APPENDIX G

SAS CODE

1. CREATE POST-STRATIFICATION VARIABLES

```
*************************************************************************
* PROGRAM: POSTRAC.SAS
* TASK:    1999 DOD HEALTH CARE SURVEY ANALYSIS (8687-200)
* PURPOSE: SAS LOAD THE POST-STRATIFICATION DATA (AS OF 9/30/1999) - FORM C.
* WRITTEN: 04/27/2000 BY KEITH RATHBUN
*
* INPUTS:  1) CHILD99.TXT - POST-STRATIFICATION RAW DATA SET - FORM C
*
* OUTPUTS: 1) POSTRAC.SD2 - POST-STRATIFICATION SAS DATA SET - FORM C
*
* NOTES:
*
* 1)  Exclamation ('!') delimited ASCII file.
* 2)  December 31, 1999 data are not available as the New DEERS extract still
*     has not been fully processed.  These data are of the same time frame
*     as the Adult Post Stratification File, September 30, 1999 for
*     eligibility and September 1, 1999 for enrollment.
* 3)  Your specifications called for those under the Age of 18 as of
*     December 1, 1999.  This is the age upon which the extraction was
*     created, however as the Adult Post Stratification File was for
*     persons 18+ years of age as of 30 September, there may be some
*     duplication between these two files.
* 4)  The Age as of 1 December 1999 was derived based up on the birth date
*     indicated on the DEERS eligibility file (popout).
* 5)  I have kept the Age as of 30 September 1999 field as it appeared on
*     the 'popout' file.
* 6)  Eligibility Flag is a 'Y' or 'N' indicating the person is eligible
*     to receive care in the MHS.  This is based on DMIS Medical Eligibility
*     not in ('0' '3' '8') or DMIS Beneficiary Category not in ('0' '9')
*     being equal to 'Y'.
* 7)  PCM Derived is either 'MIL' or 'CIV' (again to avoid confusion with
*     the 00 & 01 in Primary Care Manager Code).  This field was code 'CIV'
*     for the following Enrollment DMIS IDs: 7901-7916, 8001-8036, 6901-6912.
*     All others were coded as 'MIL'.
* 8)  All files were merged together based on the common fields of Sponsor
*     SSN and DEERS Dependent Suffix.
* 9)  DMIS Derived ZIP Code contains the Unit ZIP code of an Active Duty
*     person and the Residence ZIP code of Dependents.  In some cases where
*     a ZIP code is missing, DMIS attempts to assign one based on other
*     available information.  For example if a Spouse's ZIP is unknown,
*     it may attempt to assign the ZIP code based on a known residence ZIP
*     of the Sponsor.
* 10) DMIS Catchment/NonCatchment ID is then appended based on the DMIS
*     Derived ZIP Code.
* 11) While a flag for enrollment was not created, a populated Alternate
*     Care Value (ACV) or PCM Derived field indicates enrollment in either
*     PRIME (ACV of 'A', 'E') or an USTF ( ACV 'U')
*
*************************************************************************
;
LIBNAME IN   "D:\KEITH\DOD99-8676\DATA";
LIBNAME OUT  "D:\KEITH\DOD99-8676\DATA";

OPTIONS PS=79 LS=132 COMPRESS=YES ERRORS=0 NOCENTER;

DATA OUT.POSTRAC;
   INFILE "D:\KEITH\DOD99-8676\DATA\CHILD99.TXT"
          RECFM=V LRECL=999 LENGTH=OLEN MISSOVER;
   INPUT  @001 SSN        $Char9.  /* Sponsor SSN                        */
               FILLER     $Char1.
               FSN        $Char1.  /* Family Sequence Number             */
               FILLER     $Char1.
               ZTYPE      $Char1.  /* Record Type                        */
               FILLER     $Char1.
               DDS        $Char2.  /* DEERS Dependent Suffix             */
               FILLER     $Char1.
               ZRZIP      $Char5.  /* Residence ZIP Code                 */
               FILLER     $Char1.
               ZRCATCH    $Char4.  /* Residence Hospital Catchment       */
               FILLER     $Char1.
               ZUCATCH    $Char4.  /* Unit Hospital Catchment            */
               FILLER     $Char1.
```

```
              ZSTATUS    $Char1.  /* Status of Sponsor                  */
              FILLER     $Char1.
              ZACFLAG    $Char3.  /* Alternate Care System History Flag */
              FILLER     $Char1.
              ZAGE_S30        3.  /* Age as of 30 September 1999        */
              FILLER     $Char1.
              ZMELIG     $Char1.  /* DMIS Medical Eligibility           */
              FILLER     $Char1.
              ZBENE      $Char1.  /* DMIS Beneficiary Category          */
              FILLER     $Char1.
              ZCATCHID   $Char4.  /* DMIS Catchment/NonCatchment ID     */
              FILLER     $Char1.
              ZDZIP      $Char5.  /* DMIS Derived ZIP Code              */
              FILLER     $Char1.
              ZACV       $Char1.  /* Alternate Care Value               */
              FILLER     $Char1.
              ZPCM       $Char2.  /* Primary Care Manager Code          */
              FILLER     $Char1.
              ZENRID     $Char4.  /* Enrollment DMIS ID                 */
              FILLER     $Char1.
              ZPCMD      $Char3.  /* PCM Derived                        */
              FILLER     $Char1.
              ZELIG      $Char1.  /* Eligibility Flag                   */
              FILLER     $Char1.
              ZAGE_D1         3.  /* Age as of 1 December 1999          */
              FILLER     $Char1.
        ;
   ************************************************************************
   * Construct SSNSMPL as SSN & FSN & DDS
   ************************************************************************;
   LENGTH SSNSMPL $12;
   SSNSMPL = SSN || FSN || DDS;

   LABEL SSNSMPL  = 'SSNSMPL - SSN & FSN & DDS          '
         SSN      = 'Sponsor SSN                        '
         FSN      = 'Family Sequence Number             '
         ZTYPE    = 'Record Type                        '
         DDS      = 'DEERS Dependent Suffix             '
         ZRZIP    = 'Residence ZIP Code                 '
         ZRCATCH  = 'Residence Hospital Catchment       '
         ZUCATCH  = 'Unit Hospital Catchment            '
         ZSTATUS  = 'Status of Sponsor                  '
         ZACFLAG  = 'Alternate Care System History Flag '
         ZAGE_S30 = 'Age as of 30 September 1999        '
         ZMELIG   = 'DMIS Medical Eligibility           '
         ZBENE    = 'DMIS Beneficiary Category          '
         ZCATCHID = 'DMIS Catchment/NonCatchment ID     '
         ZDZIP    = 'DMIS Derived ZIP Code              '
         ZACV     = 'Alternate Care Value               '
         ZPCM     = 'Primary Care Manager Code          '
         ZENRID   = 'Enrollment DMIS ID                 '
         ZPCMD    = 'PCM Derived                        '
         ZELIG    = 'Eligibility Flag                   '
         ZAGE_D1  = 'Age as of 1 December 1999          '
    ;
    DROP FILLER;
RUN;

TITLE1 "1999 DOD Health Survey Post-Stratification (8687-200)";
TITLE2 "Program Name: POSTRAC.SAS By Keith Rathbun";
TITLE3 "Program Outputs: POSTRAC.SD2";

PROC SORT DATA=OUT.POSTRAC; BY SSNSMPL; RUN;
PROC CONTENTS DATA=OUT.POSTRAC; RUN;

PROC FREQ DATA=OUT.POSTRAC;
   TABLES FSN
          ZTYPE
          DDS
          ZRCATCH
          ZUCATCH
          ZSTATUS
          ZACFLAG
```

```
              ZAGE_S30
              ZAGE_D1
              ZMELIG
              ZBENE
              ZCATCHID
              ZACV
              ZPCM
              ZENRID
              ZPCMD
              ZELIG
       /MISSING LIST;
    RUN;
*****************************************************************************
*
* PROGRAM:  IMPORTC.SAS
* TASK:     DOD HEALTH CARE SURVEY ANALYSIS (8687-100)
* PURPOSE:  Import the 1999 DoD HCSDB final adult (Form C) Dataset
* WRITTEN:  03/30/2000 BY KEITH RATHBUN
* SEQUENCE: First program run
*
* INPUTS:  1) HCRET02.XPT - 1999 DOD Health Survey Data - Child (Form C)
*                           (SAS Transport Format)
*
* OUTPUTS: 1) IMPORTC1.SD2 - 1999 DOD Health Survey Data - Child (Form 1)
*                           (SAS SD2 Format)
*          2) IMPORTC2.SD2 - 1999 DOD Health Survey Data - Child (Form 2)
*                           (SAS SD2 Format)
*          3) IMPORTC.SD2  - 1999 DOD Health Survey Data - Combined Child
*                           (Form 1 and 2 - SAS SD2 Format)
*
* NOTES:   1) The input file name is the same for child form 1 and 2.  Upon
*             export, the output files are renamed to be unique.  After
*             exporting the data to SD2 format, these two files are stacked
*             together and the FORM (1 or 2) indicator is created.
*
*****************************************************************************
*;
OPTIONS PS=79 LS=132 COMPRESS=YES NOCENTER NOFMTERR;

LIBNAME   SURVEY "D:\KEITH\DOD99-8676\DATA";

%MACRO PROCESS(FILENAME=,DIRNAME=,FORM=);
LIBNAME   LIBRARY "D:\KEITH\DOD99-8676\DATA\&DIRNAME";
LIBNAME XPT XPORT "D:\KEITH\DOD99-8676\DATA\&DIRNAME.\&filename..xpt";

****************************************************************************
* Import the 1999 DoD HCSDB final child (Form C) Dataset and produce
* summary report of CONTENTS and final status.
****************************************************************************;
PROC COPY IN=XPT OUT=SURVEY; RUN;

****************************************************************************
* Rename the 1999 DoD HCSDB final child (Form C) Dataset for consistency
* with program naming convention
****************************************************************************;
PROC DATASETS LIBRARY=SURVEY MEMTYPE=DATA;
   CHANGE &FILENAME=IMPORTC&FORM;
RUN;

TITLE1 "8687:  1999 DOD Health Survey Data - Final Form A (Adult)";
PROC CONTENTS DATA=SURVEY.IMPORTC&FORM; RUN;

TITLE2 "Unformatted Dump - Child Form &FORM";
PROC PRINT DATA=SURVEY.IMPORTC&FORM(OBS=50);
   FORMAT _ALL_;
RUN;

TITLE2 "FREQ of Final Status - Child Form &FORM";
PROC FREQ  DATA=SURVEY.IMPORTC&FORM;
   TABLE FLAG_FIN FLAG_DUP /MISSING LIST;
RUN;

%MEND PROCESS;
```

```
%PROCESS(FILENAME=HCRET02,DIRNAME=CHILD_1,FORM=1);
%PROCESS(FILENAME=HCRET02,DIRNAME=CHILD_2,FORM=2);

DATA SURVEY.IMPORTC;
   SET SURVEY.IMPORTC1(IN=IN1)
       SURVEY.IMPORTC2(IN=IN2);
   IF      IN1 THEN FORM = 1;
   ELSE IF IN2 THEN FORM = 2;
   LABEL FORM = "Child Form #";
RUN;
***************************************************************************
*
* PROGRAM:  CONVAR1C.SAS
* TASK:     1999 DOD HEALTH CARE SURVEY ANALYSIS (8676-100)
* PURPOSE:  Construct KEY Post-Stratification variables
*
* WRITTEN:  05/18/2000 BY DON JANG
*
* INPUTS:   1) POSTRAC.SD2 - Post-Stratification dataset - FORM C
*
* OUTPUTS:  2) CONVAR1C.SD2 - Constructed Variables FILE1C
*
* INCLUDES: 1) CACSMPL.INC - Colapsing rules for catchment area
*           2) CONSVAR0.INC - XREGION and CONUS
*
***************************************************************************
*;
LIBNAME  IN  "D:\KEITH\DOD99-8676\DATA";
LIBNAME  OUT "D:\KEITH\DOD99-8676\DATA";
OPTIONS PS=79 LS=132 COMPRESS=YES NOCENTER;

DATA OUT.CONVAR1C;
   SET IN.POSTRAC;

   LENGTH BFGROUPP 3.
          AGEGROUP 3.
          ENGROUPP 3.
