ED 280 856 TM 870 182

AUTHOR	Wise, Lauress L., II
TITLE	The PROC BRRVAR Procedure: Documentation. Technical Report No. 28.
INSTITUTION	American Institutes for Research in the Behavioral Sciences. Palo Alto, CA. Statistical Analysis Group in Education.
SPONS AGENCY	National Center for Education Statistics (ED), Washington, DC.
REPORT NO	AIR-87600-3-83-TR
PUB DATE	Mar 83
CONTRACT	300-78-0150
NOTE	48p.
PUB TYPE	Reports - Research/Technical (143) -- Computer Programs (101)
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	*Computer Software; Elementary Secondary Education; *Error of Measurement; Estimation (Mathematics); Higher Education; National Surveys; *Sampling; Statistical Analysis; Statistical Bias
IDENTIFIERS	*Balanced Repeated Replication; *Sampling Error; Statistical Analysis System

ABSTRACT

BRRVAR, which uses the Balanced Repeated Replication approach, was designed for use with the Statistical Analysis System (SAS). It was created for the National Center for Education Statistics, to enlarge their capacity to estimate and analyze sampling errors for statistics generated from educational surveys with complex sampling designs. BRRVAR requires that the primary sampling units be organized into pairs that reflect the actual sampling design. Common statistics are generated, including sums, means, standard deviations, covariances, correlations, and regression coefficients. For each statistic, an overall estimate is computed along with the estimate of the standard error of the statistic and the approximate 95% confidence bounds for the statistic. The appendices include examples of an input dataset, SAS statements and output, and a computer program listing. (Author/GDC)

Technical Report No. 28

The PROC BRRVAR Procedure: Documentation

Lauress L. Wise, II

Prepared by

SAGE
STATISTICAL ANALYSIS GROUP IN EDUCATION

For the

National Center for Education Statistics

American Institutes for Research

Box 1113, Palo Alto, California 94302

AIR-87600-3/83-TR

TECHNICAL REPORT 28

THE BRRVAR PROCEDURE: DOCUMENTATION

Submitted to the National Center for Education Statistics

by

Lauress L. Wise

Statistical Analysis Group in Education
American Institutes for Research
P. O. Box 1113
Palo Alto, California 94302

This work was done under Contract No. 300-78-0150 with the National Center for Education Statistics, Department of Health, Education, and Welfare. However, the content does not necessarily reflect the position or policy of either agency, and no official endoresment should be inferred.

March 1983

TABLE OF CONTENTS

ABSTRACT

The BRRVAR procedure provides estimates of the sampling error or statistics generated from complex samples. BRRVAR uses the Balanced Repeated Replication approach (BRR), which requires that the primary sampling units be organized into pairs that reflect the actual sampling design. Common statistics are generated, including sums, means, standard deviations, covariances, correlations, and regression coefficients. For each statistic, an overall estimate is computed along with the estimate of the standard error of the statistic and the approximate 95% confidence bounds for the statistic.

INTRODUCTION

BRRVAR is a user-created procedure for use with the Statistical Analysis System (SAS). It was created for the National Center for Education Statistics to enlarge their capacity to estimate and analyze sampling errors for statistics generated from educational surveys with complex sampling designs. This program drew heavily from an earlier SAS procedure, NASSVAR, designed by David Morganstern and implemented by Greg Binzer, both of Westat, Inc. The most significant enhancements in the current program are the "automatic" application of an appropriate design matrix for defining the replicate half-samples and the much wider range of statistics which can be requested simply.

The use of BRRVAR may require considerable preparation of the input data file. If the initial sampling design was created specifically for BRR analyses, it specified sampling strata from which exactly two units were randomly drawn. (If the sampling was multi-stage, e.g. schools were sampled and then students were sampled within school, only the first or primary stage is of imporance here.) In this case, all that is required is that the file contain one or more variables that can by used to identify the different strata and the different PSUs within each stratum. The file must be sorted by these variables prior to calling BRRVAR.

In most cases, the actual sample drawn does not match the BRR assumptions exactly. It is then necessary to create pairs of primary sampling units and sort the file so that these pairs are adjacent. In some instances, the number of primary sampling units may be quite large. This is true, for example, if some primary sampling units were sampled with certainty so that the secondary units within these primary sampling units have to be treated as if they were the primary units. In such cases, it may be desirable to define "pseudo-PSUs" which are actually clusters of PSUs. If several schools were sampled with certainty for example, and then some number of students were sampled from each school, it would be desirable to define two randomly equivalent clusters of students for each school and then treat these clusters as "pseudo-PSUs". Similarly, if a relatively large number of schools were selected from some sampling stratum, it would be desirable to organize these schools into two randomly equivalent clusters and then use these clusters as "pleudo-PSUs."

In addition to specifiyng the variables that identify primary sampling units, it is necessary to specify a case weight using the WEIGHT statement. It is assumed that such a weight will have been put on the file for data from the kind of complex samples for which BRRVAR estimates are required. In general, the sampling weight will be the inverse of the overall sampling probability for each case (particularly if sums are of interest). If no such weight exists, then a dummy weight must be created.

Sepcification

The following statements are used with the BRRAVAR procedure:

```
PROC BRRVAR options and parms;

VAR(IABLES) variable list;

WEIGHT weight variable;

STRATID variable list;

PSUID variable list.
```

The PROC BRRVAR statement must include certain required parameters as described below. The VARIABLES, WEIGHT, STRATID, and PSUID statements are also required. Their function and syntax is described below.

PROC BRRVAR Statement

The following options may be provided in the PROC BRRVAR statement:

- DATA = <SAS dataset>, names the SAS dataset to be used by PROC BRRVAR. If no input dataset is specified, BRRVAR uses the most recently created SAS dataset.
- SUMS = causes sums to be estimated for each variable in the VARIABLE list.
- MEAN = causes means to be estimated for each variable in the VARIABLE list.

- STD = causes standard deviations to be estimated for each variable in the variable list.

- COV = causes covariances to be estimated for each variable in the variable list. (Note: this option is not compatible with the CORR option; if both are specified, correlations take precedence and covariances will not be printed.).

- CORR = causes correlations to be estimated for each pair of variables in the VARIABLE list.

- REGR = causes regression coefficients and a multiple R square to be estimated for each regresson equation specified. The last NDEP variables (1 if NDEP is not specified are treated as criterion variables. The remaining variables are treated as predictors. A separate regression equation is estimated for each criterion, using all of the predictor variables for each equation. If CORR has been specified, standardized regresssion coefficients are computed, otherwise raw regression coefficients are computed.

- NOMISS = causes listwise deletion of missing values (i.e., if any variable is missing, the entire case is excluded from all statistics). If only univariate statistics are requested and NOMISS is not specified, missing values will be deleted separately for each variable. If any of the multivariate statistics are requested (COV, CORR, or REGR) NOMISS is automatically forced.

- DESIGN = causes the replicate half-sample design matrix to be printed. The design matrix contains one row for each PSU and one column for each half-sample. The entries tell whether the PSU associated with the row was included in the half-sample indicated by the column.

- DUMP = requests a printout of the half-sample estimates for each statistic. In the current version this dump is relatively unformatted, with no labelling of the columns or rows.

- NPSU = <number of PSUs>, specifies the number of PSUs in the sample design. This parameter is primarily for documentation purposes at present. The actual number of PSUs is computed and compared to the expected number specified here. In case of discrepancies, the acutal number is used.

- NREP = <number of replicates>, specifies the number of half-sample replicates to be used. If omitted, NREP is set to the value of NPSU. (If both NPSU and NREP are omitted, you are in some trouble.) In the current version, a maximum of 48 replicates can be created appropriately. If NREP is larger than this, the half-samples begin repeating the same design so that no new information is added.

- NDEP = <number of dependent variables>, this parameter must be omitted unless REGR is specified. It gives the number of dependent varialbes for which regressions are to be performed. The last NDEP variables in the variable list are taken as dependent variables. If REGR is specified, but NDEP is omitted, NDEP is set equal to 1.

VAR Statement

VAR variables;

The VAR statement lists the variables to be analyzed. All of the variables listed must be numeric. If omitted, all numeric variables in the input dataset will be analyzed.

STRATID Statement

STRATID variable(s);

The variables in the STRATID must be sufficient to uniquely identify the individual sampling stratum. The file must be sorted by the STRATID variables, otherwise execution will terminate. There should be exactly two PSUs within each sampling stratum. If only 1 is found, a "dummy" PSU with no cases is created for purposes of balancing the half-samples. If more than two PSU are found, the excess PSUs are ignored (after a warning message is printed).

PSUID Statement

PSUID variables;

The variables in the PSUID list must be sufficient to identify the individual PSUs within each stratum. The file must be sorted by the variables in the PSUID list within each stratum. As indicated above, there must be exactly two PSUs within each sampling stratum. If the sample did not follow a strict BRR design, it may be necessary to construct "pseudo strata" or "pseudo PSUs" in order to obtain appropriate results. Consult your local sampling statistician.

WEIGHT Statement

WEIGHT variable;

The WEIGHT statement identifies the variable that is to be used as a weight for each case. One and only one weight variable must be specified. If the sample is unweighted, then a dummy weight variable must first be created. (If the sample is unweighted, do you really need a complicated algorithm like this?)

DETAILS

Missing Values

BRRVAR's treatment of missing values depends upon whether any multivariate statistics are requested. If COV, CORR, or REGR are specified, listwise deletion of cases with missing data is always performed. If only univariate statistics are requested, BRRVAR deletes missing values separately for each variable, unless NOMISS is specified in the PROC BRRVAR statement. If NOMISS is specified, then listwise deletion of cases with any missing values is always performed.

Missing values are not allowed for the weight variable. If any missing values are encountered, the program will terminate on the presumption that the weight variable was incorrectly specified.

Printed Output

The printed output from BRRVAR includes the following:

- a Summary Report indicating the number of cases processed and the number with missing values,

- a design matrix (if requested) indicating which PSUs were included in each of the replicate half-samples,

- all requested univariate statistics, including the overall estimate, the estimated sampling error of the overall estimate, and the lower and upper 95% confidence bounds for the overall estimate,

- if requested, an estimated correlation coeficient is printed for each pair of variables in the variable list along with the estimated sampling error of each correlation coefficient and the lower and upper confidence bounds of each correlation coefficient,

- if requested, BRRVAR prints a relatively unformatted dump of the estimates generated from each of the replicate half-samples. The columns of this dump correspond to the half-samples and the rows to the statistics being estimated. The first column gives the overall estimate.

EXAMPLE

The attached example in APPENDIX A shows the JCL necessary to access BRRVAR at COMNET. In this example, a single stratum variable and a single PSU identifier are read. The input dataset contains five records for each PSU with three variables for each record. The data are already sorted by PSU within stratum.

PROC BRRVAR is called twice. In the first instance, simple univarite statistics are requested. The second request asks for both correlations and a regression run. In addition, the DESIGN and DUMP options are invoked to provide additional information.

The printout resulting from the sample dataset is shown following the listing of the input records.

REFERENCES

Casady, R. J. The estimation of variance components using balanced repeated replications. Proceedings of the Social Statistics Section, American Statistical Association, 1975, 352-357.

Plackett, R. L. & Burman, P. J. The design of optimum multifactorial experiments. Biometrika, 1946, 33, 305-325.

