POSITION="5";OUTPUT;

%IF &RIGHTBAR=1 %THEN %DO;
Y=VSTART3;
*RIGHT AXES;
DO I=H2 TO (H4-1) BY 6;
ZIP=((I-H2)/6)*20;
X=I;
SIZE=1;ZIP2=PUT(ZIP,3.);TEXT=ZIP2||"%";OUTPUT;
END;
Y=VSTART2;
POSITION="5";STYLE="COMPLEX";SIZE=1;FUNCTION="LABEL";
X=(H2+(SCALE/2));Y=VSTART2;TEXT="&TITLE2";OUTPUT;
%END;

END;

*****;
*LEFT ANNOTATE DATA SET FOR BOX & WHISKER DIAGRAM OPTION.
*****;
%IF &LEFTBAR=1 %THEN %DO;

```

```

*SORT IN PREPARATION FOR PROC UNIVARIATE;
PROC SORT DATA=CENTER1;BY CATEGORY;

*****;
*GET SUMMARIES FOR BOX & WHISKER DIAGRAMS.
*****;

PROC UNIVARIATE NOPRINT DATA=CENTER1;
  BY CATEGORY;
  VAR SALARY;
  OUTPUT OUT=BW1
  N=NUMBER
  MAX=MAX
  P5=P5
  P10=P10
  Q3=Q3
  MEDIAN=MEDIAN
  Q1=Q1
  P90=P90
  P95=P95
  MIN=MIN
  SUM=SUM
  ;

*****;
*MERGE BW1 WITH THE DATA SET ORDER SO THAT BOX & WHISKER
OBSERVATIONS CAN BE LINED UP IN THE SAME ORDER AS
CENTER OBSERVATIONS.
*****;

DATA BW1;MERGE BW1 ORDER; BY CATEGORY;

*****;
*SORT BW1 TO PUT OBSERVATIONS IN ORDER FOR PRINTING.
*****;
PROC SORT DATA=BW1;BY ORDER;

*****;
*PRINT THE DATA FOR INCLUSION AS A REPORT TABLE.
*****;
OPTIONS LS=110;

PROC PRINT LABEL SPLIT=" " DATA=BW1;
VAR P5 P10 Q1 MEDIAN Q3 P90 P95;ID CATEGORY;
FORMAT P5 P10 Q1 MEDIAN Q3 P90 P95 COMMA9.
  CATEGORY &FMT...;
LABEL
CATEGORY="&TITLE3"
P5='5th Percentile'
P10='10th Percentile'
Q1='25th Percentile'
MEDIAN='Median'
Q3='75th Percentile'
P90='90th Percentile'
P95='95th Percentile'
;
TITLE
"Percentiles of &bwlabel within &TITLE3 Categories";

RUN;

OPTIONS LS=80;

*****;
*BUILD THE ANNOTATE DATA SET.
*****;

DATA ANNO4;SET BW1 END=EOF;
KEEP FUNCTION TEXT STYLE
  X Y XSYS YSYS POSITION
  WHEN SIZE LINE;
LENGTH TEXT $50 STYLE $8 POSITION $1
  FUNCTION $8 WHEN $1
  ZIP2 $3 ZIP1 $3 ;
RETAIN XSYS '3' YSYS '3' Y 10;

```

```

*ASSIGN VALUES TO HORIZONTAL MARGIN VARIABLES;
H0=50; H1=34; H2=66;
H3=3; H4=97;

*ASSIGN VALUES TO VERTICAL STARTING LOCATIONS;
VSTART1=8;
VSTART2=18;
VSTART3=22;

*SCALE IS THE LENGTH OF THE BAR INTERVAL;
SCALE=31;

*SET VERTICAL LOCATION FOR FIRST LABEL;
IF ORDER=1 THEN Y=VSTART2;

*DEFINE AN ARRAY FOR BOX & WHISKER SUMMARIES;
ARRAY L{7} MIN P5 Q1 MEDIAN Q3 P95 MAX ;