   ;
   ************************************************************************
   * Calculate BFGROUPP
   ************************************************************************;
   IF ZTYPE = 'S' AND ZSTATUS IN ('A','B','J','N','P','V') THEN BFGROUPP = 1;
   ELSE IF ZTYPE = 'T' AND ZSTATUS IN ('A','B','J','N','P','V') THEN BFGROUPP = 2;
   ELSE BFGROUPP = 3;

   ************************************************************************
   * Calculate AGEGROUP
   ************************************************************************;
   IF      ZAGE_D1 LE  5 THEN AGEGROUP = 1;
   ELSE IF ZAGE_D1 LE 12 THEN AGEGROUP = 2;
   ELSE AGEGROUP = 3;

   ************************************************************************
   * Calculate ENGROUPP
   ************************************************************************;
   IF      ZPCMD = 'MIL' THEN ENGROUPP = 1;
   ELSE IF ZPCMD = 'CIV' THEN ENGROUPP = 2;
   ELSE ENGROUPP = 3;

   ************************************************************************
   * Assign CACSMPL
   ************************************************************************;
   IF ENGROUPP = 1 THEN CELLP = ZENRID*1;
   ELSE CELLP = ZCATCHID*1;
   %INCLUDE "D:\KEITH\DOD99-8676\SAS\CACSMPL.INC";
   %INCLUDE "D:\KEITH\DOD99-8676\SAS\CONSVAR0.SAS";

   ************************************************************************
   * Assign SUPREGP
   ************************************************************************;
   IF XREGION IN (1,2,5) THEN SUPREGP = 1;
   ELSE IF XREGION IN (6,9,10,11,12,16) THEN SUPREGP = 2;
```

```
   ELSE SUPREGP = 3;

   ************************************************************************
   * Remove Non US regions
   ************************************************************************;
   IF XREGION IN(13,14,15) THEN DELETE;

   LABEL  BFGROUPP = 'Beneficiary Group as of Dec. 1, 1999'
          AGEGROUP = 'Age Group as of Dec. 1, 1999'
          ENGROUPP = 'Enrollment Group as of Dec. 1, 1999'
          CACSMPL  = 'CACSMPL - Catchment Area'
          CELLP    = 'Catchment Area Post-Cell'
          SUPREGP  = 'SUPREGP - Super Region as of Dec 1, 1999'
          XREGION  = 'XREGION - Region'
          CONUS    = 'CONUS - CONUS/OCONUS Indicator'
          POSTSTR  = 'Post-Stratification Identifier';
   LENGTH XREGION  3.
          CONUS    3. ;
   ;

   POSTSTR = SUPREGP*1000 + ENGROUPP*100 + AGEGROUP*10;

RUN;

TITLE1 "1999 DOD Health Survey Post-Stratification Assignments (8687-100)";
TITLE2 "Program Name: CONVAR1A.SAS By Don Jang and Keith Rathbun";
TITLE3 "Program Output: CONVAR1A.SD2";
TITLE3 "WHERE ZELIG = Y";

PROC CONTENTS; RUN;

PROC FREQ;
WHERE ZELIG = "Y";
TABLES POSTSTR /MISSING LIST;
RUN;

*****************************************************************************
*
* PROGRAM: POSTRATC.SAS
* TASK:    1999 DOD HEALTH CARE SURVEY ANALYSIS (8687-100)
* PURPOSE: MERGE THE POST-STRATIFICATION DATA (AS OF 9/30/1999) WITH
*          ENTIRE FORM C SAMPLE
* WRITTEN: 05/19/2000 BY KEITH RATHBUN
*
* INPUTS:  1) CONVAR1C.SD2 - Post-Stratification file with constructed
*             variables for weighting and response rates - FORM C
*          2) XWALK.SD2    - CROSS-WALK FILE
*          3) FORMATC.SD2  - Formatted FORM C Questionnaire/Sample File
*
* OUTPUTS: 1) POSTRATC.SD2 - Post-Stratification file merged with
*             FORM C Questionnaire/Sample File
*
*****************************************************************************
;
LIBNAME IN1 "F:\";
LIBNAME IN2 "D:\KEITH\DOD99-8676\DATA";
LIBNAME OUT "D:\KEITH\DOD99-8676\DATA";
LIBNAME LIBRARY "D:\KEITH\DOD99-8676\DATA\FMTLIBC";

OPTIONS PS=79 LS=132 COMPRESS=YES NOCENTER;

*****************************************************************************
* SORT the XWALK file by SSNSMPL for merge purposes
*****************************************************************************;
PROC SORT DATA=IN1.XWALK OUT=XWALK; BY SSNSMPL; RUN;

*****************************************************************************
* SORT the Post-Stratification file by SSNSMPL for merge purposes
*****************************************************************************;
PROC SORT DATA=IN2.CONVAR1C OUT=CONVAR1C; BY SSNSMPL; RUN;

*****************************************************************************
* MERGE the Post-Stratification file with the XWALK file by SSNSMPL to
```

```
* attach the MPRID for merging with the Form C Questionnaire/Sample file
*****************************************************************************;
DATA MERGED;
   MERGE XWALK(IN=IN1)
         CONVAR1C(IN=IN2);
   BY SSNSMPL;
   IF IN1 AND IN2;
RUN;

*****************************************************************************
* SORT the MERGED file by MPRID so that it can be merged with the
* Form C Questionnaire/Sample file
*****************************************************************************;
PROC SORT DATA=MERGED; BY MPRID; RUN;

*****************************************************************************
* SORT the Form C Questionnaire/Sample file by MPRID
*****************************************************************************;
PROC SORT DATA=IN2.FORMATC OUT=FORMATC; BY MPRID; RUN;

*****************************************************************************
* MERGE the Post-Stratification file with the Form A Questionnaire/Sample
* file by MPRID
*****************************************************************************;
DATA OUT.POSTRATC;
   MERGE MERGED(IN=IN1)
         FORMATC(IN=IN2);
   BY MPRID;

   *************************************************************************
   * Calculate missing Post-Stratification values (i.e. No corresponding
   * SSNSMPL) from the original sampling variables.
   *************************************************************************;
   IF BFGROUPP = . THEN BFGROUPP = BGCSMPL;
   IF ENGROUPP = . THEN ENGROUPP = ENLSMPL;
   IF SUPREGP  = . THEN SUPREGP  = SUPREG;
   IF AGEGROUP = . THEN AGEGROUP = AGESMPL;

   POSTSTR = SUPREGP*1000 + ENGROUPP*100 + AGEGROUP*10;

   *************************************************************************
   * KEEP every sampled beneficiary.
   *************************************************************************;
   IF IN2;
RUN;

TITLE1 "1999 DOD Health Survey Post-Stratification (8687-100)";
TITLE2 "Program Name: POSTRATC.SAS By Keith Rathbun";
TITLE3 "Program Outputs: POSTRATC.SD2";
PROC CONTENTS; RUN;

PROC FREQ;
TABLES POSTSTR
       AGESMPL AGEGROUP AGESMPL*AGEGROUP
       ENLSMPL ENGROUPP ENLSMPL*ENGROUPP
       SUPREG  SUPREGP  SUPREG*SUPREGP
   /MISSING LIST;
RUN;
```

2. IMPLEMENT CODING SCHEME AND CODING TABLES

```
*************************************************************
*
* PROGRAM:  CSCHM99C.SAS
* PURPOSE:  APPLY CODING SCHEME TO DATA.
* WRITTEN:  03/01/00 Rankin
* DERIVED FROM:  Cschem99.sas
* MODIFIED:  6/22/2000
* PREVIOUS PROGRAM: FORMATC.SAS
* PURPOSE:
* INPUT: POSTRATC.SD2
* OUTPUT: CSCHM99C.SD2
* NOTE:  THERE IS ONE DATASET FOR BOTH VERSIONS
* USE FORM INDICATOR FOR CODING SCHEME PROCESSING
*************************************************************;

OPTIONS PS=79 LS=132 PAGENO=1;

LIBNAME  LIBRARY 'D:\KEITH\DOD99-8676\DATA\FMTLIBC';
LIBNAME  HCDATA  'D:\KEITH\DOD99-8676\DATA';
FILENAME GETFMTS 'D:\KEITH\DOD99-8676\SAS';

%LET INDATA= POSTRATC;
%LET OUTDATA= CSCHM99C;

/* Vairable names in survey -- become recoded varibles               */
/* Note: Includes questions from both versions of the questionnaire  */

%Let varlist1 =
C99C01  C99C02  C99C03  C99C04  C99C05  C99C06  C99C07
C99C08  C99C09  C99C10  C99C11  C99C12  C99C13  C99C14
C99C15  C99C16  C99C17  C99C18  C99C19  C99C20  C99C21
C99C22  C99C23  C99C24  C99C25  C99C26  C99C27  C99C28
C99C29  C99C30  C99C31  C99C32  C99C33  C99C34  C99C35
C99C36  C99C37  C99C38  C99C39  C99C40  C99C41  C99C42
C99C43  C99C44  C99C45  C99C46  C99C47  C99C48  C99C49
C99C50  C99C51  C99C52  C99C53  C99C54  C99C55  C99C56
C99C57  C99C58A C99C58B C99C58C C99C58D C99C58E C99C58F
C99C59  C99C60A C99C60B C99C60C C99C60D C99D61  C99D62
C99D62A C99D63  C99D63A C99D64  C99D64A C99D65  C99D65A
C99D66  C99D66A C99D67  C99D67A C99D67B C99D67C C99D67D
C99D68  C99D68A C99D69  C99D70  C99D71A C99D71B C99D71C
C99D71D C99D71E C99D72  C99D73  C99D74A C99D74B C99D74C