APPENDIX A

Examples of an Input Dataset, SAS Statements and SAS Output

EXAMPLE: Input Dataset

```
//X        JOB (ED,AIR),'SAGE-WYOUNG',PRTY=8,
//             MSGCLASS=P,NOTIFY=XEJQNY,MSGLEVEL=(1,1)
/*ROUTE PRINT R207
// EXEC XSAS,OPTIONS='S=72',PRINT=P
//WORK  DD UNIT=SYSDA,SPACE=(CYL,(5,5))
//SASLIB DD DSN=NCES.XEJQNY.SAGELIB,DISP=SHR
//SYSIN  DD *
DATA TEMP;
     INPUT STR 1 PSU $2
           X    3-4  Y  5-6  Z  7-8;

     WT=4.0;
CARDS;
1A 1 5 1
1A 2 2 2
1A 3 3 3
1A 4 4 4
1A 5 1 5
1B 3 6 7
1B 4 3 6
1B 5 4 5
1B 6 5 4
1B 7 2 3
2A 1 5 1
2A 2 2 2
2A 3 3 3
2A 4 4 4
2A 5 1 5
2B 3 6 9
2B 4 3 8
2B 5 4 7
2B 6 5 6
2B 7 2 5
3A 1 3 3
3A 2 9 4
3A 3 8 5
3A 4 1 6
3A 5 4 7
3B 3 7 5
3B 4 3 4
3B 5 4 3
3B 6 1 2
3B 7 3 1
4A 1 9 5
4A 212 6
4A 314 7
4A 418 8
4A 516 9
4B 313 7
4B 412 6
4B 519 5
4B 621 4
4B 716 3
;
```

EXAMPLE: SAS Statements

```

PROC BRRVAR NREP=8 SUM MEAN STD;
     VAR X Y Z;
     STRATID STR;
     PSUID PSU;
     WEIGHT WT;
;
PROC BRRVAR NPSU=9 NREP=8 MEAN SUM STD REGR NDEP=1 DESI
GN DUMP;
     VAR X Y Z;
     STRATID STR;
     PSUID PSU;
     WEIGHT WT;
PROC CORR;
     VAR X Y Z;
```

EXAMPLE: SAS Output

```
                              S T A T I S T I C A L   A N A L Y S I S   S Y S T E M     3:30 THURSDAY, MARCH 10, 1983   1
                                   PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES
                                   SUMMARY INFORMATION


        40 OBSERVATIONS READ
       160 WEIGHTED OBSERVATIONS READ

   MISSING VALUES DELETED SEPARATELY FOR EACH VAR

         9  REPLICATES IN   8 PSU PAIR DESIGN
                              S T A T I S T I C A L   A N A L Y S I S   S Y S T E M     3:30 THURSDAY, MARCH 10, 1983   2
                                   PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES
                                   UNIVARIATE STATISTICS
                                                              95% CONFIDENCE BOUNDS
STATISTIC  VARIABLE      WTD. N      ESTIMATE    STD. ERROR       LOWER       UPPER  LABEL

SUM        X              160.0      640.000       80.000      483.200     796.800

SUM        Y              160.0     1052.000       62.354      929.786    1174.214

SUM        Z              160.0      760.000      105.830      552.573     967.427

MEAN       X              160.0        4.000        0.500        3.020       4.980

MEAN       Y              160.0        6.575        0.390        5.811       7.339

MEAN       Z              160.0        4.750        0.661        3.454       6.046

STD        X              160.0        1.732        0.124        1.489       1.975

STD        Y              160.0        5.449        0.505        4.460       6.439

STD        Z              160.0        2.107        0.271        1.575       2.638
```

EXAMPLE: SAS Output (cont.)

```
                                  STATISTICAL ANALYSIS SYSTEM      3:30 THURSDAY, MARCH 10, 1983    3
                                        PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES
                                        HALF SAMPLE DESIGN MATRIX
PSU              MATRIX

     1 A       0 1 0 1 0 0 1 1
     1 B       1 0 1 0 1 1 0 0
     2 A       0 0 1 0 0 1 1 1
     2 B       1 1 0 1 1 0 0 0
     3 A       0 1 0 0 1 1 1 0
     3 B       1 0 1 1 0 0 0 1
     4 A       0 0 0 1 1 1 0 1
     4 B       1 1 1 0 0 0 1 0
       9 PSUS SPECIFIED
       8 PSUS FOUND
```

```
                                  STATISTICAL ANALYSIS SYSTEM      3:30 THURSDAY, MARCH 10, 1983    4
                                        PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES
                                        SUMMARY INFORMATION


        40 OBSERVATIONS READ
       160 WEIGHTED OBSERVATIONS READ

         0 WEIGHTED OBS DELETED FOR MISSING VALUES

         9  REPLICATES IN   8 PSU PAIR DESIGN
```

EXAMPLE: SAS Output (cont.)

```
                                    STATISTICAL ANALYSIS SYSTEM       3:30 THURSDAY, MARCH 10, 1983    5
                                          PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES
                                          UNIVARIATE STATISTICS
                                                               95% CONFIDENCE BOUNDS
STATISTIC  VARIABLE      WTD. N     ESTIMATE   STD. ERROR     LOWER        UPPER  LABEL

SUM        X              160.0      640.000      80.000      483.200      796.800

SUM        Y              160.0     1052.000      62.354      929.786     1174.214

SUM        Z              160.0      760.000     105.830      552.573      967.427

MEAN       X              160.0        4.000       0.500        3.020        4.980

MEAN       Y              160.0        6.575       0.390        5.811        7.339

MEAN       Z              160.0        4.750       0.661        3.454        6.046

STD        X              160.0        1.732       0.124        1.489        1.975

STD        Y              160.0        5.449       0.505        4.460        6.439

STD        Z              160.0        2.107       0.271        1.575        2.638
```

```
                                    STATISTICAL ANALYSIS SYSTEM       3:30 THURSDAY, MARCH 10, 1983    6
                                          PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES
                                          CORRELATION ESTIMATES
FIRST      SECOND     CORRELATION                               95% CONFIDENCE BOUNDS
VARIABLE   VARIABLE      WTD. N     ESTIMATE   STD. ERROR   LOWER        UPPER     PROB C>0

Y          X              160.0       0.026       0.260     -0.482        0.535
Z          X              160.0       0.069       0.351     -0.618        0.756
Z          Y              160.0       0.341       0.273     -0.194        0.877
```

EXAMPLE: SAS Output (cont.)

```
                                        STATISTICAL  ANALYSIS  SYSTEM      3:30 THURSDAY, MARCH 10, 1983
                                               PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES
                                               REGRESSION COEFFICIENTS
FIRST      SECOND     REGRESSION
                                                                 95% CONFIDENCE BOUNDS
VARIABLE   VARIABLE      WTD. N        ESTIMATE    STD. ERROR   LOWER        UPPER      PROB C>0

Z          X              160.0          0.033        0.410      -0.770        0.836
Z          Y              160.0          0.342        0.284      -0.216        0.899
MULT RSQ   Z              160.0          0.127        0.252      -0.366        0.621
```

```
                                        STATISTICAL  ANALYSIS  SYSTEM      3:30 THURSDAY, MARCH 10, 1983
                                               PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES
                                               REPLICATE STATISTICS
FIRST      SECOND     REPLICATE S
                                                                 95% CONFIDENCE BOUNDS
VARIABLE   VARIABLE      WTD. N        ESTIMATE    STD. ERROR   LOWER        UPPER      PROB C>0


    0.033  -0.662   0.042  -0.371   0.438   0.432   0.524   0.043   0.395
    0.342   0.048  -0.051   0.314   0.639   0.471   0.631   0.260   0.800
    0.127   0.442  -0.010   0.248   0.277   0.147   0.496   0.173   0.590
  160.000 160.000 160.000 160.000 160.000 160.000 160.000 160.000 160.000
    1.732   1.414   1.732   1.658   1.732   1.732   1.658   1.658   1.658
    5.449   5.783   5.886   5.925   4.928   4.766   4.924   6.030   5.052
    2.107   2.000   2.000   1.732   2.449   1.732   2.000   1.732   2.236
  160.000 160.000 160.000 160.000 160.000 160.000 160.000 160.000 160.000
    0.026  -0.043   0.284   0.153  -0.281  -0.315  -0.138   0.455  -0.146
    0.069  -0.707   0.289  -0.174   0.000   0.000   0.302   0.522   0.135
    0.341   0.048  -0.047   0.307   0.588   0.424   0.619   0.206   0.783
  640.000 800.000 640.000 720.000 640.000 640.000 560.000 560.000 560.000
 1052.0001112.0001128.0001072.000 976.0001072.0001032.0001088.000 936.000
  760.000 800.000 800.000 640.000 800.000 960.000 800.000 640.000 640.000
    0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
    0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
    0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
    0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
```

EXAMPLE: SAS Output (cont.)

```
                         S T A T I S T I C A L   A N A L Y S I S   S Y S T E M    3:30 THURSDAY, MARCH 10, 1983   5

VARIABLE    N          MEAN          STD DEV            SUM           MINIMUM         MAXIMUM

X          40    4.00000000       1.75411604   160.00000000        1.00000000      7.00000000
Y          40    6.57500000       5.51867459   263.00000000        1.00000000     21.00000000
Z          40    4.75000000       2.13337340   190.00000000        1.00000000      9.00000000

                    CORRELATION COEFFICIENTS / PROB > ^R^ UNDER HO:RHO=0 / N = 40
                                             X        Y        Z

                              X        1.00000  0.02649  0.06852
                                        0.0000   0.8711   0.6744
                              Y        0.02649  1.00000  0.34138
                                        0.8711   0.0000   0.0311
                              Z        0.06852  0.34138  1.00000
                                        0.6744   0.0311   0.0000
```