*SCALE BOX & WHISKER SUMMARIES FOR GRAPHING;
DO I=1 TO 7;
  L{I}=ROUND(SCALE*L{I}/&LEFTMAX);
  IF L{I} GT SCALE THEN L{I}=SCALE;
  L{I}=SCALE-L{I}+H3;
END;

*DRAW THE BOX & WHISKER DIAGRAM;
X=P5;Y=Y+6;FUNCTION='MOVE';OUTPUT;
X=Q3;FUNCTION='DRAW';LINE=1;OUTPUT;
X=Q1;FUNCTION='MOVE';OUTPUT;
X=Q3;Y=Y+3;LINE=1;STYLE='E';
  FUNCTION='BAR';OUTPUT;
X=MEDIAN;Y=Y-2.5;FUNCTION='LABEL';POSITION='B';
  SIZE=.75;STYLE='SIMPLEX';
  TEXT='M';OUTPUT;
Y=Y-.5;
FUNCTION='MOVE';OUTPUT;
X=P95;FUNCTION='DRAW';LINE=1;OUTPUT;
IF P95=H3 THEN DO;
  X=H3-2;FUNCTION='SYMBOL';TEXT='*';STYLE='COMPLEX';
  SIZE=1;OUTPUT;
END;

*PROVIDE LABELS;
IF EOF THEN DO;

Y=VSTART2;
X=(H3+(SCALE/2));FUNCTION='LABEL';
  POSITION='5';STYLE='COMPLEX';
  SIZE=1;TEXT='&TITLE1';OUTPUT;
Y=Y-8;
  TEXT='BOX & WHISKER SUMMARIES';OUTPUT;

Y=VSTART3;
*LEFT AXES;
DO I=H1 TO (H3+1) BY -6;
  ZIP=((H1-I)*10/300)*&LEFTMAX;
  X=I-1;
  SIZE=1;TEXT=PUT(ZIP,COMMA7.);OUTPUT;
END;

END;

%END;

*****;
*SET THE ANNOTATE DATA SETS TOGETHER AND SAVE EXTERNALLY. ;
*****;

DATA b.&ANNO;
  SET
    %IF &LEFTBAR=2 %THEN %DO; ANNO1(IN=INL) %END;
    %IF &LEFTBAR=1 %THEN %DO; ANNO4(IN=INL) %END;
    %IF &RIGHTBAR=2 %THEN %DO; ANNO2(IN=INR) %END;
    ANNO3(IN=INS);
  LENGTH TYPE $6;

```

```

IF INL THEN TYPE='LEFT';
IF INR THEN TYPE='RIGHT';
IF INS THEN TYPE='CENTER';

*****;
*PRINT THE ANNOTATE DATA FOR DIAGNOSTICS. ;
*****;
PROC PRINT;
  TITLE
  'ANNOTATE DATA SET';
RUN;

%MEND ZITO;
*****;
*END OF MACRO ZITO. ;
*****;

```

REFERENCES

¹ 1992 *The Registered Nurse Population, Findings from the National Sample Survey of Registered Nurses, March 1992*, U.S. Department of Health & Human Services, Public Health Service, U.S. Government Printing Office, ISBN 0-16-042616-2.

² Tukey, J.W.. (1977), *Exploratory Data Analysis*, Reading, MA: Addison Wesley.

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

Four SAS software programs (FIGURE1.SAS, DS01.SAS, WING01.SAS and GRAPH.SAS) are streamed together into a single file named SESUG98.SAS for download from the Applied & Engineering Statistics Department server at George Mason University.

From your WEB browser, go to
<ftp://www.galaxy.gmu.edu/pub/software>
 and click on SESUG98.SAS

Via anonymous ftp, ftp to www.galaxy.gmu.edu and use
 Login: anonymous
 Password: "user name"

Then (noting that names are case sensitive) type
 cd pub/software
 get SESUG98.SAS
 bye

Donald T. Gantz, Ph.D.
 Applied & Engineering Statistics
 MS 4A7
 George Mason University
 Fairfax, VA 22030
 (703) 993-1695
dgantz@gmu.edu

Yan Yuan
 The Urban Institute
 2100 M Street, N.W.
 Washington, D.C. 20037
 (202)261-5227
yyuan@ui.urban.org