C99D74D C99D74E C99D74F C99D75  C99E61  C99E62  C99E62A
C99E62B C99E63  C99E63A C99E63B C99E64  C99E64  C99E64B
C99E65  C99E65A C99E65B C99E66  C99E66A C99E66B C99E67
C99E67A C99E67B C99E67C C99E67D C99E68  C99E68A C99E68B
C99E69  C99E69A C99E70  C99E71  C99E72A C99E72B C99E72C
C99E72D C99E72E C99E73  C99E74  C99E75A C99E75B C99E75C
C99E75D C99E75E C99E75F C99E76;


/*  _O variables are the original values from the survey response */

%Let varlist2 =
C99C01_O C99C02_O C99C03_O C99C04_O C99C05_O C99C06_O C99C07_O
C99C08_O C99C09_O C99C10_O C99C11_O C99C12_O C99C13_O C99C14_O
C99C15_O C99C16_O C99C17_O C99C18_O C99C19_O C99C20_O C99C21_O
C99C22_O C99C23_O C99C24_O C99C25_O C99C26_O C99C27_O C99C28_O
C99C29_O C99C30_O C99C31_O C99C32_O C99C33_O C99C34_O C99C35_O
C99C36_O C99C37_O C99C38_O C99C39_O C99C40_O C99C41_O C99C42_O
C99C43_O C99C44_O C99C45_O C99C46_O C99C47_O C99C48_O C99C49_O
C99C50_O C99C51_O C99C52_O C99C53_O C99C54_O C99C55_O C99C56_O
C99C57_O C99C58AO C99C58BO C99C58CO C99C58DO C99C58EO C99C58FO
C99C59_O C99C60AO C99C60BO C99C60CO C99C60DO C99D61_O C99D62_O
C99D62AO C99D63_O C99D63AO C99D64_O C99D64AO C99D65_O C99D65AO
C99D66_O C99D66AO C99D67_O C99D67AO C99D67BO C99D67CO C99D67DO
C99D68_O C99D68AO C99D69_O C99D70_O C99D71AO C99D71BO C99D71CO
C99D71DO C99D71EO C99D72_O C99D73_O C99D74AO C99D74BO C99D74CO
C99D74DO C99D74EO C99D74FO C99D75_O C99E61_O C99E62_O C99E62AO
C99E62BO C99E63_O C99E63AO C99E63BO C99E64_O C99E64AO C99E64BO
C99E65_O C99E65AO C99E65BO C99E66_O C99E66AO C99E66BO C99E67_O
C99E67AO C99E67BO C99E67CO C99E67DO C99E68_O C99E68AO C99E68BO
```

```
C99E69_O C99E69AO C99E70_O C99E71_O C99E72AO C99E72BO C99E72CO
C99E72DO C99E72EO C99E73_O C99E74_O C99E75AO C99E75BO C99E75CO
C99E75DO C99E75EO C99E75FO C99E76_O;


TITLE 'DoD 1999 Child Survey';
TITLE2 'Apply Coding Scheme';


 DATA HCDATA.&OUTDATA;

  %INCLUDE GETFMTS(FORMATS); /* label and format statements for */
                             /* original variables */

  SET HCDATA.&INDATA;


/*  First set up new variables that capture the original values */

  ARRAY RECODE &VARLIST1;
  ARRAY ORIG &VARLIST2;

   DO I = 1 to DIM(ORIG);
    ORIG(I) = RECODE(I);
   END;
  DROP I;


**** DRC entered B for all surveys not returned we will skip **;
**** around coding scheme for all surveys not returned       **;

  IF SCANDATE = .B AND C99C01 = .B THEN GO TO NOSURVEY;


/* Note 1:  C99C02, C99C03: child enrolled in Tricare Prime */

 IF C99C02 = 1 THEN N1 = 1;
 ELSE IF C99C02 = 2 THEN N1 = 2;
 ELSE IF C99C02 = 3 THEN DO;
  IF C99C03 = . THEN C99C03 = .N;
  ELSE C99C03 = .C;
  N1=3;
 END;
 ELSE IF C99C02 IN (., .A) THEN DO;
  IF C99C03 NE . THEN N1 = 4;
  ELSE IF C99C03 = . THEN DO;
   C99C02 = 3;
   N1 = 5;
  END;
 END;

/* Note 2:  C99C04, C99C05 -- C99C07: personal doctor or nurse */

  ARRAY NOTE2 C99C05 C99C06 C99C07;
   NOTMISS=0;
   NOTNA=0;
   NOTMISNA=0;
  DO OVER NOTE2;
    IF NOTE2 NE . THEN NOTMISS+1;    /* check for all missing */
    IF NOTE2 NE .N THEN NOTNA+1;   /* check for all NA */
    IF NOTE2 NE . AND NOTE2 NE .N THEN NOTMISNA+1; /* missing or NA */
  END;

  /*  NOTMISS=0  =>  ALL ARE BLANK
      NOTNA=0    =>  ALL ARE N/A
      NOTMISNA=0 =>  ALL ARE BLANK OR N/A
      NOTMISNA>0 =>  AT LEAST ONE IS MARKED   */

 IF C99C04=1 AND (NOTMISS = 0 OR NOTMISNA > 0) THEN N2 = 1;
 ELSE IF C99C04 IN (1, ., .A) AND
  NOTMISNA=0 AND NOTMISS > 0 THEN DO;  /* combination of missing */
                                       /* and NA                 */
  C99C04=2;
```

```
   DO OVER NOTE2;
    IF NOTE2 = . THEN NOTE2=.N;
    ELSE NOTE2 = .C;
   END;
  N2=2;
  END;

 ELSE IF C99C04 IN (2, ., .A) AND
  NOTMISNA > 0 THEN DO;        /*   at least one is marked   */
   C99C04=1;
   N2=3;
 END;
 ELSE IF C99C04 = 2 AND
  (NOTMISS=0 OR (NOTMISS>0 AND NOTMISNA=0))
  THEN DO;               /* all are blank or blank and NA */
   N2=4;
    DO OVER NOTE2;
     NOTE2=.N;
    END;
  END;
  ELSE IF C99C04 = .  AND NOTMISS=0 THEN N2=5; /* all missing */
  ELSE IF C99C04 = .A AND NOTMISS=0 THEN DO;  /* all missing */
   C99C04= 2;
   N2=6;
    DO OVER NOTE2;
     NOTE2= .N;
    END;
  END;

 DROP NOTMISS NOTMISNA NOTNA;

/*  Note 3: C99C05 -- C99C07:   personal doctor or nurse      */
/*  Note 3 and 4 are somewhat redundant, but for consistency  */
/*  the question is coded similarly to note 2                 */

/* Note: since recoded variables are overwritten by Note2     */
/* use original responses to recode further                   */

  ARRAY NOTE3 C99C06 C99C07;
  ARRAY NOTE3_O C99C06_O C99C07_O;

  N3NOTMIS=0;
  N3NOTNA=0;
  N3NMISNA=0;
  DO OVER NOTE3_O;
   IF NOTE3_O NE . THEN N3NOTMIS+1;     /* check for all missing */
   IF NOTE3_O NE .N THEN N3NOTNA+1;     /* check for all NA */
   IF NOTE3_O NE . AND NOTE3_O NE .N    /* missing or NA */
   THEN N3NMISNA+1;
  END;

  /*  N3NOTMIS=0  =>  ALL ARE BLANK
      N3NOTNA=0    =>  ALL ARE N/A
      N3NMISNA=0 =>  ALL ARE BLANK OR N/A
      N3NMISNA>0 =>  AT LEAST ONE IS MARKED   */

 IF C99C05 IN (.N, .C) THEN DO;
  N3 = 1;
  DO OVER NOTE3_O;
   IF NOTE3_O = . THEN NOTE3 = .N;
   ELSE NOTE3 =.C;
  END;
 END;
 ELSE IF (C99C05 >=1 OR C99C05 IN (., .A)) AND N3NMISNA > 0  /* at least */
 THEN N3 =2;                                          /* one is marked */
 ELSE IF (C99C05 >=1 OR C99C05 =.A) AND (N3NOTMIS = 0 OR (N3NOTMIS> 0 AND N3NMISNA=0))
 THEN DO;  /* all missing or missing and NA */
  C99C05 = .N;
  DO OVER NOTE3_O;
   IF NOTE3_O = . THEN NOTE3 = .N;
   ELSE NOTE3 = .C;
  END;
 N3=3;
```

```
 END;
 ELSE IF C99C05 = . THEN DO;
  IF N3NMISNA=0 AND N3NOTMIS > 0 THEN DO;
   C99C05 = .N;
   N3 = 4;
   DO OVER NOTE3_O;
    IF NOTE3_O = . THEN NOTE3= .N;
    ELSE NOTE3 = .C;
   END;
  END;
  ELSE IF N3NOTMIS=0 THEN N3=5;
 END;

DROP N3NMISNA N3NOTMIS N3NOTNA;

/** Note 4 -- personal doctor: C99C06, C99C07  **/;

 IF C99C06 IN (.N, .C) THEN DO;
  N4 = 1;
  IF C99C07_O = . THEN C99C07 = .N;
  ELSE C99C07 = .C;
 END;
 ELSE IF (C99C06 >=1 OR C99C06 IN (., .A)) AND C99C07_O NE .N
 THEN N4 =2;
 ELSE IF (C99C06 >=1 OR C99C06 IN (., .A)) AND C99C07_O = .N
 THEN DO;
  C99C06 = .N;
  C99C07 = .C;
  N4 =3;
 END;

/* Note 5:  C99C08 - C99C11, Primary care manager */

  ARRAY NOTE5 C99C09-C99C11;
  N5MARK=0;
  N5NOTNA=0;

   DO OVER NOTE5;
    IF NOTE5 NE . THEN N5MARK+1 ; /* check for all missing */
    IF NOTE5 NOT IN (-6,.N, .) THEN N5NOTNA+1; /* check for missing or NA */

   END;

 IF C99C08 = 1 AND N5MARK = 0 /* all missing */
 THEN N5=1;
 ELSE IF C99C08 IN (1, .A, .) AND N5NOTNA > 0 THEN DO; /* at least one is marked */
  C99C08 = 1;
  N5=2;
 END;

 ELSE IF C99C08 IN (1, .A, .) AND N5NOTNA = 0 AND N5MARK NE 0
 THEN DO; /* blank or NA */
  C99C08 = 2;
  N5=3;
   DO OVER NOTE5;
    IF NOTE5 = . THEN NOTE5 = .N;
    ELSE NOTE5 = .C;
   END;
 END;
 ELSE IF C99C08 IN (2, 3, -6, .N) AND N5MARK = 0 THEN DO; /* all missing */
  N5=4;
   DO OVER NOTE5;
    NOTE5 = .N;
   END;
  END;
 ELSE IF C99C08 IN (2, 3, -6, .N) AND N5NOTNA > 0
 THEN DO;  /*  at least one is marked   */
  C99C08 = 1;
  N5 = 5;
 END;
 ELSE IF C99C08 IN (2, 3, -6, .N) AND N5NOTNA=0 AND N5MARK>0
 THEN DO;      /* blank or NA */
  C99C08 = 2;
```

```
  N5 =6;
   DO OVER NOTE5;
    IF NOTE5 = . THEN NOTE5 = .N;
    ELSE NOTE5 = .C;
   END;
 END;
 ELSE IF C99C08 = .A AND N5MARK = 0 THEN DO;
  C99C08 = 3;
  N5 = 7;
   DO OVER NOTE5;
    IF NOTE5 = . THEN NOTE5=.N;
    ELSE NOTE5 = .C;
   END;
 END;
 ELSE IF C99C08 = . AND N5MARK = 0 THEN N5=8;

 DROP N5MARK N5NOTNA;

/*  Note 6 C99C10, C99C11 Primary Care Manager */

 IF C99C10 IN (.C, .N) AND C99C11 IN (.C, .N) THEN N6=1;
 ELSE IF C99C10 >=1 OR C99C10 IN (., .A) THEN N6=2;
 ELSE IF C99C10_O IN (.N, -6) THEN DO;
  N6=3;
  IF C99C11 = . THEN C99C11 = .N;
  ELSE C99C11 = .C;
 END;

/*  Note 7 C99C12 -- C99C16:  Health Care from a specialist  */

  ARRAY NOTE7 C99C13-C99C16;
  N7MARK=0;
  N7NOTNA=0;

   DO OVER NOTE7;
    IF NOTE7 NE . THEN N7MARK+1 ;
    IF NOTE7 NOT IN (-6, .N, .) THEN N7NOTNA+1;
   END;

  IF C99C14 IN (2, .) AND N7NOTNA > 0 THEN
  N7NOTNA=N7NOTNA-1;

  IF C99C12 = 1 AND (N7MARK = 0 OR (N7MARK > 0 AND N7NOTNA > 0))
  THEN N7=1; /* all are missing or at least one is marked */
  ELSE IF C99C12 IN (1, ., .A) AND (N7MARK > 0 AND N7NOTNA=0
  AND C99C14 IN (2, .))