APPENDIX B

Program Listing

```
//X        JOB (ED.AIR).'D-WYOUNG'.PRTY=8.
//             MSGCLASS=P.NOTIFY=XEJQNY.MSGLEVEL=(1.1)
/*ROUTE PRINT R207
//WYYSTEP1 EXEC XPLIXCL.SYSOUT=P.
//   PARM.PLI='NM.ND.A.NEST.X.NC(S).MAR(2.72.1).NSYN(C).NOP.OF.C'.
//   PARM.LKED='MAP.XREF.LIST'.
//   COND.LKED=(4.LT,PLI)
//PLI.SYSIN  DD *
    MAIN:PROC OPTIONS(MAIN) :
         /*---------------------------------------------------*/
         /*                                                   */
         /*   BRRVAR : SAMPLING VARIANCE ESTIMATES USING      */
         /*            BALANCED REPEATED REPLICATIONS (BRR):  */
         /*                                                   */
         /*   PROC BRRVAR  < OPTIONS PARMS >:                 */
         /*        VAR     < VARIABLE LIST >:                 */
         /*        STRATID <STRATUM IDENTIFIER(S) >:          */
         /*        PSUID   < PSU IDENTIFIER(S) >:             */
         /*                NOTE: FILE MUST BE SORTED BY       */
         /*                      STRATUM AND PSU IDENTIFIERS: */
         /*        WEIGHT  < WEIGHT VARIABLE >:               */
         /*                NOTE: WEIGHT VARIABLE LIST MUST    */
         /*                CONTAIN EXACTLY 1 VARIABLE         */
         /*        ADJBY   < ADJ FILE IDENTIFIERS >;          */
         /*                - NOT YET IMPLEMENTED -            */
         /*                MUST BE A SUBSET OF THE PSU        */
         /*                IDENTIFIERS:                       */
         /*        COMPUTE < COMPUTED VARIABLES >:            */
         /*                - NOT YET FULLY IMPLEMENTED -      */
         /*                                                   */
         /*   OPTIONS:                                        */
         /*                                                   */
         /*      DESIGN-PRINT DESIGN MATRIX FOR HALF SAMPLES  */
         /*        DUMP-PRINT STATISTICS FOR EACH HALF SAMPLE */
         /*         SUM-ESTIMATE SUM OF EACH VARIABLE         */
         /*        MEAN-ESTIMATE MEAN OF EACH VARIABLE        */
         /*        STD -ESTIMATE  STD OF EACH VARIABLE        */
         /*        COV -ESTIMATE  COV MATRIX FOR THE VARIABLES*/
         /*        CORR-ESTIMATE  COR MATRIX FOR THE VARIABLES*/
         /*        REGR-PERFORM REGRESSION                    */
         /*             NOTE: (1) LAST VARIABLE IN VAR LIST   */
         /*                       IS THE DEPENDENT VARIABLE   */
         /*                   (2) IF CORR IS SPECIFIED.       */
         /*                       STANDARDIZED BETAS ARE      */
         /*                       ESTIMATED. OTHERWISE RAW    */
         /*                       REGRESSION COEFFICIENTS ARE */
         /*                       SHOWN;                      */
         /*                                                   */
         /*   PARAMETERS:                                     */
         /*                                                   */
         /*        NPSU-TOTAL NUMBER OF PSU'S IN THE SAMPLE   */
         /*                                                   */
         /*        NRED-(OPTIONAL) NUMBER OF REPLICATE HALF-  */
         /*             SAMPLES TO BE USED                    */
         /*             NOTE: IF OMITTED NREP IS              */
         /*                   MIN(100.NPSU/2)                 */
         /*        NDEP-NUMBER OF DEPENDENT VARIABLES IF      */
         /*             REGRESSION IS SPECIFIED               */
         /*             NOTE: THE LAST NDEP VARIABLES IN THE  */
/*                   VAR LIST ARE TAKEN AS DEPENDENT */
/*                   VARIABLES.  EACH IS REGRESSED   */
/*                   AGAINST THE PREDICTOR (NON-     */
/*                   DEPENDENT) VARIABLES.           */
/*                   DEFAULT=1.                      */
/*                                                   */
/*   EXTERNAL ROUTINES:                              */
/*                                                   */
/*        SETPARM                                    */
/*        ALLOC                                      */
/*        PROCESS                                    */
/*        PRINTIT                                    */
/*                                                   */
/*                                                   */
/*---------------------------------------------------*/
/*                                                   */
/*   DECLARE LINKAGE TO EXTERNAL SAS ROUTINES        */
/*                                                   */
DCL SASPLO ENTRY.
    UNUSED ENTRY(FIXED BIN(31)).
    BYPASS ENTRY(FIXED BIN (31)).
    SASLOG ENTRY(FIXED BIN(31),FIXED BIN(31)) :
/*                                                   */
/*   DECLARE EXTERNAL ROUTINES..                     */
/*                                                   */
DCL SETPARM ENTRY((*.*)FLOAT BIN(53)).
    ALLOC ENTRY((*.*)FLOAT BIN(53)),
     PRINTIT ENTRY((*.*) FLOAT BIN(53)).
    PROCESS ENTRY((*,*)FLOAT BIN(53)) :
/*                                                    */
/*   DECLARE EXTERNAL PROGRAM VARS..                  */
/*                                                    */
DCL (MEANFLG.
     STDFLG.
     COVFLG,
     CORRFLG.
     REGRFLG.
     SUMFLG,
     SSQFLG.
     NOMSFLG.
     DESFLG,
     DUMFLG.
     OUTFLG) BIT(1) EXTERNAL.
    (INPTR.
     PSUPTR.
     PSUPTR2.
     STRPTR.
     STRPTR2.
     OUTPTR) POINTER EXTERNAL.
    (NREP,
     NV1.
     NSTAT.
     NCOMP.
     NDEP,
     NPSUID.
     NSTRID.
     NADJID.
     NOBS.
     NMISS.NTOT) FIXED BIN(31) EXTERNAL.
     (WGTOBS.WGTMISS.NUMPSU) FLOAT BIN(53) EXTERNAL.
```

```
              ESTAB(1.1) FLOAT BIN(53) CONTROLLED :
         DCL I FIXED BIN(31) STATIC:
         /*                                                   */
         /*   MESSAGES...                                     */
         /*                                                   */
           DCL MSG1  CHAR(80) INIT(' VERSION 2A').
             MSG1A CHAR(80) INIT(' PREPARED FOR N.C.E.S.').
             MSG1B CHAR(80) INIT(' BY THE STATISTICAL ANANLSYS').
             MSG1C CHAR(80) INIT('    GROUP IN EDUCATION (SAGE)').
             MSG2 CHAR(80) INIT(' AMERICAN INSTITUTES FOR RESEARCH
').
             MSG3 CHAR(80) INIT(' P. O. BOX 1113').
             MSG4 CHAR(80) INIT(' PALO ALTO. CA 94302').
             MSG5 CHAR(80) INIT(' (415) 493-3550').
             MSGF1 FIXED BIN(31) BASED(ADDR(MSG1)).
             MSGF1A FIXED BIN(31) BASED(ADDR(MSG1A)).
             MSGF1B FIXED BIN(31) BASED(ADDR(MSG1B)).
             MSGF1C FIXED BIN(31) BASED(ADDR(MSG1C)).
             MSGF2 FIXED BIN(31) BASED(ADDR(MSG2)).
             MSGF3 FIXED BIN(31) BASED(ADDR(MSG3)).
             MSGF4 FIXED BIN(31) BASED(ADDR(MSG4)).
             MSGF5 FIXED BIN(31) BASED(ADDR(MSG5)):
         /*                                                   */
         /*   CREATE SAS ENVIRONMENT.. THEN ROCK & ROLL..     */
         /*                                                   */
         CALL SASPLO :
         ALLOCATE ESTAB :
         CALL SETPARM(ESTAB):
         CALL ALLOC(ESTAB);
         CALL PROCESS(ESTAB):
         CALL PRINTIT(ESTAB);
         /*                                                   */
         /*   PRINT MESSAGES AND HALT..                       */
         /*                                                   */
         CALL BYPASS(I);
         IF I=0 THEN DO;
            CALL UNUSED(-1) :           /*   PRINT CORE MSG..   *
/
            CALL SASLOG(MSGF1.80) :
            CALL SASLOG(MSGF1A.80) ;
            CALL SASLOG(MSGF1B.80) :
            CALL SASLOG(MSGF1C.80) :
            CALL SASLOG(MSGF2.80) :
            CALL SASLOG(MSGF3.80) :
            CALL SASLOG(MSGF4.80) :
            CALL SASLOG(MSGF5.80) :
            CALL SASLOG(MSGF1.80) :
         END:
  END MAIN :
 /*MEM2*/
/*
//LKED.SYSLMOD DD DSN=NCES.XEJQNY.SAGELIB.DISP=OLD
//LKED.SYSLIB DD DSN=SYS2.PLIXLINK.DISP=SHR
//            DD DSN=SYS2.PLIXBASE.DISP=SHR
//            DD DSN=SYS2.PLIXTASK.DISP=SHR
//            DD DSN=NCES.XEJQNY.SAGELIB.DISP=SHR
//            DD DSN=SYS2.SAS.V796.SUBLIB.DISP=SHR
//LKED.SYSIN  DD *
 ENTRY  ENTRY
 INCLUDE SYSLIB(SASPLO.PROCESS.ALLOC.SETPARM)
```

```
01780  NAME BRRVAR2(R)
01790 /*
END OF DATA
```

```
//X        JOB (ED.AIR).'D-WYOUNG'.PRTY=8.
//             MSGCLASS=P,NOTIFY=XEJQNY.MSGLEVEL=(1,1)
/*ROUTE PRINT R207
//WYYSTEP1 EXEC XPLIXCL.SYSOUT=F,
//   PARM.PLI='NM.ND.A.NEST.X,NC(S).MAR(2.72.1).NSYN(C).NOP.OF.C'.
//   PARM.LKED='MAP.XREF.LIST,NCAL'.
//   COND.LKED=(4.LT.PLI)
//PLI.SYSIN  DD *
  ALLOC:PROC(ESTAB) :
         /*--------------------------------------------------*/
         /*                                                  */
         /*  ALLOC-DETERMINES AMOUNT OF CORE REUQIRED FOR   */
         /*         NASSVAR DATA STRUCTURES..                */
         /*                                                  */
         /*         ALLOCATES REQUIRED CORE                  */
         /*                                                  */
         /*         ZEROS CORE ALLOCATED                     */
         /*                                                  */
         /*         EXTERNAL FUNCTION 'ALLCTRN' IS INVOKED   */
         /*         TO DETERMINE IF COMPVAR AND OUTVAR       */
         /*         STATEMENTS ARE USED..  'ALLCTRN'         */
         /*         TAKES CARE OF ALLOCATING CORE FOR THESE  */
         /*         STRUCTURES.. 'ALLCTRN' RETURNS '1'B IF   */
         /*         THE COMPVAR/OUTVAR CONSTRUCTION IS USED.. */
         /*                                                  */
         /*---- ---------------------------------------------*/
         /*                                                  */
         /*      CLARE EXTERNAL LINKAGE TO SAS..             */
         /*                                                  */
         DCL NL 'A  ENTRY(FIXED BIN(31))
                 RETURNS(FIXED BIN(31)).
             GETMEM ENTRY(FIXED BIN(31).FIXED BIN(31).PTR.
                          FIXED BIN(31)),
             ZERO ENTRY(PTR.FIXED BIN(31)),
             MEMERR ENTRY(FIXED BIN(31)).
             LODOUT ENTRY:
         /*                                                  */
         /*   EXTERNAL VARIABLES...                          */
         /*                                                  */
         DCL (INPTR.PSUPTR,PSUPTR2.STRPTR.STRPTR2.OUTPTR)
             PTR EXTERNAL.
             (COVFLG.CORRFLG.REGRFLG.OUTFLG.SSCFLG.NOMSFLG)
             BIT(1) EXTERNAL.
             (MEANFLG.STDFLG.SUMFLG) BIT(1) EXTERNAL.
             (NSTAT.NREP.NV1.NPSUID.NADJID.NSTRID)
             FIXED BIN(31) EXTERNAL.
              ESTAB(*.*) FLOAT BIN(53) CONTROLLED :
         /*                                                  */
         /*   LOCALS..                                       */
         /*                                                  */
         DCL (NV.NV2.NREPL.NEED.GOT) FIXED BIN(31).
              WORKVEC(1) FLOAT BIN(53) BASED(INPTR).
              PTR PTR.
              NULL BUILTIN:
         /*                                                  */
         /*  FREE ESTIMATE TAB... PATCH TO RUN UNDER THE     */
         /*  OPTIMIZER WHICH REQUIRES THAT A CONTROLLED      */
         /*  STRUCTURE MUST BE ALLOCATED IN ORDER TO BE      */
         /*  PASSED.. (NOT THE CASE WITH THE 'F' COMPILER..  */
/*  AT THIS POINT WE HAVE NO IDEA WHAT DIMENSIONS    */
/*  THIS TABLE MUST HAVE.... WILL ALLOCATE IT LATER  */
/*  IN THIS MODULE....                               */
/*                                                   */
FREE ESTAB :
/*   LOAD SAS OUTPUT MODS IF OUTPUT SPECIFIED BEFORE */
/*   ALLOCATING ANY CORE(OUTMODS NEED ABOUT 1K..)    */
/*                                                   */
 OUTFLG='0'B:   /* NOT YET IMPLEMENTED */
IF OUTFLG THEN CALL LODOUT :
/*                                                  */
/*     NEED=DOUBLE WORDS REQUIRED FOR TEMPORY       */
/*          STORAGE                                 */
/*                                                  */
/*         NV1-FOR INPUT VARS                       */
/*      NPSUID-FOR PSU ID VARS                      */
/*      NPSUID-FOR PRIOR PSU ID                     */
/*      NSTRID-FOR STRAT ID VARS                    */
/*      NSTRID-FOR PRIOR STRAT ID                   */
/*                                                  */
 NEED=8*NV1+16*NPSUID+16*NADJID+16*NV1:
 CALL GETMEM(NEED.GOT,INPTR.0);
 IF NEED < GOT THEN DO:
    CALL MEMERR(0):
    STOP:
 END:
 PSUPTR=ADDR(WORKVEC(NV1+1)):
 PSUPTR2=ADDR(WORKVEC(NV1+NPSUID+1)):
 STRPTR=ADDR(WORKVEC(NV1+2*NPSUID+1)):
 STRPTR2=ADDR(WORKVEC(NV1+2*NPSUID+NSTRID+1)):

/*                                                   */
/*   ZERO REGION RETURNED...                         */
/*                                                   */
CALL ZERO(INPTR.NEED) :
/*                                                  */
/*    NSTAT=DOUBLE WORDS REQUIRED FOR EACH          */
/*          REPLICATE HALF SAMPLE:                  */
/*                NV1=NV+1                          */
/*                NV2=NV*NV1/2                      */
/*            NV1 - FOR RECORD COUNTS (IF \NOMISS)  */
/*            NV1 - FOR WEIGHTED SUMS (IF \NOMISS)  */
/*      NV1+NCOMP - FOR SUMS/MEANS                  */
/*            NV1 - FOR SSQS/STDS                   */
/*            NV2 - FOR SSCP/COV/CORR               */
/*                                                  */
NV=NV1-1:
IF COVFLG ^  CORRFLG ^  REGRFLG THEN DO:
   SSQFLG='1'B:
   NV2=NV1*NV/2:
END:
ELSE DO:
   SSQFLG='0'B:
   NV2=0:
END:
NSTAT=2*NV1+NV2:
IF NOMSFLG THEN NSTAT=NSTAT+4:
           ELSE NSTAT=NSTAT+2*NV1:
OUTPTR=NULL:
/*         OUTPUT NOT SUPPORTED                     */
```

```
         /*                                                    */
         /*   ALLOCATE 2 DIM TABLE TO HOLD ESTIMATES..         */
         /*                                                    */
         NREP1=NREP+1:
         NEED=8*NREP1*NSTAT:
         CALL GETMEM(-1,GOT,PTR,0) :
         IF NEED > GOT THEN DO :
            CALL MEMERR(NEED-GOT) :
            STOP :
         END :
         ALLOCATE ESTAB(NREP1.NSTAT):
  END ALLOC :
 /*MEM4*/
/*
//LKED.SYSLMOD DD DSN=NCES.XEJQNY.SAGELIB.DISP=SHR
//LKED.SYSLIB DD
//             DD
//             DD
//             DD
//             DD
//             DD DSN=NCES.XEJQNY.SAGELIB.DISP=SHR
//             DD DSN=SYS2.SAS.V796.SUBLIB.DISP=SHR
//             DD DSN=SYS2.SAS.R795.GLIBRARY.DISP=SHR
//             DD DSN=SYS2.PLIXBASE.DISP=SHR
//LKED.SYSIN   DD *
 NAME ALLOC(R)
/*
END OF DATA
```

```
//X        JOB (ED,AIR),'D-WYOUNG',PRTY=8,
//             MSGCLASS=P,NOTIFY=XEJQNY,MSGLEVEL=(1,1)
/*ROUTE PRINT R207
//WYYSTEP1 EXEC XPLIXCL,SYSOUT=P,
//   PARM.PLI='NM,ND,A,NEST,X,NC(S),MAR(2,72,1),NSYN(C),NOP,OF,C',
//   PARM.LKED='MAP,XREF,LIST,NCAL',
//   COND.LKED=(4,LT,PLI)
//PLI.SYSIN  DD *
  SETPARM:PROC(ESTAB) ;
        /*--------------------------------------------------*/
        /*                                                  */
        /*-  SETPARM-DISCOVERS PROCEDURE OPTIONS AND PARMS  */
        /*         VALIDATES SAMPLE DESIGN..                */
        /*                                                  */
        /*--------------------------------------------------*/
        /*                                                  */
        /*   DECLARE EXTERNAL LINKAGES...                   */
        /*                                                  */
        DCL IOPT ENTRY(FIXED BIN(31))
                 RETURNS(FIXED BIN(31)),
            PARM ENTRY(FIXED BIN(31)) RETURNS(FLOAT BIN(53)),
            ERROR ENTRY(FIXED BIN(31), *,*) FLOAT BIN(53)),
            OBSERR ENTRY,
            NOVAR ENTRY(FIXED BIN(31)) RETURNS(FIXED BIN(31)) ;
        /*                                                  */
        /*   DECLARE EXTERNAL PROGRAM VARIABLES...          */
        /*                                                  */
        DCL (NREP,NV1,NDEP,NPSUID,NSTRID,NADJID,NCOMP)
            FIXED BIN(31) EXTERNAL,
             NUMPSU FLOAT BIN(53) EXTERNAL,
             ESTAB(*,*) FLOAT BIN(53) CONTROLLED,
            (SSQFLG,CORRFLG,MEANFLG,STDFLG,COVFLG,REGRFLG,SUMFLG,
             DESFLG,DUMFLG,NOMSFLG) BIT(1) EXTERNAL;
        /*                                                  */
        /*   ERROR MESSAGES..                               */
        /*                                                  */
        DCL ERR1 CHAR(80) INIT(
            ' ERROR:  PSU IDENTIFIER(S) NOT SPECIFIED.'),
            ERR2 CHAR(80) INIT(
            ' ERROR:  ADJ BY LIST NOT COMPATIBLE WITH PSU ID LIST.
').
            ERR3 CHAR(80) INIT(
            ' ERROR:  NO WEIGHT VARIABLES SPECIFIED.'),
            ERR4 CHAR(80) INIT(
            ' ERROR:  NUMPSU PARM MISSING OR INVALID.'),
            ERR5 CHAR(80) INIT(
            ' ERROR:  MORE THAN 64 REPLICATES SPECIFIED.'),
            ERR6 CHAR(80) INIT(
            ' ERROR:   STRATUM IDENTIFIER(S) NOT SPECIFIED.'),
            ERR7 CHAR(80) INIT(
            ' ERROR:   INVALID SPECIFICATION FOR NDEP.'),
            ERR1F FIXED BIN(31) BASED(ADDR(ERR1)),
            ERR2F FIXED BIN(31) BASED(ADDR(ERR2)),
            ERR3F FIXED BIN(31) BASED(ADDR(ERR3)),
            ERR4F FIXED BIN(31) BASED(ADDR(ERR4)),
            ERR5F FIXED BIN(31) BASED(ADDR(ERR5)),
            ERR6F FIXED BIN(31) BASED(ADDR(ERR6)),
            ERR7F FIXED BIN(31) BASED(ADDR(ERR7));
        /*                                                  */
/*   DISCOVER PROCEDURE OPTIONS..                       */
/*                                                      */
IF IOPT(1)=1 THEN MEANFLG='1'B :
ELSE MEANFLG='0'B ;
IF IOPT(2)=1 THEN STDFLG='1'B :
ELSE STDFLG='0'B ;
IF IOPT(3)=1 THEN COVFLG='1'B :
ELSE COVFLG='0'B ;
IF IOPT(4)=1 THEN CORRFLG='1'B ;
ELSE CORRFLG='0'B ;
IF IOPT(5)=1 THEN REGRFLG='1'B :
ELSE REGRFLG='0'B :
IF IOPT(6)=1 THEN SUMFLG='1'B;
ELSE SUMFLG='0'B;
IF IOPT(7)=1 THEN DESFLG='1'B;
ELSE DESFLG='0'B;
IF IOPT(8)=1 THEN DUMFLG='1'B ;
ELSE DUMFLG='0'B :
IF IOPT(9)=1 THEN NOMSFLG='1'B;
ELSE NOMSFLG='0'B;
/*                                                      */
/*   INSURE COMPATIBILITY OF OPTIONS & SET DEFAULTS */
/*                                                      */
IF \(MEANFLG^STDFLG^COVFLG^CORRFLG^REGRFLG)
   THEN SUMFLG='1'B;
IF COVFLG^CORRFLG^REGRFLG THEN SSQFLG='1'B;
                          ELSE SSQFLG='0'B;
IF SSQFLG THEN NOMSFLG='1'B:
/*   DETERMINE NUMBER OF VARS ON EACH LIST             */
NPSUID=NOVAR(3);
IF NPSUID < 1 THEN DO:

   CALL ERROR(ERR1F,ESTAB);
END:
NADJID=NOVAR(4):
IF NADJID > NPSUID ^ NADJID > 0
   THEN DO:                /* RATIO ADJUSTMENTS */
                            /* NOT YET HERE     */
     CALL ERROR(ERR2F,ESTAB);
   END:
   NSTRID=NOVAR(5);
   IF NSTRID < 1 THEN DO:
      CALL ERROR(ERR6F,ESTAB);
   END:
/*                                                      */
/*   DETERMINE NUMBER VARS ON WEIGHT STATEMENT..        */
/*                                                      */
IF   NOVAR(6) \= 1 THEN DO :
   CALL ERROR(ERR3F,ESTAB) :
END :
/* DETERMINE NO OF COMPUTED VARS                        */
NCOMP=0;
NCOMP=NOVAR(7):

/*                                                      */
/*                     SET    NUMPSU                    */
/*                                                      */
NUMPSU=0.00;
NUMPSU=PARM(1);
/*                                                      */
```

```
        /*                      CHECK    NREP                    */
        /*                                                       */
        NREP=0.0;
        NREP=PARM(2);
        IF NREP < 1 THEN NREP=(NUMPSU/2+1);
           ELSE NREP=NREP+1;
        NV1=NOVAR(1)+1;
        /*                     CHECK    NDEP                       */
        NDEP=0;
        NDEP=PARM(3);
        IF REGRFLG THEN IF NDEP < 1 THEN NDEP=1;
        IF NDEP >= NOVAR(1) ^ (\REGRFLG & NDEP > 0) THEN DO;
           CALL ERROR(ERR7F,ESTAB);
        END;
  END SETPARM ;
 /*MEM3*/
/*
//LKED.SYSLMOD DD DSN=NCES.XEJGNY.SAGELIB,DISP=OLD
//LKED.SYSLIB DD
//            DD
//            DD
//            DD
//            DD
//            DD DSN=NCES.XEJGNY.SAGELIB,DISP=SHR
//            DD DSN=SYS2.SAS.V796.SUBLIB,DISP=SHR
//            DD DSN=SYS2.SAS.R795.GLIBRARY,DISP=SHR
//            DD DSN=SYS2.PLIXBASE,DISP=SHR
//LKED.SYSIN  DD *
 NAME SETPARM(R)
/*
END OF DATA
```

```
//X        JOB (ED.AIR).'D-WYOUNG'.PRTY=8,
//             MSGCLASS=P,NOTIFY=XEJQNY.MSGLEVEL=(1,1)
/*ROUTE PRINT R207
//WYYSTEP1 EXEC XPLIXCL.SYSOUT=P.
//   PARM.PLI='NM,ND,A,NEST,X,NC(S).MAR(2,72,1),NSYN(C).NOP,OF,C'.
//   PARM.LKED='MAP.XREF,LIST,NCAL',
//   COND.LKED=(4,LT,PLI)
//PLI.SYSIN  DD *
  PROCESS:PROC(ESTAB) :
          /*----------------------------------------------------*/