  THEN DO;    /* blank or no usage or NA */
   C99C12 = 2;
   N7=2;
    DO OVER NOTE7;
     IF NOTE7 = . THEN NOTE7= .N;
     ELSE NOTE7 = .C;
    END;
  END;
  ELSE IF C99C12 IN (2, ., .A) AND N7MARK>0 AND N7NOTNA >0
  THEN DO;  /* at least one is marked */
   C99C12=1;
   N7=3;
  END;
  ELSE IF C99C12 = 2 AND (N7MARK = 0 OR  N7NOTNA = 0) THEN DO;
   N7=4;
   DO OVER NOTE7;
    IF NOTE7 = . THEN NOTE7 = .N;
    ELSE NOTE7 = .C;
   END;
  END;
  ELSE IF C99C12 = . AND N7MARK = 0 THEN N7=5;
  ELSE IF C99C12 = .A AND N7MARK = 0 THEN DO
   C99C12 = 2;
   N7 = 6;
   DO OVER NOTE7;
    NOTE7 = .N;
   END;
```

```
  END;

 DROP N7MARK N7NOTNA;

/******* Note 8 - health care specialist C99C14 -- C99C16  ******/

  ARRAY NOTE8 C99C15 C99C16;
  N8NOTMIS=0;
  N8NOTNA=0;

   DO OVER NOTE8;
    IF NOTE8  NE . THEN N8NOTMIS+1;
    IF NOTE8 NOT IN (.N, .) THEN N8NOTNA+1;
   END;

  IF C99C14 IN (.C, .N) AND C99C14 IN (.C, .N) AND C99C16 IN (.C, .N)
  THEN DO;
   N8=1;
    DO OVER NOTE8;
     IF NOTE8 IN (.N, .) THEN NOTE8 = .N;
     ELSE NOTE8 = .C;
    END;
  END;
  ELSE IF C99C14 = 1 AND (N8NOTMIS=0 OR       /*  all are blank  or */
  N8NOTNA >0) THEN N8=2;  /*  at lest one is marked */
  ELSE IF C99C14 IN (1, ., .A) AND
  N8NOTMIS > 0 AND N8NOTNA = 0 THEN DO; /* blank or NA */
   C99C14 = 2;
   N8 = 3;
   DO OVER NOTE8;
    IF NOTE8 = . THEN NOTE8 = .N;
    ELSE NOTE8 = .C;
   END;
  END;
  ELSE IF C99C14 IN (2, ., .A) AND N8NOTMIS > 0
  AND N8NOTNA > 0 THEN DO;   /* at least one is marked */
   C99C14 = 1;
   N8 = 4;
  END;
  ELSE IF C99C14 = 2 AND (N8NOTMIS = 0 OR N8NOTNA=0)
  THEN DO;  /*  all missing or missing or NA */
   N8=5;
   DO OVER NOTE8;
    IF NOTE8 = . THEN NOTE8 = .N;
    ELSE NOTE8 = .C;
   END;
  END;
  ELSE IF C99C14 = . AND N8NOTMIS = 0 THEN N8=6;
  ELSE IF C99C14 = .A AND N8NOTMIS = 0 THEN DO;
   C99C14 =2;
   N8=7;
   DO OVER NOTE8;
    IF NOTE8 = . THEN NOTE8 = .N;
    ELSE NOTE8 = .C;
   END;
  END;

  DROP N8NOTMIS N8NOTNA;

/** Note 9 -- call during regular office hours:  C99C17, C99C18 **/


 IF C99C17 = 1 AND (C99C18 GE 1 OR C99C18 IN (., .A)) THEN N9=1;
 ELSE IF C99C17 IN (1, ., .A) AND C99C18 IN (.N, -6) THEN DO;
  C99C17 = 2;
  N9 = 2;
 END;
 ELSE IF C99C17 IN (2, ., .A) AND (C99C18 GE 1 OR C99C18 = .A)
 THEN DO;
  C99C17 = 1;
  N9 = 3;
 END;
 ELSE IF C99C17 = 2 AND C99C18 IN (-6, .N, .) THEN DO;
```

```
  N9=4;
  IF C99C18 = . THEN C99C18 = .N;
  ELSE C99C18 = .C;
 END;
 ELSE IF C99C17 = . AND C99C18 = . THEN N9 =5;
 ELSE IF C99C17 = .A AND C99C18 = . THEN DO;
  C99C17 = 2;
  C99C18 = .N;
  N9=6;
 END;

/**** Note 10 -- regular or routine health care: C99C19 - C99C21 *****/

 ARRAY NOTE10 C99C20 C99C21;

 N10NMIS=0;
 N10NOTNA=0;

  DO OVER NOTE10;
   IF NOTE10 NE . THEN N10NMIS+1;
   IF NOTE10 NOT IN (., .N) THEN N10NOTNA+1;
  END;

 IF C99C19 = 1 AND (N10NMIS = 0      OR (N10NMIS > 0 AND N10NOTNA >0))
 THEN N10=1;   /*  all are missing or at least one is marked */
 ELSE IF C99C19 IN (1, ., .A) AND N10NMIS > 0 AND N10NOTNA = 0
 THEN DO;   /* Blank or NA */
  C99C19 = 2;
  N10=2;
  DO OVER NOTE10;
   IF NOTE10 = . THEN NOTE10 = .N;
   ELSE NOTE10 = .C;
  END;
 END;
 ELSE IF C99C19 IN (2, ., .A) AND N10NMIS > 0 AND N10NOTNA > 0
 THEN DO;  /*  At least one is marked  */
  C99C19 = 1;
  N10=3;
 END;
 ELSE IF C99C19 = 2 AND (N10NMIS = 0 OR (N10NMIS>0 AND N10NOTNA=0))
 THEN DO;  /* All are blank or Blank or NA   */
  N10=4;
  DO OVER NOTE10;
   IF NOTE10 = . THEN NOTE10 = .N;
   ELSE NOTE10 = .C;
  END;
 END;
 ELSE IF C99C19 = . AND N10NMIS = 0 THEN N10=5;
 ELSE IF C99C19 = .A AND N10NMIS = 0 THEN DO;
  C99C19 = 2;
  N10=6;
  DO OVER NOTE10;
   NOTE10 = .N;
  END;
 END;

 DROP N10NOTNA N10NMIS;

/******* Note 11 - immediate care: C99C22 -- C99C24 ********/

 ARRAY NOTE11 C99C23 C99C24;

 N11NMIS=0;
 N11NOTNA=0;

  DO OVER NOTE11;
   IF NOTE11 NE . THEN N11NMIS+1;
   IF NOTE11 NOT IN (., .N) THEN N11NOTNA+1;
  END;

 IF C99C22 = 1 AND (N11NMIS=0 OR (N11NMIS>0 AND N11NOTNA>0))
 THEN N11=1;  /*  All are missing or at least one is marked */
 ELSE IF C99C22 IN (1, ., .A) AND N11NOTNA=0 AND N11NMIS > 0
```

```
 THEN DO;  /*  Blank or NA */
  C99C22 = 2;
  N11=2;
  DO OVER NOTE11;
   IF NOTE11 = . THEN NOTE11 = .N;
   ELSE NOTE11 = .C;
  END;
 END;
 ELSE IF C99C22 IN (2, ., .A) AND N11NMIS > 0 AND N11NOTNA >0
 THEN DO;  /* At least one is marked */
  C99C22 = 1;
  N11=3;
 END;
 ELSE IF C99C22 = 2 AND (N11NMIS = 0 OR (N11NMIS > 0 AND N11NOTNA=0))
 THEN DO;  /* All blank or blank or NA  */
  N11=4;
  DO OVER NOTE11;
   IF NOTE11 = . THEN NOTE11 = .N;
   ELSE NOTE11 = .C;
  END;
 END;
 ELSE IF C99C22 = . AND N11NMIS = 0 THEN N11=5;
 ELSE IF C99C22 = .A AND N11NMIS = 0 THEN DO;
  C99C22=2;
  N11=6;
  DO OVER NOTE11;
   IF NOTE11 = . THEN NOTE11 = .N;
   ELSE NOTE11 = .C;
  END;
 END;

 DROP N11NMIS N11NOTNA;

/******* Note 12 - well-patient care:  C99C25 - C99C27      *******/


 ARRAY NOTE12 C99C26 C99C27;

  N12NMISS=0;
  N12NOTNA=0;

   DO OVER NOTE12;
    IF NOTE12 NE . THEN N12NMISS+1;
    IF NOTE12 NOT IN (., .N) THEN N12NOTNA+1;
   END;

 IF C99C25 = 1 AND (N12NMISS=0 OR N12NOTNA>0)
 THEN N12=1;  /*  All are missing or at least one is marked */
 ELSE IF C99C25 IN (1, ., .A) AND N12NOTNA=0 AND N12NMISS > 0
 THEN DO;  /*  Blank or NA */
  C99C25 = 2;
  N12=2;
  DO OVER NOTE12;
   IF NOTE12 = . THEN NOTE12 = .N;
   ELSE NOTE12 = .C;
  END;
 END;
 ELSE IF C99C25 IN (2, ., .A) AND N12NMISS > 0 AND N12NOTNA >0
 THEN DO;  /* At least one is marked */
  C99C25 = 1;
  N12=3;
 END;
 ELSE IF C99C25 = 2 AND (N12NMISS = 0 OR (N12NMISS > 0 AND N12NOTNA=0))
 THEN DO;  /* All blank or blank or NA  */
  N12=4;
  DO OVER NOTE12;
   IF NOTE12 = . THEN NOTE12 = .N;
   ELSE NOTE12 = .C;
  END;
 END;
 ELSE IF C99C25 = . AND N12NMISS = 0 THEN N12=5;
 ELSE IF C99C25 = .A AND  N12NMISS = 0 THEN DO;
  C99C25=2;
```

```
  N12=6;
  DO OVER NOTE12;
   IF NOTE12 = . THEN NOTE12 = .N;
   ELSE NOTE12 = .C;
  END;
 END;

 DROP N12NMISS N12NOTNA;

/******* Note 13 - doctor's office or clinic:  C99C29 -- C99C43 **/

 ARRAY NOTE13 C99C30-C99C43;
 ARRAY NOTE13A C99C30-C99C37 C99C39-C99C41;

 N13NMISS=0;
 N13NOTNA=0;

  DO OVER NOTE13;
   IF  NOTE13 NE . THEN N13NMISS+1;
  END;

  DO OVER NOTE13A;
   IF NOTE13A NOT IN (., .N) THEN N13NOTNA+1;
  END;

 IF C99C42 NOT IN (., 3) THEN N13NOTNA+1;
 IF C99C29 = 1 AND (N13NMISS=0 OR (N13NMISS>0 AND N13NOTNA=0))
 THEN DO;  /*  All are blank or blank or no usage */
  N13=1;
  DO OVER NOTE13;
   IF NOTE13 = . THEN NOTE13 = .N;
   ELSE NOTE13 = .C;
  END;
 END;
 ELSE IF C99C29 IN (1, ., .A) AND N13NMISS > 0 AND N13NOTNA >0
 THEN N13 =    2;  /* At least one is marked */
 ELSE IF C99C29 GE 2 AND (N13NMISS=0 OR
 (N13NMISS > 0 AND N13NOTNA >0))
 THEN N13=3;  /*  All are blank or at least one is marked */
 ELSE IF (C99C29 GE 2 OR C99C29 IN (., .A)) AND
 (N13NMISS > 0 AND N13NOTNA=0) THEN
 DO;  /* Blank or no usage or NA */
  C99C29 = 1;
  N13=4;
  DO OVER NOTE13;
   IF NOTE13 = . THEN NOTE13 = .N;
   ELSE NOTE13= .C;
  END;
 END;

 ELSE IF C99C29 = . AND N13NMISS = 0 THEN N13=5;
 ELSE IF C99C29 = .A AND N13NMISS = 0 THEN DO;
  C99C29=1;
  N13=6;
  DO OVER NOTE13;
   NOTE13 = .N;
  END;
 END;

 DROP N13NMISS N13NOTNA;

/******* Note14 - old enough to talk with doctors: C99C38, C99C39  ***/

 IF C99C38 IN (.C, .N) AND C99C39 IN (.C, .N) THEN N14 = 1;
 ELSE IF C99C38 = 1 AND C99C39 = . THEN N14 = 2;
 ELSE IF C99C38 = 1 AND C99C39 = .N THEN DO;
  C99C38 = .C;
  C99C39 = .C;
  N14=3;
 END;
 ELSE IF C99C38 IN (1, ., .A) AND (C99C39 GE 1 OR C99C39 IN (.A, .D))
 THEN DO;  /*  any mark but -6, NA */
  C99C38 = 1;
```

```
 N14=4;
END;
ELSE IF C99C38 IN (., .A) AND C99C39 = .N THEN DO;
 C99C38 = 2;
 C99C39 = .C;
 N14=5;
END;
ELSE IF C99C38 = 2 AND (C99C39 GE 1 OR C99C39 IN (.A, .D))
THEN DO; /* any mark but -6, NA */
 C99C38 = 1;
 N14=6;
END;
ELSE IF C99C38 = 2 AND (C99C39 = .N OR C99C39 = .) THEN
DO;
 IF C99C39 = . THEN C99C39 = .N;
 ELSE C99C39 = .C;
 N14=7;
END;
ELSE IF C99C38 = . AND C99C39 = . THEN N14=8;