          /*                                                    */
          /*   PROCESS-COMPUTES ESTIMATES OF CHARACTERISTICS    */
          /*           SPECIFIED FOR EACH REPLICATE IN THE      */
          /*           SAMPLE DESIGN..                          */
          /*                                                    */
          /*           WRITES REPLICATE LEVEL RECORDS TO OUTPUT */
          /*           DS IF THE OUTPUT OPTION IS IN EFFECT..   */
          /*                                                    */
          /*           COMPILES AND EXECUTES PROGRAMMING        */
          /*           STATEMENTS FOLLOWING THE PROC IF         */
          /*           THE OUTVAR/COMPVAR CONSTRUCTION          */
          /*           IS USED..                                */
          /*                                                    */
          /*           NATIONAL ESTIMATES ARE STORED IN         */
          /*           ARRAY ESTAB(NREPS.VARNO)..               */
          /*                                                    */
          /*----------------------------------------------------*/
          /*                                                    */
          /*   EXTERNALS..                                      */
          /*                                                    */
          DCL (OUTFLG,OUTVFLG,SSQFLG,NOMSFLG,DESFLG,DUMFLG)
              BIT(1) EXTERNAL,
              (OUTPTR,INPTR,STRPTR,STRPTR2,
               PSUPTR,PSUPTR2) PTR EXTERNAL,
              (NOBS,NV1,NSTAT,NREP,NPSUID,NMISS,NSTRID)
              FIXED BIN(31) EXTERNAL,
               (WGTOBS, NUMPSU,WGTMISS) FLOAT BIN(53) EXTERNAL ;
          /*                                                    */
          /*   SAS LINKAGE..                                    */
          /*                                                    */
          DCL INPUT ENTRY RETURNS(FIXED BIN(31)),
              VARX ENTRY(FIXED BIN(31),FLOAT BIN(53)),
              OBSPTR ENTRY RETURNS(PTR),
              SETDSN ENTRY(FIXED BIN(31)),
              PUTOUT ENTRY(FIXED BIN(31),PTR),
              FBUF ENTRY,
              BYPASS ENTRY(FIXED BIN(31)),
              CLSOUT ENTRY,
              IOPT  ENTRY(FIXED BIN(31))
                    RETURNS(FIXED BIN(31)),
              LINSIZ ENTRY(FIXED BIN(31)),
              STITLE ENTRY(FIXED BIN(31),FIXED BIN(31)),
              NAMEV ENTRY(FIXED BIN(31),FIXED BIN(31),
                          FIXED BIN(15),),
              ERROR ENTRY(FIXED BIN(31),(*,*)FLOAT BIN(53)),
              MISSING ENTRY(FLOAT BIN(53)) RETURNS(BIT(1)):
          /*                                                    */
          /*  LOCALS..                                          */
          /*                                                    */
```

```
                DCL ESTAB(*,*) FLOAT BIN(53) CONTROLLED,
                    WGT FLOAT BIN(53) STATIC,
                    NPSU FLOAT BIN(53) STATIC INIT(0),
                    INHOLD(1) FLOAT BIN(53) BASED(INPTR),
                    BYPSU(1)  FLOAT BIN(53) BASED(PSUPTR),
                    BYPSU2(1) FLOAT BIN(53) BASED(PSUPTR2),
                    BYCHPSU(1)  CHAR(8)     BASED(PSUPTR),
                    BYCHPS2(1)  CHAR(8)     BASED(PSUPTR2),
                    BYSTR(1) FLOAT BIN (53) BASED(STRPTR),
                    BYSTR2(1) FLOAT BIN(53) BASED(STRPTR2),
                    BYCHAR2(64) BIT(1) STATIC,
                    BYCHSTR(1) CHAR(8) BASED(STRPTR),
                    BYCHST2(1) CHAR(8) BASED(STRPTR2),
                    BYCHAR(64)  BIT(1) STATIC,
                    OUTVEC FIXED BIN(31) BASED(OUTPTR),
                   (OPTR,IPTR) PTR,
                   (ADDR,SUBSTR) BUILTIN,
                   (IR,IV,K,NV,KV,IR1,JV,L,LNSZ,CENTER) FIXED BIN(31) ST
ATIC;
                DCL DESIGN(16)  CHAR(8)  STATIC INIT(
                    /* 8*/  '1110100X',
                    /*16*/  '11110101', '1001000X',
                    /*24*/  '11111010', '11001100', '1010000X',
                    /*32*/  '00001010', '11101100', '01111100', '1101001
X',
                    /*48*/  '11111011', '11001010', '11100100', '1101100
0',
                            '10101100', '0010000X' );
                DCL MXREPS(5)   FIXED BIN(31) STATIC INIT(8, 16, 24, 32,
 48),
                    REPIND(5)   FIXED BIN(31) STATIC INIT(1,  2,  4,  7,
 11),
                    (X, WT, XWT,REPLID) FLOAT BIN(53) STATIC,
                    DESVEC      CHAR(64)  BASED(DESPTR),
                    (STRATNO, ISINDX, IRINDX) FIXED BIN(31) STATIC,
                    (LASTPSU, TESTPSU)        CHAR(1)       STATIC,
                    DESERR BIT(1) STATIC,
                    DESPTR    POINTER,
                    FLAT_EST(1) FLOAT BIN(53),
                    1 NAMESTR,
                    2 NTYPE    FIXED BIN(15),
                    2 NPOS     FIXED BIN(15),
                    2 NLNG     FIXED BIN(15),
                    2 NVAR0    FIXED BIN(15),
                    2 NNAME    CHAR(8),
                    2 NLABEL   CHAR(40),
                    2 NFORM    CHAR(8),
                    2 NIFORM   CHAR(8),
                    2 NFL      FIXED BIN(15),
                    2 NFD      FIXED BIN(15),
                    2 NF       FIXED BIN(15),
                    2 NJUST    FIXED BIN(15) ;
                /*                                                     */
                /*   ERROR MESSAGES..                                  */
                /*                                                     */
                DCL ERR1 CHAR(80) INIT(
                    ' ERROR:  MISSING VALUE FOR WEIGHT INVALID.'),
                    ERR1F FIXED BIN(31) BASED(ADDR(ERR1));
                DCL ERR2  CHAR(80) STATIC INIT(
                    ' ERROR:  FILE NOT SORTED BY STRAT & PSU ID VARIABLE
S.').
             ERR2F FIXED BIN(31) BASED(ADDR(ERR2));
         ON ENDPAGE(SYSPRINT) BEGIN :
            CALL STITLE(0,L) :
            PUT EDIT('PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES'
)
                    (SKIP(1).COL(CENTER).A) :
            PUT SKIP (2) ;
         END;

         /*  SET LINESIZE AND CENTER                      */
         /*                                                        */
         CALL LINSIZ(LNSZ);
         CENTER=(LNSZ-35)/2;
         IF IOPT(50) = 0 THEN CENTER=1:

         /*   IF OUTPUT SPECIFIED DEFINE OUTPUT DS & VARS..     */
         /*                                                        */
         IF OUTFLG THEN CALL PREPOUT :
            /*                                                     */
            /*    INITIALIZE VALUES                                */
            /*                                                     */
             DO IV=1 TO NPSUID:
                CALL NAMEV(3,IV.NTYPE.NAMESTR):
                IF NTYPE=2 THEN
                   BYCHAR(IV)='1'B;
                ELSE BYCHAR(IV)='0'B;
             END;
            DO IV=1 TO NSTRID;
               CALL NAMEV(5,IV,NTYPE,NAMESTR);
               IF NTYPE=2 THEN BYCHAR2(IV)='1'B;
                          ELSE BYCHAR2(IV)='0'B;
            END;
            LASTPSU='1'B;
            NOBS=0; WGTOBS=0;
            DO IR=1 TO NREP+1  ;
               DO IV=1 TO NSTAT;
                  ESTAB(IR.IV)=0.0  ;
               END ;
            END  ;
            NV=NV1-1;
            IR1=NREP+1;
         STRATNO=0;
         LASTPSU='1';
         DO IR = 1 TO 5;
            IF NREP <= MXREPS(IR)+1 THEN GO TO SETDES:
         END;
         IR=5;
SETDES:
         IV=REPIND(IR);
         DESPTR=ADDR(DESIGN(IV));
         ISINDX=1;
         IF DESFLG THEN DO;
            SIGNAL ENDPAGE(SYSPRINT);
            PUT EDIT('HALF SAMPLE DESIGN MATRIX','PSU','MATRIX')
                    (COL(CENTER),A.SKIP(2),A.COL(20),A) :
         END;
         PUT SKIP;

            /*                                                     */
            /*   STEP THRU OBS & COMPUTE ESTIMATES AT THE      */
            /*   REPLICATE LEVEL...                            */
            /*                                                 */
            DO WHILE(INPUT=0) :
               CALL VARX(5. BYSTR(1)):
               CALL VARX(3. BYPSU(1)):

               IF NOBS = 0 THEN GO TO NEWSTR:
            /*                  CHECK FOR NEXT STRATUM          */
            DO IV=1 TO NSTRID:
               IF BYCHAR2(IV) THEN DO:
                  IF BYCHSTR(IV) > BYCHST2(IV)
                     THEN GO TO NEWSTR:
                     ELSE IF BYCHSTR(IV) < BYCHST2(IV)
                          THEN CALL ERROR(ERR2F.ESTAB):
                   END:
                   ELSE DO;
                     IF BYSTR(IV) > BYSTR2(IV)
                        THEN GO TO NEWSTR;
                        ELSE IF BYSTR(IV) < BYSTR2(IV)
                             THEN CALL ERROR(ERR2F.ESTAB):
                    END:
                 END:
            /*                CHECK FOR NEXT PSU                */
               DO IV = 1 TO NPSUID;
                 IF BYCHAR(IV) THEN DO:
                    IF BYCHPSU(IV) > BYCHPS2(IV) THEN GO TO NEWPS
U:
          ELSE IF BYCHPSU(IV)<BYCHPS2(IV) THEN CALL ERROR(ERR2F.E
STAB):
                 END:
                 ELSE IF BYPSU(IV) > BYPSU2(IV) THEN GO TO NEWPSU
:
            ELSE IF BYPSU(IV)<BYPSU2(IV) THEN CALL ERROR(ERR2F.ES
TAB):
               END:
      GO TO PROCOBS;
NEWSTR:
            IF NOBS > 0 THEN CALL ADDSUMS;
            IF LASTPSU='0' THEN DO;
               CALL PRNTSTR:
               PUT EDIT(' DUMMY PSU CREATED ')(A);
            END:
            STRATNO=STRATNO+1;
            ISINDX=ISINDX+1:
            IF SUBSTR(DESVEC,ISINDX,1)='X' THEN ISINDX=1;
            DESERR='0'B:
            LASTPSU='0';
            DO IV=1 TO NSTRID:
               BYSTR2(IV)=BYSTR(IV):
            END;
            GO TO SETPSU:
NEWPSU:
            CALL ADDSUMS:
            IF LASTPSU='0' THEN LASTPSU='1';
            ELSE DESERR='1'B:
SETPSU:
            NPSU=NPSU+1;
            DO IV=1 TO NPSUID;
               BYPSU2(IV)=BYPSU(IV);
            END:
            IF DESFLG^DESERR THEN DO:
               CALL PRNTSTR:
               CALL PRNTPSU:
            END;
            IF DESERR THEN
               PUT EDIT(' MORE THAN 2 PSUS. THIS ONE SKIPPED')(A)
:
PROCOBS:
               CALL VARX(6.WGT):
               IF MISSING(WGT) THEN DO:
                  CALL ERROR(ERR1F.ESTAB):
               END;
               NOBS=NOBS+1;  WGTOBS=WGTOBS+WGT:
            IF DESERR THEN GO TO ENDCASE;
               CALL VARX(1.INHOLD(1)):
               IF NOMSFLG THEN DO IV=1 TO NV;
                  IF MISSING(INHOLD(IV)) THEN DO:
                     NMISS=NMISS+1:
                     WGTMISS=WGTMISS+WGT:
                     GO TO ENDCASE:
                   END:
               END;

                    IR=NREP+1:
                    KV=0:
                    IF \NOMSFLG THEN KV=2*NV1:
                    K=2*NV1+KV:
                    DO IV=1 TO NV1;
                       IF IV = NV1 THEN X=1.0;  ELSE X=INHOLD(IV):
                       KV=KV+1;
                       IF \NOMSFLG THEN DO:
                          IF MISSING(X) THEN GO TO ENDVAR:
                          ESTAB(IR.IV)=ESTAB(IR,IV)+1.0;
                          ESTAB(IR.IV+NV1)=ESTAB(IR.IV+NV1)+WGT:
                       END;
                       XWT=X*WGT:
                       ESTAB(IR,KV)=ESTAB(IR.KV)+XWT;
                       ESTAB(IR.KV+NV1)=ESTAB(IR.KV+NV1)+X*XWT:
                       IF IV > 1 & SSQFLG = '1'B THEN DO JV=1 TO I
V-1:
                          K=K+1:
                          ESTAB(IR,K)=ESTAB(IR,K)+XWT*INHOLD(JV):
                       END:
ENDVAR:             END;
ENDCASE:     END:

         CALL ADDSUMS:
         PUT SKIP EDIT(NUMPSU,' PSUS SPECIFIED',
                        NPSU.' PSUS FOUND')(F(10).A.SKIP.F(10).A):
         NUMPSU=NPSU;
         /*                                                     */
         /*   IF OUTPUT OPT IN EFFECT CLOSE THE DS..            */
         /*                                                     */
         IF OUTFLG THEN DO :
            CALL SETDSN(2) :
            CALL FBUF :
            CALL CLSOUT :
         END ;
         /*                                                     */
```

```
    /*                                                  */
PREPOUT:PROC :
        /*----------------------------------------------*/
        /*                                              */
        /*   PREPOUT-INTERNAL PROC TO DEFINE OUTPUT DS  */
        /*           & ALL OUTPUT VARIABLES..           */
        /*           OUTPUT DS WILL CONTAIN THE         */
        /*           FOLLOWING:                         */
        /*                                              */
        /*           1. REPLICATE ID   REPLID           */
        /*           2. ANY BY VARS    DEF ON INPUT     */
        /*           3. REPLICATE ESTS DEF ON INPUT     */
        /*                                              */
        /*----------------------------------------------*/
        /*                                              */
        /*   VAR DEF STRUCT..                           */
        /*                                              */
        DCL 1 NAMESTR,
            2 NTYPE   FIXED BIN(15),
            2 NPOS    FIXED BIN(15),
            2 NLNG    FIXED BIN(15),
            2 NVARO   FIXED BIN(15),
            2 NNAME   CHAR(6).
            2 NLABEL  CHAR(40).
            2 NFORM   CHAR(8).
            2 NIFORM  CHAR(8),
            2 NFL     FIXED BIN(15),
            2 NFD     FIXED BIN(15),
            2 NF      FIXED BIN(15),
            2 NJUST   FIXED BIN(15) :
        /*                                              */
        /*   SAS LINKAGE..                              */
        /*                                              */
        DCL OPNOUT ENTRY,
            GETDS ENTRY(PTR),
            HISTRY ENTRY(PTR),
            NAMEV ENTRY(FIXED BIN(31),FIXED BIN(31),
                        FIXED BIN(15),),
            ONAME2 ENTRY(FIXED BIN(15),FIXED BIN(31)),
            ONAMES ENTRY(FIXED BIN(15),FIXED BIN(31),
                         FLOAT BIN(53)),
            ENDNAM ENTRY,
            DLTOUT ENTRY,
            ABUF ENTRY ;
        /*                                              */
        /*  OUTVAR STRUCTURE...                         */
        /*                                              */
        DCL 1 OUTSTR(1) BASED(OUTPTR).
            2 OUTADDR  PTR,
            2 VARADDR  PTR,
            2 OUTNAME  CHAR(8) :
       /*                                               */
       /*   DEFINE OUTPUT DS..                          */
       /*                                               */
       CALL SETDSN(2) :            /* OUTPUT IS CURRENT.  */
       CALL OPNOUT :               /* OPEN OUTPUT DS..    */
       CALL SETDSN(1) :            /* INPUT IS CURRENT..  */
       CALL FBUF :                 /*  FREE INPUT BUFFER  */
       CALL GETDS(IPTR) :          /*  ADDRESS INPUT..    */
       CALL SETDSN(2) :            /*  OUTPUT IS CURRENT  */
```

```
            CALL HISTRY(IPTR) :          /* GET DS HISTORY..    */
            /*                                                  */
            /*   DEFINE REPLICATE IDENTIFIER..                  */
            /*                                                  */
            NNAME='REPL_ID' ;
            NLABEL='REPLICATE IDENTIFIER' ;
            NFORM,NIFORM=' ' ;
            NTYPE=1 ;
            NLNG=8 ;
            NFL,NFD= 0 ;
            CALL ONAMES(NTYPE,OUTVEC,REPLID) ;
            /*                                                  */
            /*   DEFINE VARS ON VARLIST..                       */
            /*                                                  */
            DO IV=1 TO NV1 ;
                    CALL SETDSN(1) ;
                    CALL NAMEV(1,IV,NTYPE,NAMESTR) ;
                    NLNG=8 ;
                    CALL SETDSN(2) ;
                    CALL ONAMES(NTYPE,OUTVEC,FLAT_EST(IV)) ;
            END ;
            /*                                                  */
            /*   DEFINE COMPUTED VARS..                         */
            /*                                                  */
            IF OUTVFLG THEN DO IV=1 TO NSTAT;
                       NNAME=OUTNAME(IV) ;
                       NLABEL='COMPUTED ESTIMATE' ;
                       NLNG=8 ;
                       NTYPE=1 ;
                       NFL,NFD=0 ;
                       NFORM,NFORM=' ' ;
                       CALL ONAMES(NTYPE,OUTVEC,FLAT_EST(IV+NV1))
 ;
            END ;
            /*                                                 */
            /*   TERMINATE DS & OUTPUT VAR DEFINITION..        */
            /*   DELETE OUTPUT ROUTINES...                     */
            /*                                                 */
            CALL ENDNAM ;
            CALL DLTOUT ;
            CALL ABUF ;
            CALL SETDSN(1) ;
            CALL ABUF ;
     END PREPOUT ;
ADDSUMS: PROC;
     /*     INTERNAL PROC TO ADD PSU SUMS      */
     /*     TO HALF-SAMPLE SUMS                */
            IR1=NREP+1;
            WGT=1.0;
               IRINDX=ISINDX;
               DO IR=1 TO NREP;
                  IF IR > 2 THEN DO;
                     IRINDX=IRINDX+1;
                     TESTPSU=SUBSTR(DESVEC,IRINDX,1);
                     IF TESTPSU = 'X' THEN DO;
                        IRINDX=1;  TESTPSU=SUBSTR(DESVEC,1,1);
                     END;
                     IF DESFLG THEN IF LASTPSU=TESTPSU
                        THEN PUT EDIT('0 ')(A);
                        ELSE PUT EDIT('1 ')(A);
```

```
                      IF LASTPSU=TESTPSU THEN GO TO ENDREP;
                    END;
                    /*  SECOND REP INCLUDES LAST OF EACH PAIR */
                    ELSE IF IR = 2 THEN DO;
                      IF DESFLG THEN PUT EDIT(LASTPSU,' ')(A,A);
                      WGT=2.0*WGT;  /* DOUBLE WEIGHT FOR HALF SAMP
LES */
                      IF LASTPSU = '0' THEN GO TO ENDREP;
                     END;
                     DO IV=1 TO NSTAT;
                        ESTAB(IR,IV)=ESTAB(IR,IV)+
                                     WGT*ESTAB(IR1,IV);
                     END;
 ENDREP:        END;
     DO IV=1 TO NSTAT;
        ESTAB(IR1,IV)=0.0;
     END;
 END ADDSUMS;
 PRNTSTR: PROC;
      /*     INTERNAL PROC TO PRINT STRATUN ID       */
             PUT SKIP;
             DO IV=1 TO NSTRID;
                IF BYCHAR2(IV) THEN
                   PUT EDIT (BYCHST2(IV),' ')(A,A);
                   ELSE PUT EDIT (BYSTR2(IV),' ')(F(8),A);
             END;
 END PRNTSTR;

 PRNTPSU: PROC;
      /*     INTERNAL PROC TO PRINT PSU ID      */
             DO IV=1 TO NPSUID;
                IF BYCHAR(IV)
                   THEN PUT EDIT(BYCHPS2(IV),' ')(A,A);
                   ELSE PUT EDIT(BYPSU2(IV),' ')(F(8),A);
             END;
 END PRNTPSU;
  END PROCESS ;
 /*MEM6*/
/*
//LKED.SYSLMOD DD DSN=NCES.XEJQNY.SAGELIB,DISP=SHR
//LKED.SYSLIB DD
//            DD
//            DD
//            DD
//            DD
//            DD DSN=NCES.XEJQNY.SAGELIB,DISP=SHR
//            DD DSN=SYS2.SAS.V796.SUBLIB,DISP=SHR
//            DD DSN=SYS2.SAS.R795.GLIBRARY,DISP=SHR
//LKED.SYSIN  DD *
 NAME PROCESS(R)
/*
END OF DATA
```

```
//X        JOB (ED.AIR),'D-WYOUNG',PRTY=8,
//             MSGCLASS=P,NOTIFY=XEJQNY,MSGLEVEL=(1,1)
/*ROUTE  PRINT R207
//WYYSTEP1 EXEC XPLIXCL,SYSOUT=P,
//   PARM.PLI='NM,ND,A,NEST,X,NC(S),MAR(2,72,1),NSYN(C),NOP,OF,C',
//   PARM.LKED='MAP,XREF,LIST,NCAL',
//   COND.LKED=(4,LT,PLI)
//PLI.SYSIN  DD *
  PRINTIT:PROC(ESTAB) ;
          /*                                                    */
          /*  EXTERNALS..                                       */
          /*                                                    */
          DCL (NV1,NOUT,NWGT,NOBS,NREP,NDEP,NMISS,NSTAT)
              FIXED BIN(31) EXTERNAL,
              (NUMPSU,WGTOBS,WGTMISS) FLOAT BIN(53) EXTERNAL,
              (MISSPTR,MWGTPTR) PTR EXTERNAL,
              (SUMFLG,MEANFLG,STDFLG,SSQFLG,COVFLG,CORRFLG,REGRFLG
-
               NOMSFLG,DUMFLG) BIT(1) EXTERNAL ;
          /*                                                    */
          /*  INTERNAL STRUCTURES..                             */
          /*                                                    */
          DCL ESTAB(*,*) FLOAT BIN(53) CONTROLLED,
              1 NAMESTR,
              2 NTYPE    FIXED BIN(15),
              2 NPOS     FIXED BIN(15),
              2 NLNG     FIXED BIN(15),
              2 NVARO    FIXED BIN(15),
              2 NNAME    CHAR(8),
              2 NLABEL   CHAR(40),
              2 NFORM    CHAR(8),
              2 NIFORM   CHAR(8),
              2 NFL      FIXED BIN(15),
              2 NFD      FIXED BIN(15),
              2 NF       FIXED BIN(15),