ELSE IF C99C38 = .A AND C99C39 = . THEN DO;
 C99C38 = 2;
 N14=9;
 C99C39 = .N;
END;

/******* Note 15 - send in any claims:  C99C44 - C99C47   *******/

ARRAY NOTE15 C99C45 - C99C47;

N15NMISS=0;
N15NOTNA=0;
N15NMISN=0;

 DO OVER NOTE15;
  IF NOTE15 NE . THEN N15NMISS+1;
  IF NOTE15 NE .N THEN N15NOTNA+1;
  IF NOTE15 NOT IN (., .N) THEN N15NMISN+1;
 END;

IF C99C44 = 1 AND (N15NMISS=0 OR N15NMISN >0)
THEN N15 =1; /*  all are blank or at least one is marked */
ELSE IF C99C44 IN (1, 3, ., .A) AND N15NOTNA=0
THEN DO;    /*   all are NA */
 C99C44 = 2;
 N15=2;
 DO OVER NOTE15;
  NOTE15 = .C;
 END;
END;
ELSE IF C99C44 IN (1, .,.A) AND (N15NMISS > 0 AND N15NMISN = 0)
THEN DO;  /*  Blank or NA */
 C99C44 = 2;
 N15=3;
 DO OVER NOTE15;
  IF NOTE15= . THEN NOTE15 = .N;
  ELSE NOTE15 = .C;
 END;
END;
ELSE IF C99C44 IN (2, 3, .A, .) AND N15NMISS > 0 AND N15NMISN > 0
THEN DO;  /* At least one is marked */
 C99C44=1;
 N15=4;
END;
ELSE IF C99C44 IN (2, 3) AND (N15NMISS=0 OR (N15NMISN =0 AND N15NOTNA>0))
THEN DO;  /* Blank or NA or All NA */
 N15=5;
 DO OVER NOTE15;
  IF NOTE15 = . THEN NOTE15 = .N;
  ELSE NOTE15 =.C;
 END;
END;
ELSE IF C99C44 = 2 AND N15NOTNA=0 THEN DO;
 N15=6; /* All are NA */
```

```
  DO OVER NOTE15;
   NOTE15=.C;
  END;
 END;
 ELSE IF C99C44 = . AND N15NMISS=0 THEN N15=7;
 ELSE IF C99C44 = .A AND N15NMISS=0 THEN DO;
  C99C44=2;
  N15=8;
  DO OVER NOTE15;
   NOTE15 = .N;
  END;
 END;
 DROP N15NMISS N15NOTNA N15NMISN;


/******* Note 16 - written materials: C99C48, C99C49       ******/

 IF C99C48 = 1 AND C99C49 IN (1, 2, 3, ., .A) THEN N16=1;
 ELSE IF C99C48 IN (1, ., .A) AND C99C49 = .N THEN DO;
  C99C48 = 2;
  C99C49 = .C;
  N16=2;
 END;
 ELSE IF C99C48 IN (2, ., .A) AND C99C49 IN (1, 2, 3, .A) THEN DO;
  C99C48=1;
  N16=3;
 END;
 ELSE IF C99C48 = 2 AND C99C49 IN (., .N) THEN DO;
  N16=4;
  IF C99C49 = . THEN C99C49 =.N;
  ELSE C99C49 =.C;
 END;
 ELSE IF C99C48 = . AND C99C49 = . THEN N16=5;
 ELSE IF C99C48 = .A AND C99C49 = . THEN DO;
  C99C48 = 2;
  C99C49 = .N;
  N16=6;
 END;

/******* Note 17 - customer service:  C99C50, C99C51    *****/


 IF C99C50 = 1 AND C99C51 IN (1, 2, 3, ., .A) THEN N17=1;
 ELSE IF C99C50 IN (1, ., .A) AND C99C51 = .N THEN DO;
  C99C50 = 2;
  C99C51 = .C;
  N17=2;
 END;
 ELSE IF C99C50 IN (2, ., .A) AND C99C51 IN (1, 2, 3, .A) THEN DO;
  C99C50=1;
  N17=3;
 END;
 ELSE IF C99C50 = 2 AND C99C51 IN (., .N) THEN DO;
  N17=4;
  IF C99C51 = . THEN C99C51 =.N;
  ELSE C99C51 =.C;
 END;
 ELSE IF C99C50 = . AND C99C51 = . THEN N17=5;
 ELSE IF C99C50 = .A AND C99C51 = . THEN DO;
  C99C50 = 2;
  C99C51 = .N;
  N17=6;
 END;

/***  Note 18 - called or written with complaint C99C52 - C99C54     **/

 ARRAY NOTE18 C99C53 C99C54;

 N18NMISS=0;
 N18NOTNA=0;

 DO OVER NOTE18;
  IF NOTE18 NE . THEN N18NMISS+1;
```

```
 IF NOTE18 NOT IN (., .N) THEN N18NOTNA+1;
END;

IF C99C52 = 1 AND (N18NMISS=0 OR N18NOTNA >0) THEN N18=1;
ELSE IF C99C52 IN (1, ., .A) AND (N18NMISS>0 AND N18NOTNA=0)
THEN DO;   /* Blank or NA */
 C99C52=2;
 N18=2;
 DO OVER NOTE18;
  IF NOTE18 = . THEN NOTE18 = .N;
  ELSE NOTE18 =.C;
 END;
END;
ELSE IF C99C52 IN (2, ., .A) AND (N18NMISS>0 AND N18NOTNA>0)
THEN DO;  /*  At least one is marked   */
 C99C52 = 1;
 N18=3;
END;
ELSE IF C99C52 = 2 AND (N18NMISS=0 OR (N18NMISS>0 AND N18NOTNA=0))
THEN DO;  /*  All blank or Blank or NA */
 N18=4;
 DO OVER NOTE18;
 IF NOTE18 = . THEN NOTE18 = .N;
 ELSE NOTE18 = .C;
END;
END;
ELSE IF C99C52 = . AND N18NMISS=0 THEN N18=5;
ELSE IF C99C52 = .A AND N18NMISS=0 THEN DO;
 C99C52 = 2;
 N18=6;
 DO OVER NOTE18;
  NOTE18 = .N;
 END;
END;

DROP N18NMISS N18NOTNA;

/** Note 19 - C99C53, C99C54:  how long to resolve complaint            ***/
/** Note: Since questions overwritten by Note 18, use original responses **/
/** .N, valid skip recode is the same response as -6, NA                **/


IF C99C53 IN (.N, .C) AND C99C54 IN (.N, .C) THEN N19=1;
ELSE IF (C99C53 GE 1 OR C99C53 IN (., .A)) AND
(C99C54 IN (1, 2, ., .A)) THEN N19=2;
ELSE IF (C99C53 GE 1 OR C99C53 IN (., .A)) AND C99C54 = 3
THEN DO;
 C99C53=6;
 N19=3;
END;
ELSE IF (C99C53 GE 1 OR C99C53 IN (., .A)) AND C99C54=.N
THEN N19=4;
ELSE IF C99C53_O = .N AND C99C54 IN (1, 2, .A) THEN N19=5;
ELSE IF C99C53_O = .N AND C99C54 = 3 THEN DO;
 C99C53=6;
 N19=6;
END;

/*** Note 20 - paperwork:  C99C55, C99C56     *******/

IF C99C55 = 1 AND C99C56 IN (1, 2, 3, ., .A) THEN N20=1;
ELSE IF C99C55 IN (1, ., .A) AND C99C56 = .N THEN DO;
 C99C55=2;
 C99C56=.C;
 N20=2;
END;
ELSE IF C99C55 IN (2, ., .A) AND C99C56 IN (1, 2, 3, .A) THEN DO;
 C99C55=1;
 N20=3;
END;
ELSE IF C99C55 = 2 AND C99C56 IN (.N, .) THEN DO;
 N20=4;
 IF C99C56 = . THEN C99C56 = .N;
```

```
  ELSE C99C56 = .C;
 END;
 ELSE IF C99C55 = . AND C99C56 = . THEN N20=5;
 ELSE IF C99C55 = .A AND C99C56 = . THEN DO;
  C99C55=2;
  C99C56=.N;
  N20=6;
 END;

/*******  Questions that are specific to version 1 ***/

 IF FORM=1 THEN GO TO FORM1;
 ELSE GO TO FORM2;

/******* Note 21 - medicine prescribed by a doctor             ***/
/******* C99D62, C99D62A                                       ***/

 FORM1:

 IF C99D62 = 1 AND C99D62A =. THEN N21=1;
 ELSE IF C99D62 IN (1, ., .A) AND C99D62A IN (1,2,.A) THEN DO;
  C99D62=1;
  N21=2;
 END;
 ELSE IF C99D62 =2 AND C99D62A IN (1,.A) THEN DO;
  C99D62=1;
  N21=3;
 END;
 ELSE IF C99D62 =2 AND C99D62A= 2 THEN DO;
  C99D2A=.C;
  N21=4;
 END;
 ELSE IF C99D62 IN (2, .A) AND C99D62A = . THEN DO;
  C99D62=2;
  N21=5;
  IF C99D62A = . THEN C99D62A= .N;
  ELSE C99D62A= .C;
 END;
 ELSE IF C99D62 = . AND C99D62A =. THEN N21=6;

/******* Note 22 - unusual medical, mental or educational services    ***/
/******* C99D63, C99D63A                 ***/

 IF C99D63 = 1 AND C99D63A =. THEN N22=1;
 ELSE IF C99D63 IN (1, ., .A) AND C99D63A IN (1,2,.A) THEN DO;
  C99D63=1;
  N22=2;
 END;
 ELSE IF C99D63 =2 AND C99D63A IN (1,.A) THEN DO;
  C99D63=1;
  N22=3;
 END;
 ELSE IF C99D63=2 AND C99D63A=2 THEN DO;
  C99D63A=.C;
  N22=4;
 END;
 ELSE IF C99D63 IN (2, .A) AND C99D63A=. THEN DO;
  C99D63=2;
  N22=5;
  IF C99D63A = . THEN C99D63A= .N;
  ELSE C99D63A= .C;
 END;
 ELSE IF C99D63= . AND C99D63A=. THEN N22=6;

/******* Note 23: - C99D64, C99D64A    ***/

 IF C99D64 = 1 AND C99D64A =. THEN N23=1;
 ELSE IF C99D64 IN (1, ., .A) AND C99D64A IN (1,2,.A) THEN DO;
  C99D64=1;
  N23=2;
 END;
 ELSE IF C99D64 =2 AND C99D64A IN (1,.A) THEN DO;
  C99D64=1;
```

```
  N23=3;
 END;
 ELSE IF C99D64=2 AND C99D64A=2 THEN DO;
  C99D64A=.C;
  N23=4;
 END;
 ELSE IF C99D64 IN (2, .A) AND C99D64A =. THEN DO;
  C99D64=2;
  N23=5;
  IF C99D64A = . THEN C99D64A= .N;
  ELSE C99D64A= .C;
 END;
 ELSE IF C99D64 = . AND C99D64A =. THEN N23=6;

/******* Note 24: - special therapy:  C99D65, C99D65A          ***/

 IF C99D65 = 1 AND C99D65A =. THEN N24=1;
 ELSE IF C99D65 IN (1, ., .A) AND C99D65A IN (1,2,.A) THEN DO;
  C99D65=1;
  N24=2;
 END;
 ELSE IF C99D65 =2 AND C99D65A IN (1,.A) THEN DO;
  C99D65=1;
  N24=3;
 END;
 ELSE IF C99D65=2 AND C99D65A=2 THEN DO;
  C99D65A=.C;
  N24=4;
 END;
 ELSE IF C99D65 IN (2, .A) AND C99D65A =. THEN DO;
  C99D65=2;
  N24=5;
  IF C99D65A = . THEN C99D65A= .N;
  ELSE C99D65A= .C;
 END;
 ELSE IF C99D65 = . AND C99D65A =. THEN N24=6;

/******* Note 25: - treatment or counseling:  C99D66, C99D66A  ***/

 IF C99D66 = 1 AND C99D66A =. THEN N25=1;
 ELSE IF C99D66 IN (1, ., .A) AND C99D66A IN (1,2,.A) THEN DO;
  C99D66=1;
  N25=2;
 END;
 ELSE IF C99D66 =2 AND C99D66A IN (1,.A) THEN DO;
  C99D66=1;
  N25=3;
 END;
 ELSE IF C99D66=2 AND C99D66A=2 THEN DO;
  C99D66A=.C;
  N25=4;
 END;
 ELSE IF C99D66 IN (2, .A) AND C99D66A =. THEN DO;
  C99D66=2;
  N25=5;
  IF C99D66A = . THEN C99D66A= .N;
  ELSE C99D66A= .C;
 END;
 ELSE IF C99D66 = . AND C99D66A =. THEN N25=6;

/******* Note 26 - C99D67, C99D67A-C99D67D   *******/

 ARRAY NOTE26 C99D67A C99D67B C99D67C C99D67D;

 N26NMISS=0;
 N26MARK=0;

  DO OVER NOTE26;
   IF NOTE26  NE . THEN N26NMISS+1;
   IF NOTE26 =1 THEN N26MARK+1;
  END;

 IF C99D67D=2 THEN N26NMISS=N26NMISS-1;