              2 NJUST    FIXED BIN(15) ;
           DCL SYSPRINT FILE PRINT;
          /*                                                    */
          /*   DECLARE LINKAGE TO SAS...                        */
          /*                                                    */
          DCL NAMEV ENTRY(FIXED BIN(31),FIXED BIN(31),
                          FIXED BIN(15),),
              BEST  ENTRY(FLOAT BIN(53),FIXED BIN(31),
                          FLOAT BIN(53)),
              IOPT  ENTRY(FIXED BIN(31))
                    RETURNS(FIXED BIN(31)),
              LINSIZ ENTRY(FIXED BIN(31)),
              STITLE ENTRY(FIXED BIN(31),FIXED BIN(31)),
              ERROR ENTRY(FIXED BIN(31),(*,*)FLOAT BIN(53)) ;
          /*                                                    */
          /*  LOCALS..                                          */
          /*                                                    */
          DCL (I,IR,IV,JV,K,NV,L,CENTER,LNSZ,IER,KV,KV1,II,NP,NP1,
IR1)
              FIXED BIN(31),
               (ADDR,SUBSTR,SQRT) BUILTIN,
               (EST,SE,LB,UB,XNR,XM,XN,XS) FLOAT BIN(53),
               FCHAR CHAR(10) ALIGNED,
               DBEST FLOAT BIN(53) BASED(ADDR(FCHAR)),
     STATS(10) FLOAT BIN(53):
/*                                                   */
/*   ERROR MESSAGES..                                */
/*                                                   */
DCL SUBTIT  CHAR(32) STATIC INIT('SUMMARY INFORMATION'):
DCL STATLBL CHAR(10) STATIC:

DCL ERR1 CHAR(80) INIT(
    ' ERROR:  LINE SIZE MUST BE > 132 IN PRESENT VERSION
.').
    ERR1F FIXED BIN(31) BASED(ADDR(ERR1)):
/*                                                   */
/*   DEFINE ENDPAGE CONDITION.. '                    */
/*                                                   */
ON ENDPAGE(SYSPRINT) BEGIN :
   CALL STITLE(0,L) :
   PUT EDIT('PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES'
)
           (SKIP(1).COL(CENTER).A) :
   PUT SKIP EDIT(SUBTIT)(COL(CENTER).A):
   PUT SKIP (2) :
   IF SUBSTR(SUBTIT.1.1) = 'U' THEN DO:
      PUT EDIT('95% CONFIDENCE BOUNDS', 'STATISTIC'.
               'VARIABLE', 'WTD. N',
          'ESTIMATE'. 'STD. ERROR'. 'LOWER'. 'UPPER'. 'L
ABEL')
          (COL(68).A,SKIP,A,COL(12),A,COL(26),A.COL(38).
A,
            COL(50).A.COL(67).A.COL(80).A.COL(87).A):
  END;
   ELSE IF SUBSTR(SUBTIT.1.1) = 'C' THEN DO:
      PUT EDIT('FIRST'. 'SECOND'. SUBSTR(SUBTIT.1.11),
          '95% CONFIDENCE BOUNDS'. 'VARIABLE'. 'VARIABLE
'.
          'WTD. N'.'ESTIMATE'.'STD. ERROR'.'LOWER '.'UPP
ER',
          'PROB C>0')
          (A. COL(12), A. COL(22), A, COL(68). A. SKIP.
           A. COL(12). A, COL(26). A. COL(39). A. COL(51
), A.
           COL(64). A. COL(77). A.COL(87).A):
   END:
   ELSE IF SUBSTR(SUBTIT.1,1) = 'R' THEN DO:
      PUT EDIT('FIRST', 'SECOND', SUBSTR(SUBTIT,1,11),
          '95% CONFIDENCE BOUNDS', 'VARIABLE', 'VARIABLE
',
          'WTD. N','ESTIMATE'.'STD. ERROR'.'LOWER '.'UPP
ER',
          'PROB C>0')
          (A, COL(12). A, COL(22), A, COL(68). A. SKIP.
           A, COL(12). A, COL(26). A, COL(39). A. COL(51
), A.
           COL(64). A. COL(77), A.COL(87).A):
   END:
   PUT SKIP:
END;

/*  SET LINESIZE AND CENTER                  */
CALL LINSIZ(LNSZ);
IF LNSZ < 80 THEN CALL ERROR(ERR1F. ESTAB):
```

```
CENTER=(LNSZ-35)/2;
 IF IOPT(50) = 0 THEN CENTER=1;

/*  PRINT SUMMARY REPORT                   */
 SIGNAL ENDPAGE(SYSPRINT);
 PUT SKIP EDIT(NOBS.' OBSERVATIONS READ')
     (F(12),A);
 PUT SKIP EDIT(WGTOBS.' WEIGHTED OBSERVATIONS READ')
     (F(12.1), A);
 PUT SKIP;
 IF NOMSFLG THEN DO;
    WGTOBS=WGTOBS-ESTAB(1.NV1);
    PUT SKIP EDIT(WGTOBS.
        ' WEIGHTED OBS DELETED FOR MISSING VALUES')(F(12.
1).A);
 END;
 ELSE PUT SKIP EDIT('   MISSING VALUES DELETED SEPARATELY
 ',
               'FOR EACH VAR')(A.A);
 PUT SKIP;
 PUT SKIP(2) EDIT(NREP.' REPLICATES IN'.NUMPSU.
     'PSU PAIR DESIGN')(F(12),X(1).A,F(4).X(1).A);

/*  COMPUTE AND PRINT SUMS                */
SUBTIT = 'UNIVARIATE STATISTICS';
NV=NV1-1;  XNR=NREP-1;
IF NOMSFLG THEN KV1=0;
   ELSE KV1=2*NV1;
IF SUMFLG THEN DO;
   SIGNAL ENDPAGE(SYSPRINT);
   STATLBL='SUM';
   XN=ESTAB(1.NV1);
   KV=KV1;
   DO IV=1 TO NV;
      KV=KV+1;
      IF \NOMSFLG THEN XN=ESTAB(1.IV+NV1);
      CALL PRINTLN(KV.IV.XN);
      PUT EDIT(NLABEL)(COL(74).A);
   END;
END;

/*  COMPUTE MEANS                     */
DO IR=1 TO NREP;
   KV=KV1;
   XN=ESTAB(IR. NV1);
   DO IV=1 TO NV;
      KV=KV+1;
      IF \NOMSFLG THEN XN=ESTAB(IR.IV+NV1);
      ESTAB(IR. KV)=ESTAB(IR.KV)/XN;
   END;
END;

IF MEANFLG THEN DO;
   STATLBL='MEAN';
   XN=ESTAB(1.NV1);
   KV=KV1;
   DO IV=1 TO NV;
      KV=KV+1;
      IF \NOMSFLG THEN XN=ESTAB(1.IV+NV1);
      CALL PRINTLN(KV.IV,XN);
      PUT EDIT(NLABEL)(COL(87).A):
   END:
END:

/* COMPUTE S. D.S                         */
DO IR=1 TO NREP:
   XN=ESTAB(IR.NV1);
   KV=KV1:
   DO IV=1 TO NV;
      KV=KV+1:
      IF \NOMSFLG THEN XN=ESTAB(1.IV+NV1):
      SE=ESTAB(IR.KV+NV1)/XN-ESTAB(IR,KV)**2;
      IF SE > 0 THEN SE=SQRT(SE):
      ESTAB(IR.KV+NV1)=SE:
   END:
END:

IF STDFLG THEN DO:
   STATLBL='STD':
   XN=ESTAB(1.NV1):
   KV=KV1:
   DO IV=1 TO NV:
      KV=KV+1:
      IF \NOMSFLG THEN XN=ESTAB(1.IV+NV1):
      CALL PRINTLN(KV+NV1.IV.XN):
      PUT EDIT(NLABEL)(COL(74).A):
   END;
END:

/* COMPUTE COVARIANCES AND CORRELATIONS */
IF SSQFLG & NV > 1
   THEN DO IR=1 TO NREP:
   K=2*NV1;
   DO IV=2 TO NV:
      XM=ESTAB(IR,IV):  XN=ESTAB(IR,NV1);
      DO JV=1 TO IV-1:
         K=K+1;
         ESTAB(IR,K)=(ESTAB(IR.K)/XN-ESTAB(IR.JV)*XM):
      END:
   END:
END;

IF CORRFLG THEN DO:
   DO IR=1 TO NREP;
      K=2*NV1;
      DO IV=2 TO NV;
         XS=ESTAB(IR.IV+NV1):
         DO JV=1 TO IV-1;
            K=K+1:
            ESTAB(IR,K)=ESTAB(IR,K)
                 /(XS*ESTAB(IR.JV+NV1));
         END:
      END;
   END:
END;

 IF COVFLG^CORRFLG THEN DO:
    IF CORRFLG THEN
            SUBTIT='CORRELATION ESTIMATES':
       ELSE SUBTIT='COVARIANCE ESTIMATES':
```

```
       XN=ESTAB(1.NV1);
   SIGNAL ENDPAGE(SYSPRINT);
   K=2*NV1:
      DO IV=2 TO NV:
         CALL NAMEV(1.IV.NTYPE.NAMESTR):
         STATLBL=NNAME;
         DO JV=1 TO IV-1:
            K=K+1:
            CALL PRINTLN(K.JV.XN):
         END:
      END;
END:

DCL CPTR.PTR.   K2 FIXED BIN(31).
    C(1) FLOAT BIN(53) BASED(CPTR):
IF REGRFLG THEN DO:
   SUBTIT='REGRESSION COEFFICIENTS';
   SIGNAL ENDPAGE(SYSPRINT):
   IR1=NREP+1:
   CPTR=ADDR(ESTAB(IR1.1)):
   NP=NV1-NDEP-1:
   NP1=NP+1:
   DO II=1 TO NDEP:
      KV=NP+II:
      DO IR=1 TO NREP;
      K=2*NV1:
      K2=0;
        DO IV=1 TO NP:
         IF IV > 1 THEN
            DO JV=1 TO IV-1:
               K=K+1:    K2=K2+1:
               C(K2)=ESTAB(IR.K);
            END;
            K2=K2+1;
            IF CORRFLG THEN C(K2)=1.0;
               ELSE C(K2)=ESTAB(IR,IV+NV1)**2;
         END:
         K=(KV-1)*(KV-2)/2+2*NV1:
         DO IV=1 TO NP:
            K=K+1:     K2=K2+1:
            C(K2)=ESTAB(IR,K):
         END:
         K2=K2+1;
         IF CORRFLG THEN XS=1.0:
            ELSE XS=ESTAB(IR,II+NP+NV1)**2:
            C(K2)=XS:
         CALL REGRLT(C,NP1,IER):
         K=0:
         K2=NP*(NP+1)/2;
         DO IV=1 TO NP:
            K=K+1;     K2=K2+1:
            ESTAB(IR,K)=C(K2);
         END:
         K=K+1;     K2=K2+1:
         ESTAB(IR.K)=(XS-C(K2))/XS:
     END;
         CALL NAMEV(1.KV.NTYPE.NAMESTR):
         STATLBL=NNAME;
         DO IV=1 TO NP:
            CALL PRINTLN(IV,IV.XN);
            END:
             STATLBL='MULT RSQ':
             CALL PRINTLN(NP1.KV.XN):
       END:
      END:

    /*    PRINT ESTIMATES FOR EACH HALF SAMPLE                         */
    /*                                                                  */
    IF DUMFLG THEN DO:
    SUBTIT='REPLICATE STATISTICS':
    SIGNAL ENDPAGE(SYSPRINT):
   DO IV=1 TO NSTAT:
      PUT SKIP:
       DO IR=1 TO NREP:
          PUT EDIT(ESTAB(IR.IV))(F(8.3)):
       END;
      END:
    END:

 PRINTLN: PROC(K.IV.XN);
        /*  SUBROUTINE TO SET UP AND PRINT A SINGLE LINE */
        /*  K POINTS TO THE ESTIMATE IN ESTAB           */
       DCL (K,IV)   FIXED BIN(31):
       DCL XN       FLOAT BIN(53);

        CALL NAMEV(1.IV.NTYPE.NAMESTR):
        EST=ESTAB(1,K):
        CALL SAMPERR(ESTAB. K. XNR. SE):
        LB=EST-1.96*SE:   UB=EST+1.96*SE:
        PUT SKIP EDIT(STATLBL. NNAME. XN. EST. SE. LB. UB. NLABE
L)
           (A. COL(12). A. COL(20). F(13.1).F(13.3).F(13.3).
            F(13,3). F(13.3). A):
 END PRINTLN:

 SAMPERR: PROC(ESTAB. II, XNR. SE):
        /*  INTERNAL SUBROUTINE TO COMPUTE THE SAMPLING STD ERRO
R*/
       DCL ESTAB(*, *)  FLOAT BIN(53)  CONTROLLED;
        DCL (SE, XNR)     FLOAT BIN(53),
            (IR. II)     FIXED BIN(31),
            NREP        FIXED BIN(31) EXTERNAL;

       SE=0.0:
        EST=ESTAB(1,II);
        DO IR=2 TO NREP:
           SE=SE+(ESTAB(IR,II)-EST)**2:
        END:
        IF SE > 0 THEN SE=SQRT(SE/XNR) :
 END SAMPERR;
REGRLT:  PROC(C.NV.IER):
   /*     REPLACES A LOWER TRIANGULAR COV MTX      */
   /*     WITH A MTX OF REGR COES WITH ERROR       */
   /*     VARS ON THE MAIN DIAG.  (EACH VAR IS     */
   /*     PREDICTED BY THE PRECEDING ONES)         */
   /*     IER IS SET TO THE NUMBER                 */

   DCL C(1)       FLOAT BIN(53),
       (B,D.E)  FLOAT BIN(53) STATIC,
       (NV.IER) FIXED BIN(31),
         (N,K.I,KP,KR,KC.KE.JJ.J) FIXED BIN(31) STATIC:

     IER=0:
     N=NV-1;
     KP=0;
     DO K=1 TO N;
        KR=KP:     KP=KP+K:
        D=C(KP);
        IF D < 1.E-20 THEN DO:
           IER=1;
           GO TO ENDREG:
        END;
        DO I=1 TO NV:
           IF I < K THEN DO:
              KR=KR+1:     KC=KP+K:     JJ=K+1:
           END;
           ELSE IF I > K THEN DO:
              KR=KR+I-1:     KC=KR:       JJ=I:
           END:
           ELSE DO;
              KR=KR+1:
              GO TO NEXTI:
           END:
           B=C(KR)/D;
           DO J=JJ TO NV:
              KE=KC+I-K:
              C(KE)=C(KE)-B*C(KC);
              KC=KC+J:
           END:
           IF I > K THEN C(KR)=B;
     NEXTI: END:
     END:
   ENDREG:
 END REGRLT;
 END PRINTIT:
 /* MEM8 */
/*
//LKED.SYSLMOD DD DSN=NCES.XEJQNY.SAGELIB.DISP=OLD
//LKED.SYSLIB DD DSN=NCES.XEJQNY.SAGELIB.DISP=SHR
//             DD DSN=SYS2.SAS.V796.SUBLIB.DISP=SHR
//             DD DSN=SYS2.PLIXBASE.DISP=SHR
//             DD DSN=SYS2.PLIXLINK,DISP=SHR
//LKED.SYSIN  DD *
 NAME PRINTIT(R)
/*
END OF DATA
```