```

```
 IF C99D67 = 1 AND N26NMISS=0 THEN N26=1;
 ELSE IF C99D67 IN (1,.,.A) AND N26NMISS> 0 THEN DO;
  C99D67=1;
  N26=2;
 END;
 ELSE IF C99D67 = 2 AND N26MARK > 0 THEN DO;
  C99D67=1;
  N26=3;
 END;
 ELSE IF C99D67 = 2 AND N26MARK=0 THEN DO;
  N26=4;
  DO OVER NOTE26;
   NOTE26 = .N;
  END;
 END;
 ELSE IF C99D67 = .  AND N26NMISS=0 THEN N26=5;
 ELSE IF C99D67 = .A AND N26NMISS=0 THEN DO;
  C99D67=2;
  N26=6;
  DO OVER NOTE26;
   NOTE26=.N;
  END;
 END;

 DROP N26NMISS N26MARK;

/******* Note 26A:  condition not mentioned above: C99D67A-D *****/

 ARRAY NOTE26A C99D67A C99D67B C99D67C C99D67D;

 N26ANOTM=0;
 N26YES=0;

 DO OVER NOTE26A;
  IF NOTE26A NOT IN (., .N) THEN N26ANOTM+1;
  IF NOTE26A = 1 THEN N26YES+1;
 END;

 IF C99D67A IN (.C, .N) THEN DO;
  N26A=1;
  DO OVER NOTE26A;
   IF NOTE26A IN (.N, .) THEN NOTE26A=.N;
   ELSE NOTE26A= .C;
  END;
 END;
 ELSE IF C99D67A = 1 AND N26ANOTM = 0 THEN N26A=2;
 ELSE IF C99D67A IN (1, ., .A) AND N26ANOTM> 0 THEN DO;
  C99D67A=1;
  N26A=3;
 END;
 ELSE IF C99D67A = 2 AND N26YES> 0 THEN DO;
  C99D67A=1;
  N26A=4;
 END;
 ELSE IF C99D67A = 2 AND N26YES=0 THEN DO;
  N26A=5;
  DO OVER NOTE26A;
   NOTE26A = .N;
  END;
 END;
 ELSE IF C99D67A = .  AND N26ANOTM=0 THEN  N26A=6;
 ELSE IF C99D67A = .A AND N26ANOTM=0 THEN DO;
  C99D67A = 2;
  N26A=7;
  DO OVER NOTE26A;
   NOTE26A = .N;
  END;
 END;

 DROP N26ANOTM N26YES;

/******* Questions that are specific to version 2 ***/
```

```
FORM2:

 IF FORM=1 THEN GO TO NOSURVEY;

/*** Note 27 - health condition: C99E62, C99E62A, C99E62B     ****/

 ARRAY NOTE27 C99E62A C99E62B;

 N27NMISS=0;
 N27YES=0;

  DO OVER NOTE27;
   IF NOTE27 NE . THEN N27NMISS+1;
   IF NOTE27 = 1 THEN N27YES+1;
  END;

 IF C99E62 = 1 AND N27NMISS=0 THEN N27=1;
 ELSE IF C99E62 IN (1, ., .A) AND N27NMISS>0 THEN DO;
  C99E62=1;
  N27=2;
 END;
 ELSE IF C99E62 = 2 AND N27YES>0 THEN DO;
  C99E62=1;
  N27=3;
 END;
 ELSE IF C99E62 = 2 AND N27YES=0 THEN DO;
  N27=4;
  DO OVER NOTE27;
   NOTE27=.N;
  END;
 END;
 ELSE IF C99E62 = . AND N27NMISS=0 THEN N27=5;
 ELSE IF C99E62 = .A AND N27NMISS=0 THEN DO;
  C99E62=2;
  N27=6;
  DO OVER NOTE27;
   NOTE27=.N;
  END;
 END;

 DROP N27NMISS N27YES;

/****** Note 27A - medicine prescribed by a doctor      *****/
/****** C99E62A, C99E62B                                *****/

 IF C99E62A IN (.C, .N) AND C99E62B IN (.C, .N) THEN N27A=1;
 ELSE IF C99E62A = 1 THEN DO;
  N27A=2;
  IF C99E62A = . THEN C99E62A= .N;
  ELSE C99E62A = .C;
 END;
 ELSE IF C99E62A = 2 THEN N27A=3;
 ELSE IF C99E62A IN (., .A) AND C99E62B IN (1, 2, .A) THEN DO;
  C99E62A=2;
  N27A=4;
 END;
 ELSE IF C99E62A = . AND C99E62B = . THEN N27A=5;
 ELSE IF C99E62A = .A AND C99E62B = . THEN DO;
  C99E62A = 1;
  N27A=6;
 END;

/*** Note 28 - medical, mental health, or educational services ***/
/*** C99E63, C99E63A, C99E63B     ****/

 ARRAY NOTE28 C99E63A C99E63B;

 N28NMISS=0;
 N28YES=0;

  DO OVER NOTE28;
   IF NOTE28 NE . THEN N28NMISS+1;
```

```
   IF NOTE28 = 1 THEN N28YES+1;
  END;

 IF C99E63 = 1 AND N28NMISS=0 THEN N28=1;
 ELSE IF C99E63 IN (1, ., .A) AND N28NMISS>0 THEN DO;
  C99E63=1;
  N28=2;
 END;
 ELSE IF C99E63 = 2 AND N28YES>0 THEN DO;
  C99E63=1;
  N28=3;
 END;
 ELSE IF C99E63 = 2 AND N28YES=0 THEN DO;
  N28=4;
  DO OVER NOTE28;
   NOTE28=.N;
  END;
 END;
 ELSE IF C99E63 = .  AND N28NMISS=0 THEN N28=5;
 ELSE IF C99E63 = .A AND N28NMISS=0 THEN DO;
  C99E63=2;
  N28=6;
  DO OVER NOTE28;
   NOTE28=.N;
  END;
 END;

DROP N28NMISS N28YES;

/****** Note 28A - lasted or expect to last 12 months   *****/
/****** C99E63A, C99E63B                                *****/

 IF C99E63A IN (.C, .N) AND C99E63B IN (.C, .N) THEN N28A=1;
 ELSE IF C99E63A = 1 THEN DO;
  N28A=2;
   IF C99E63A = . THEN C99E63A= .N;
   ELSE C99E63A = .C;
 END;
 ELSE IF C99E63A = 2 THEN N28A=3;
 ELSE IF C99E63A IN (., .A) AND C99E63B IN (1, 2, .A) THEN DO;
  C99E63A=2;
  N28A=4;
 END;
 ELSE IF C99E63A = . AND C99E63B = . THEN N28A=5;
 ELSE IF C99E63A = .A AND C99E63B = . THEN DO;
  C99E63A = 1;
  N28A=6;
 END;

/*** Note 29 - medical, mental health, or educational services ***/
/*** C99E64, C99E64A, C99E64B     ****/

 ARRAY NOTE29 C99E64A C99E64B;

 N29NMISS=0;
 N29YES=0;
  DO OVER NOTE29;
   IF NOTE29 NE . THEN N29NMISS+1;
   IF NOTE29 = 1  THEN N29YES+1;
  END;

 IF C99E64 = 1 AND N29NMISS=0 THEN N29=1;
 ELSE IF C99E64 IN (1, ., .A) AND N29NMISS >0 THEN DO;
  C99E64=1;
  N29=2;
 END;
 ELSE IF C99E64 = 2 AND N29YES>0 THEN DO;
  C99E64=1;
  N29=3;
 END;
 ELSE IF C99E64 = 2 AND N29YES=0 THEN DO;
  N29=4;
  DO OVER NOTE29;
```

```
   NOTE29=.N;
  END;
 END;
 ELSE IF C99E64 = . AND  N29NMISS=0 THEN N29=5;
 ELSE IF C99E64 = .A AND N29NMISS=0 THEN DO;
  C99E64=2;
  N29=6;
  DO OVER NOTE29;
   NOTE29=.N;
  END;
 END;

DROP N29NMISS N29YES;

/****** Note 29A - lasted or expect to last 12 months   *****/
/****** C99E64A, C99E64B                                *****/

 IF C99E64A IN (.C, .N) AND C99E64B IN (.C, .N) THEN N29A=1;
 ELSE IF C99E64A = 1 THEN DO;
  N29A=2;
  IF C99E64A = . THEN C99E64A= .N;
  ELSE C99E64A = .C;
 END;
 ELSE IF C99E64A = 2 THEN N29A=3;
 ELSE IF C99E64A IN (., .A) AND C99E64B IN (1, 2, .A) THEN DO;
  C99E64A=2;
  N29A=4;
 END;
 ELSE IF C99E64A = . AND C99E64B = . THEN N29A=5;
 ELSE IF C99E64A = .A AND C99E64B = . THEN DO;
  C99E64A = 1;
  N29A=6;
 END;

/*** Note 30 - medical, mental health, or educational services ***/
/*** C99E65, C99E65A, C99E65B     ****/

 ARRAY NOTE30 C99E65A C99E65B;

 N30NMISS=0;
 N30YES=0;

  DO OVER NOTE30;
   IF NOTE30 NE . THEN N30NMISS+1;
   IF NOTE30 = 1 THEN N30YES+1;
  END;

 IF C99E65 = 1 AND N30NMISS=0 THEN N30=1;
 ELSE IF C99E65 IN (1, ., .A) AND N30NMISS>0 THEN DO;
  C99E65=1;
  N30=2;
 END;
 ELSE IF C99E65 = 2 AND N30YES>0 THEN DO;
  C99E65=1;
  N30=3;
 END;
 ELSE IF C99E65 = 2 AND N30YES=0 THEN DO;
  N30=4;
  DO OVER NOTE30;
   NOTE30=.N;
  END;
 END;
 ELSE IF C99E65 = .  AND N30NMISS=0 THEN N30=5;
 ELSE IF C99E65 = .A AND N30NMISS=0 THEN DO;
  C99E65=2;
  N30=6;
  DO OVER NOTE30;
   NOTE30=.N;
  END;
 END;

 DROP N30NMISS N30YES;
```

```
/****** Note 30A - lasted or expect to last 12 months   *****/
/****** C99E65A, C99E65B                                *****/

 IF C99E65A IN (.C, .N) AND C99E65B IN (.C, .N) THEN N30A=1;
 ELSE IF C99E65A = 1 THEN DO;
  N30A=2;
  IF C99E65A = . THEN C99E65A= .N;
  ELSE C99E65A = .C;
 END;
 ELSE IF C99E65A = 2 THEN N30A=3;
 ELSE IF C99E65A IN (., .A) AND C99E65B IN (1, 2, .A) THEN DO;
  C99E65A=2;
  N30A=4;
 END;
 ELSE IF C99E65A = . AND C99E65B = . THEN N30A=5;
 ELSE IF C99E65A = .A AND C99E65B = . THEN DO;
  C99E65A = 1;
  N30A=6;
 END;

/*** Note 31 - problems which need counseling  ***/
/*** C99E66, C99E66A, C99E66B                  ***/

 ARRAY NOTE31 C99E66A C99E66B;

 N31NMISS=0;
 N31YES=0;

  DO OVER NOTE31;
   IF NOTE31 NE . THEN N31NMISS+1;
   IF NOTE31=1 THEN N31YES+1;
  END;

 IF C99E66 = 1 AND N31NMISS=0 THEN N31=1;
 ELSE IF C99E66 IN (1, ., .A) AND N31NMISS>0 THEN DO;
  C99E66=1;
  N31=2;
 END;
 ELSE IF C99E66 = 2 AND N31YES>0 THEN DO;
  C99E66=1;
  N31=3;
 END;
 ELSE IF C99E66 = 2 AND N31YES=0 THEN DO;
  N31=4;
  DO OVER NOTE31;
   NOTE31=.N;
  END;
 END;
 ELSE IF C99E66 = .  AND N31NMISS=0 THEN N31=5;
 ELSE IF C99E66 = .A AND N31NMISS=0 THEN DO;
  C99E66=2;
  N31=6;
  DO OVER NOTE31;
   NOTE31=.N;
  END;
 END;

 DROP N31NMISS N31YES;

/****** Note 31A - lasted or expect to last 12 months   *****/
/****** C99E66A, C99E66B                                *****/

 IF C99E66A IN (.C, .N) AND C99E66B IN (.C, .N) THEN N31A=1;
 ELSE IF C99E66A = 1 THEN DO;
  N31A=2;
  IF C99E66A = . THEN C99E66A= .N;
  ELSE C99E66A = .C;
 END;
 ELSE IF C99E66A = 2 THEN N31A=3;
 ELSE IF C99E66A IN (., .A) AND C99E66B IN (1, 2, .A) THEN DO;
  C99E66A=2;
  N31A=4;
 END;
```

```
 ELSE IF C99E66A = . AND C99E66B = . THEN N31A=5;
 ELSE IF C99E66A = .A AND C99E66B = . THEN DO;
  C99E66A = 1;
  N31A=6;
 END;

/******* Note 32 - C99E67, C99E67A-C99E67D   *******/

 ARRAY NOTE32 C99E67A C99E67B C99E67C C99E67D;

 N32NMISS=0;
 N32MARK=0;

  DO OVER NOTE32;
   IF NOTE32 NOT NE . THEN N32NMISS+1;
   IF NOTE32 =1 THEN N32MARK+1;
  END;

 IF C99E67 = 1 AND N32NMISS=0 THEN N32=1;
 ELSE IF C99E67 IN (1, ., .A) AND N32NMISS> 0 THEN DO;
  C99E67=1;
  N32=2;
 END;
 ELSE IF C99E67 = 2 AND N32MARK > 0 THEN DO;
  C99E67=1;
  N32=3;
 END;
 ELSE IF C99E67 = 2 AND N32MARK=0 THEN DO;
  N32=4;
  DO OVER NOTE32;
   NOTE32 = .N;
  END;
 END;
 ELSE IF C99E67 = . AND  N32NMISS=0 THEN N32=5;
 ELSE IF C99E67 = .A AND N32NMISS=0 THEN DO;
  C99E67=2;
  N32=6;
  DO OVER NOTE32;
   NOTE32=.N;
  END;
 END;

 DROP N32NMISS N32MARK;


/******* Note 32A:  condition not mentioned above: C99E67A-D *****/;

 ARRAY NOTE32A C99E67A C99E67B C99E67C C99E67D;

 N32ANOTM=0;
 N32YES=0;

 /******* Note 32A:  condition not mentioned above: C99E67A-D *****/

 DO OVER NOTE32A;
  IF NOTE32A NOT IN (., .N) THEN N32ANOTM+1;
  IF NOTE32A = 1 THEN N32YES+1;
 END;

 IF C99E67A IN (.C, .N) THEN DO;
  N32A=1;
  DO OVER NOTE32A;
   IF NOTE32A IN (.N, .) THEN NOTE32A=.N;
   ELSE NOTE32A= .C;
  END;
 END;
 ELSE IF C99E67A = 1 AND N32ANOTM = 0 THEN N32A=2;
 ELSE IF C99E67A IN (1, ., .A) AND N32YES> 0 THEN DO;
  C99E67A=1;
  N32A=3;
 END;
 ELSE IF C99E67A = 2 AND N32YES> 0 THEN DO;
  C99E67A=1;
```

```
  N32A=4;
 END;
 ELSE IF C99E67A = 2 AND N32YES=0 THEN DO;
  N32A=5;
  DO OVER NOTE32A;
   NOTE32A = .N;
  END;
 END;
 ELSE IF C99E67A = .  AND N32YES=0 THEN N32A=6;
 ELSE IF C99E67A = .A AND N32YES=0 THEN DO;
  C99E67A = 2;
  N32A=7;
  DO OVER NOTE32A;
   NOTE32A = .N;
  END;
 END;

 DROP N32ANOTM N32YES;


/*** Note 33 - problems which need counseling  ***/
/*** C99E66, C99E68A, C99E68B                  ***/

 ARRAY NOTE33 C99E68A C99E68B;

 N33NMISS=0;
 N33YES=0;

  DO OVER NOTE33;
   IF NOTE33 NE . THEN N33NMISS+1;
   IF NOTE33=1 THEN N33YES+1;
  END;

 IF C99E68 = 1 AND N33NMISS=0 THEN N33=1;
 ELSE IF C99E68 IN (1, ., .A) AND N33NMISS>0 THEN DO;
  C99E68=1;
  N33=2;
 END;
 ELSE IF C99E68 = 2 AND N33YES>0 THEN DO;
  C99E68=1;
  N33=3;
 END;
 ELSE IF C99E68 = 2 AND N33YES=0 THEN DO;
  N33=4;
  DO OVER NOTE33;
   NOTE33=.N;
  END;
 END;
 ELSE IF C99E68 = . AND N33NMISS=0 THEN N33=5;
 ELSE IF C99E68 = .A AND N33NMISS=0 THEN DO;
  C99E68=2;
  N33=6;
  DO OVER NOTE33;
   NOTE33=.N;
  END;
 END;

DROP N33NMISS N33YES;

/****** Note 33A - lasted or expect to last 12 months   *****/
/****** C99E68A, C99E68B                                *****/

 IF C99E68A IN (.C, .N) AND C99E68B IN (.C, .N) THEN N33A=1;
 ELSE IF C99E68A = 1 THEN DO;
  N33A=2;
  IF C99E68A = . THEN C99E68A= .N;
  ELSE C99E68A = .C;
 END;
 ELSE IF C99E68A = 2 THEN N33A=3;
 ELSE IF C99E68A IN (., .A) AND C99E68B IN (1, 2, .A) THEN DO;
  C99E68A=2;
  N33A=4;
 END;
```

```
 ELSE IF C99E68A = . AND C99E68B = . THEN N33A=5;
 ELSE IF C99E68A = .A AND C99E68B = . THEN DO;
  C99E68A = 1;
  N33A=6;
 END;

NOSURVEY:

  /* missing values */
  /* note: count only those questions included in either form1 or */
  /* form 2 */


  ARRAY MISS MISS_9 MISS_8 MISS_7 MISS_6 MISS_5 MISS_4 MISS_1 ;
  MISS_TOT = 0;
   DO OVER MISS;
    MISS = 0;
   END;
  ARRAY MISSARAY &VARLIST2;
  ARRAY FORM1 C99D61_O C99D62_O C99D62AO C99D63_O C99D63AO
              C99D64_O C99D64AO C99D65_O C99D65AO C99D66_O
          C99D66AO C99D67_O C99D67AO C99D67BO C99D67CO
          C99D67DO C99D68_O C99D68AO C99D69_O C99D70_O
          C99D71AO C99D71BO C99D71CO C99D71DO C99D71EO
          C99D72_O C99D73_O C99D74AO C99D74BO C99D74CO
              C99D74DO C99D74EO C99D74FO C99D75_O;


  ARRAY FORM2 C99E61_O C99E62_O C99E62AO C99E62BO C99E63_O
              C99E63AO C99E63BO C99E64_O C99E64AO C99E64BO
              C99E65_O C99E65AO C99E65BO C99E66_O C99E66AO
          C99E66BO C99E67_O C99E67AO C99E67BO C99E67CO
          C99E67DO C99E68_O C99E68AO C99E68BO C99E69_O
          C99E69AO C99E70_O C99E71_O C99E72AO C99E72BO
          C99E72CO C99E72DO C99E72EO C99E73_O C99E74_O
          C99E75AO C99E75BO C99E75CO C99E75DO C99E75EO
          C99E75FO C99E76_O;


   DO OVER MISSARAY;
    IF (MISSARAY EQ . ) THEN MISS_9 = MISS_9 + 1;
    ELSE IF (MISSARAY EQ .A) THEN MISS_8 = MISS_8 + 1;
    ELSE IF (MISSARAY EQ .O) THEN MISS_7 = MISS_7 + 1;
    ELSE IF (MISSARAY EQ .N) THEN MISS_6 = MISS_6 + 1;
    ELSE IF (MISSARAY EQ .D) THEN MISS_5 = MISS_5 + 1;
    ELSE IF (MISSARAY EQ .I) THEN MISS_4 = MISS_4 + 1;
    ELSE IF (MISSARAY EQ .C) THEN MISS_1 = MISS_1 + 1;
  END;

 IF FORM=1 THEN DO;
  DO OVER FORM2;     /* subtract missing values that are */
                     /* form 2 specific                  */
    IF FORM2 = . THEN MISS_9=MISS_9-1;
  END;
 END;

 IF FORM=2 THEN DO;
  DO OVER FORM1;     /* subtract missing values that are */
                     /* form 1 specific                  */
   IF FORM1 = . THEN MISS_9=MISS_9-1;
  END;
 END;

 DO OVER MISS;
  MISS_TOT=MISS_TOT + MISS;
 END;


/*************************************************************/

  OUTPUT;

RUN;
```

```
PROC CONTENTS DATA=HCDATA.&OUTDATA;
  RUN;

%MACRO GETFREQS (TABLES, NOTE, FORM1, FORM2);

 PROC FREQ DATA=HCDATA.&OUTDATA;
  WHERE FORM GE &FORM1 AND FORM LE &FORM2;
   TABLES &TABLES/MISSING LIST;
  FORMAT _ALL_ ;
  TITLE3 "CODING SCHEME FOR NOTE &NOTE";
 RUN;

%MEND GETFREQS;

%GETFREQS (N1*C99C02_O*C99C03_O, 1, 1, 2);
%GETFREQS (N2*C99C04_O*C99C05_O*C99C06_O*C99C07_O, 2, 1, 2);
%GETFREQS (N3*C99C05_O*C99C06_O*C99C07_O, 3, 1, 2);
%GETFREQS (N4*C99C06_O*C99C07_O, 4, 1, 2);
%GETFREQS (N5*C99C08_O*C99C09_O*C99C10_O*C99C11_O, 5, 1, 2);
%GETFREQS (N6*C99C10_O*C99C11_O, 6, 1, 2);
%GETFREQS (N7*C99C12_O*C99C13_O*C99C14_O*C99C15_O*C99C16_O, 7, 1, 2);
%GETFREQS (N8*C99C14_O*C99C15_O*C99C16_O, 8, 1, 2);
%GETFREQS (N9*C99C17_O*C99C18_O, 9, 1, 2);
%GETFREQS (N10*C99C19_O*C99C20_O*C99C21_O, 10, 1, 2);
%GETFREQS (N11*C99C22_O*C99C23_O*C99C24_O, 11, 1, 2);
%GETFREQS (N12*C99C25_O*C99C26_O*C99C27_O, 12, 1, 2);
%GETFREQS (N14*C99C38_O*C99C39_O*C99C38*C99C39, 14, 1, 2);
%GETFREQS (N15*C99C44_O*C99C45_O*C99C46_O*C99C47_O, 15, 1, 2);
%GETFREQS (N16*C99C48_O*C99C49_O, 16, 1, 2);
%GETFREQS (N17*C99C50_O*C99C51_O, 17, 1, 2);
%GETFREQS (N18*C99C52_O*C99C53_O*C99C54_O, 18, 1, 2);
%GETFREQS (N19*C99C53_O*C99C54_O, 19, 1, 2);
%GETFREQS (N20*C99C55_O*C99C56_O, 20, 1, 2);

/*** Questions specific to form 1  ***/

%GETFREQS (N21*C99D62_O*C99D62AO*C99D62*C99D62A, 21, 1, 1);
%GETFREQS (N22*C99D63_O*C99D63AO*C99D63*C99D63A, 22, 1, 1);
%GETFREQS (N23*C99D64_O*C99D64AO*C99D64*C99D64A, 23, 1, 1);
%GETFREQS (N24*C99D65_O*C99D65AO*C99D65*C99D65A, 24, 1, 1);
%GETFREQS (N25*C99D66_O*C99D66AO*C99D66*C99D66A, 25, 1, 1);
%GETFREQS (N26*C99D67*C99D67_O*C99D67AO*C99D67BO*C99D67CO*C99D67DO,
 26, 1, 1);
%GETFREQS (N26A*C99D67A*C99D67AO*C99D67BO*C99D67CO*C99D67DO, 26A, 1, 1);

/*** Questions specific to form 2  ***/

%GETFREQS (N27*C99E62_O*C99E62AO*C99E62BO, 27, 2, 2);
%GETFREQS (N27A*C99E62AO*C99E62BO, 27A, 2, 2);
%GETFREQS (N28*C99E63_O*C99E63AO*C99E63BO, 28, 2, 2);
%GETFREQS (N28A*C99E63AO*C99E63BO, 28A, 2, 2);
%GETFREQS (N29*C99E64_O*C99E64AO*C99E64BO, 29, 2, 2);
%GETFREQS (N29A*C99E64AO*C99E64BO, 29A, 2, 2);
%GETFREQS (N30*C99E65_O*C99E65AO*C99E65BO, 30, 2, 2);
%GETFREQS (N30A*C99E65AO*C99E65BO, 30A, 2, 2);
%GETFREQS (N31*C99E66_O*C99E66AO*C99E66BO, 31, 2, 2);
%GETFREQS (N31A*C99E66AO*C99E66BO, 31A, 2, 2);
%GETFREQS (N32*C99E67*C99E67_O*C99E67AO*C99E67BO*C99E67CO*C99E67DO,
 32, 2, 2);
%GETFREQS (N32A*C99E67_O*C99E67A*C99E67AO*C99E67BO*C99E67CO*C99E67DO,
 32A, 2, 2);
%GETFREQS (N33*C99E68_O*C99E68AO*C99E68BO, 33, 2, 2);
%GETFREQS (N33A*C99E68AO*C99E68BO, 33A, 2, 2);


PROC MEANS DATA=HCDATA.&OUTDATA N NMISS MIN MAX SUM MEAN;
TITLE3 'Frequency Checks - Missing Value Totals';
  VAR MISS_TOT MISS_1 MISS_4 MISS_5 MISS_6-MISS_9;
RUN;
```

3. CREATE STATUS FLAG FOR RECORD SELECTION

```
*************************************************************************
*
* PROGRAM: SELECTC.SAS
* TASK:    DOD HEALTH CARE SURVEY ANALYSIS (8676-200)
* PURPOSE: ASSIGN FINAL STATUS FOR RECORD SELECTION PURPOSES.