PARSER

```
//X            JOB (ED,AIR),'D-WYOUNG',PRTY=8,
//                 MSGCLASS=P,NOTIFY=XEJQNY,MSGLEVEL=(1,1)
/*ROUTE PRINT   R93
//ASM  EXEC XASMVCL,CREG=500K
//ASM.SYSPRINT  DD SYSOUT=P
//ASM.SYSLIB DD DSN=SYS2.SAS.V796.MACLIB,DISP=SHR
//ASM.SYSIN DD *
               PRINT    NOGEN
PROC           SASPROC NAME=BRRVAR,LOADMOD=BRRVAR2,DATA=ASGIVEN
OPTS           SASLIST MEAN,1,MEANS,1
OPTS           SASLIST STD,2
OPTS           SASLIST COV,3
OPTS           SASLIST COR,4,CORR,4
OPTS           SASLIST REG,5,REGR,5
OPTS           SASLIST SUM,6,SUMS,6
OPTS           SASLIST DES,7,DESIGN,7
OPTS           SASLIST DUMP,8
OPTS           SASLIST NOMISS,9,NM,9
PARMS          SASLIST NUMPSU,1,NPSU,1,MODE=NUMERIC
PARMS          SASLIST NREP,2,NREPS,2,MODE=NUMERIC
PARMS          SASLIST NDEP,3,NDEPS,3,MODE=NUMERIC
LISTS          SASLIST VARIABLES,1,VAR,1,MODE=NUMERIC
LISTS          SASLIST CLASSES,2,CLASS,2,MODE=ALPHANUM
LISTS          SASLIST PSUID,3,PSU,3,MODE=ALPHANUM
LISTS          SASLIST ADJBY,4,MODE=ALPHANUM
LISTS          SASLIST STRATID,5,STR,5,MODE=ALPHANUM
LISTS          SASLIST WEIGHT,6,WGT,6,MODE=NUMERIC
LISTS          SASLIST COMPUTE,7,COMP,7,MODE=NUMERIC
DATASETS       SASLIST DATA,1,MODE=DSIN
DATASETS       SASLIST OUT,2,MODE=DSOUT
DATASETS       SASLIST OUTEST,3,MODE=DSOUT
DATASETS       SASLIST ADJDATA,4,MODE=DSIN
DEFAULT        SASDFLT NO=1,NAME=_LAST_,MODE=DSIN
DEFAULT        SASDFLT NO=2,NAME=_DATA_,MODE=DSOUT
               BR     RT
INIT           ENTER
               SASLOAD SASART
               BALR     14,15
               STM      0,1,COMPLINK
               MVC      @NAME,=CL8'_OBS_'
               BAL      RT,PSUEDO
INIT           LEAVE
PSUEDO         MVI      @MAJNAME,C' '
               MVI      @SW,@SWPSE+@SWDEF
               MVI      @MODE,@A+@S
               L        15,$ADDLOC
               BR       15
BRRVARD        DSECT
COMPLINK       DS       2F
BRRVAR         CSECT
TERMER         LM       15,0,COMPLINK
               LA       1,1
               BR       15
TERMSUB        ENTER
               LM       15,0,COMPLINK
               LA       1,2
               BALR     14,15
               SASDLT   SASART
TERMSUB        LEAVE
```

```
STMTREC  ENTER
         COMP    OUTVAR.OUTVAR
         COMP    OUTVARS.OUTVAR
         COMP    OUTV.OUTVAR
         COMP    COMPVAR.CMPVAR
         COMP    COMPV.CMPVAR
         COMP    COMP.CMPVAR
         LM      15.0,COMPLINK
         SR      1,1
         BALR    14.15
STMTREC  LEAVE
OUTVAR   LA      0.16
         BAL     RT.STMTST
         WORD    .
         SR      5.5
OUTL     COMP    C':'.OUTE
         LIST    OUTI.OUTL
         NAME
         DATA    DATA=W1.N=8
         LA      5,1(5)
         B       OUTI
OUTE     ST      5.SUBLIST
         BAL     RT.STMTEND
         B       STMTRECX
CMPVAR   L       6.@DSADDR
         USING   D$TBL,6
         WORD    .
CMPL     COMP    C':'.CMPE
         LIST    COMPI,CMPL
         LA      0,D$VAR
         ST      0.@LIST
         DO      $VARFIND
         LH      0,@VARNO
         LTR     0,0
         BZ      COMPI
         LPADD   D$VARLST+16
         MVC     LNAME(12).@NAME
         MVC     22(2,2).@VARNO
         MVI     @MODE.@A+@S
         MVI     @SW.@SWDEF
         MVI     @MAJNAME.C' '
         DO      $ADDLOC
         B       COMPI
CMPE     B       STMTRECX
         SASEND  INIT=INIT.STMTREC=STMTREC.TERMERR=TERMER.TERM=TERMSU
         LTORG
         END
//LKED.SYSLMOD DD DSN=NCES.XEJQNY.SAGELIB.DISP=SHR
//LKED.SYSPRINT  DD SYSOUT=P
//LKED.SYSIN DD *
 SETSSI AC000001
 NAME BRRVAR(R)
END OF DATA
```