* WRITTEN: 11/09/99 BY KEITH RATHBUN
* INPUTS:  1) CSCHM99C.SD2 - 1999 DOD Health Survey Data
* OUTPUTS: 1) SELECTC.SD2 - 1999 DOD Health Survey Data w/FNSTATUS
*
*************************************************************************
*;
LIBNAME IN      "D:\KEITH\DOD99-8676\DATA";
LIBNAME OUT     "D:\KEITH\DOD99-8676\DATA";
LIBNAME LIBRARY "D:\KEITH\DOD99-8676\DATA\FMTLIBC";

OPTIONS PS=79 LS=132 COMPRESS=YES ERRORS=2 NOCENTER;

PROC SORT DATA=IN.CSCHM99C OUT=TEMPC1; BY MPRID; RUN;

DATA TEMPC2 OUT.DUPSC;
   SET TEMPC1;
   BY MPRID;
   ************************************************************************
   * Count key variables common to both form1 and form2.
   * (Total=35/36 for form1/form2), 50% rule = GE 17
   ************************************************************************;
   ARRAY KEYVAR C99C01  C99C02  C99C04  C99C08  C99C12  C99C17  C99C19
                C99C22  C99C25  C99C28  C99C29  C99C44  C99C48  C99C50
                C99C52  C99C55  C99C57  C99C59
                 ;
   KEYCOUNT = 0;
   DO I = 1 TO DIM(KEYVAR); DROP I;
      IF KEYVAR(I) NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT = KEYCOUNT + 1;
   END;

   ************************************************************************
   * Count question 58 - multiple response item.
   ************************************************************************;
   IF C99C58A NOT IN (.,.A,.O,.I,.B) OR
      C99C58B NOT IN (.,.A,.O,.I,.B) OR
      C99C58C NOT IN (.,.A,.O,.I,.B) OR
      C99C58D NOT IN (.,.A,.O,.I,.B) OR
      C99C58E NOT IN (.,.A,.O,.I,.B) OR
      C99C58F NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
```

```
************************************************************************
* Count question 60 - multiple response item.
************************************************************************;
IF C99C60A NOT IN (.,.A,.O,.I,.B) OR
   C99C60B NOT IN (.,.A,.O,.I,.B) OR
   C99C60C NOT IN (.,.A,.O,.I,.B) OR
   C99C60D NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;

IF FORM = 1 THEN DO;
   ***********************************************************************
   * Count FORM1 specific questions
   ***********************************************************************;
   IF C99D61  NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D62  NOT IN (.,.A,.O,.I,.B) OR
      C99D62A NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D63  NOT IN (.,.A,.O,.I,.B) OR
      C99D63A NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D64  NOT IN (.,.A,.O,.I,.B) OR
      C99D64A NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D65  NOT IN (.,.A,.O,.I,.B) OR
      C99D65A NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D66  NOT IN (.,.A,.O,.I,.B) OR
      C99D66A NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D67  NOT IN (.,.A,.O,.I,.B) OR
      C99D67A NOT IN (.,.A,.O,.I,.B) OR
      C99D67B NOT IN (.,.A,.O,.I,.B) OR
      C99D67C NOT IN (.,.A,.O,.I,.B) OR
      C99D67D NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D68  NOT IN (.,.A,.O,.I,.B) OR
      C99D68A NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D69  NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D70  NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D71A NOT IN (.,.A,.O,.I,.B) OR
      C99D71B NOT IN (.,.A,.O,.I,.B) OR
      C99D71C NOT IN (.,.A,.O,.I,.B) OR
      C99D71D NOT IN (.,.A,.O,.I,.B) OR
      C99D71E NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D72  NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D73  NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D74A NOT IN (.,.A,.O,.I,.B) OR
      C99D74B NOT IN (.,.A,.O,.I,.B) OR
      C99D74C NOT IN (.,.A,.O,.I,.B) OR
      C99D74D NOT IN (.,.A,.O,.I,.B) OR
      C99D74E NOT IN (.,.A,.O,.I,.B) OR
      C99D74F NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99D75  NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
END;
ELSE IF FORM = 2 THEN DO;
   ***********************************************************************
```

```
   * Count FORM2 specific questions
   ********************************************************************;
   IF C99E61  NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E62  NOT IN (.,.A,.O,.I,.B) OR
      C99E62A NOT IN (.,.A,.O,.I,.B) OR
      C99E62B NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E63  NOT IN (.,.A,.O,.I,.B) OR
      C99E63A NOT IN (.,.A,.O,.I,.B) OR
      C99E63B NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E64  NOT IN (.,.A,.O,.I,.B) OR
      C99E64A NOT IN (.,.A,.O,.I,.B) OR
      C99E64B NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E65  NOT IN (.,.A,.O,.I,.B) OR
      C99E65A NOT IN (.,.A,.O,.I,.B) OR
      C99E65B NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E66  NOT IN (.,.A,.O,.I,.B) OR
      C99E66A NOT IN (.,.A,.O,.I,.B) OR
      C99E66B NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E67  NOT IN (.,.A,.O,.I,.B) OR
      C99E67A NOT IN (.,.A,.O,.I,.B) OR
      C99E67B NOT IN (.,.A,.O,.I,.B) OR
      C99E67C NOT IN (.,.A,.O,.I,.B) OR
      C99E67D NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E68  NOT IN (.,.A,.O,.I,.B) OR
      C99E68A NOT IN (.,.A,.O,.I,.B) OR
      C99E68B NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E69  NOT IN (.,.A,.O,.I,.B) OR
      C99E69A NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E70  NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E71  NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E72A NOT IN (.,.A,.O,.I,.B) OR
      C99E72B NOT IN (.,.A,.O,.I,.B) OR
      C99E72C NOT IN (.,.A,.O,.I,.B) OR
      C99E72D NOT IN (.,.A,.O,.I,.B) OR
      C99E72E NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E73  NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E74  NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E75A NOT IN (.,.A,.O,.I,.B) OR
      C99E75B NOT IN (.,.A,.O,.I,.B) OR
      C99E75C NOT IN (.,.A,.O,.I,.B) OR
      C99E75D NOT IN (.,.A,.O,.I,.B) OR
      C99E75E NOT IN (.,.A,.O,.I,.B) OR
      C99E75F NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
   IF C99E76  NOT IN (.,.A,.O,.I,.B) THEN KEYCOUNT + 1;
END;
```

```
   ************************************************************************
   * Set flag for duplicates
   ************************************************************************;
   LENGTH DUPFLAG $3;
   DUPFLAG = 'NO';
   IF NOT (FIRST.MPRID AND LAST.MPRID) THEN DUPFLAG = 'YES';

   ************************************************************************
   * Determine final status (FNSTATUS)
   ************************************************************************;
   FNSTATUS = 0;
   IF FLAG_FIN = 1 THEN DO;
      ************************************************************
      **** APPLY THE COMPLETE QUESTIONNAIRE RULE (50% OF KEY   ****
      **** VARIABLES).                                          ****
      ************************************************************;
      IF KEYCOUNT GE 17 THEN FNSTATUS = 11;
      ELSE FNSTATUS = 12;
   END;
   ELSE IF FLAG_FIN IN(3,6,8,10,11,14,16,21) THEN DO;
      FNSTATUS = 20;
   END;
   ELSE IF FLAG_FIN IN(2,4,5,7,12,13,15) THEN DO;
      FNSTATUS = 30;
   END;
   ELSE IF FLAG_FIN IN(9,17,18,19,20,22) THEN DO;
      IF FLAG_FIN IN (18,19,20) THEN DO;
         FNSTATUS = 42;
      END;
      ELSE DO;
         FNSTATUS = 41;
      END;
   END;

   IF DUPFLAG = 'YES' THEN OUTPUT OUT.DUPSC;
   ELSE OUTPUT TEMPC2;

RUN;

***************************************************************************
* Select the "most complete" questionaire from duplicates and
* SET it back into the non-duplicates file. For now assume the lowest
* FNSTATUS Value is the "most complete".
***************************************************************************
;
PROC SORT DATA=OUT.DUPSC;
BY MPRID FNSTATUS;
RUN;
```

```
DATA DEDUPED;
   SET OUT.DUPSC;
   BY MPRID FNSTATUS;
   IF FIRST.MPRID; *KEEP only the first - most complete questionaire;
RUN;

DATA OUT.SELECTC;
   SET TEMPC2 DEDUPED;
   LABEL FNSTATUS = "Final Status"
         DUPFLAG  = "Multiple Response Indicator"
         STRATUM  = "Sampling STRATUM"
         KEYCOUNT = "# Key Questions Answered"
        ;
RUN;

TITLE1 "1999 DOD Health Survey FNSTATUS assignment (8676-200)";
TITLE2 "Program Name: SELECTC.SAS By Keith Rathbun";
TITLE3 "Program Output: SELECTC.SD2";

PROC CONTENTS DATA=OUT.SELECTC; RUN;

PROC FREQ DATA=OUT.SELECTC;
TABLES FORM FNSTATUS KEYCOUNT FLAG_FIN
       FORM*FNSTATUS*KEYCOUNT*FLAG_FIN
   /MISSING LIST;
RUN;
```

4. FINAL RECORD SELECTION AND REMOVE CONFIDENTIAL DATA

```
*************************************************************************
*
*  PROJECT:  1999 DOD Health Care Survey (8687-100)
*  PROGRAM:  SANITIZC.SAS
*  PURPOSE:  Create a file containing all records selected for analysis
*            (completed questionaire = FNSTATUS = 11) and delete
*            confidential fields from the file.
*
*    INPUT:  1) SELECTC.SD2 - All sampled children - Sampling, post-
*               stratification, and item response variables plus recodes.
*
*   OUTPUT:  2) SANITIZC.SD2 - Same as above except with confidential
*               fields removed and only with FNSTATUS = 11
*
* SEQUENCE:  After record selection but before the constructed variables
*
*    NOTES:  1) Drop all sensitive and all unnecessary variables.  Keep
*               only the observations with FNSTATUS = 11.
*
*************************************************************************;
LIBNAME        IN 'D:\KEITH\DOD99-8676\DATA\';
LIBNAME       OUT 'D:\KEITH\DOD99-8676\DATA';
LIBNAME   LIBRARY 'D:\KEITH\DOD99-8676\DATA\FMTLIBC';

OPTIONS PS=79 LS=132 COMPRESS=YES ERRORS=0 NOCENTER;

DATA OUT.SANITIZC;
   SET IN.SELECTC (DROP= AGE
                         AGE_N
                         AGESMPL
                         ARVDATE
                         ARVDAY
                         ARVMO
                         ARVYEAR
                         BATCH
                         BGCSMPL
                         DUPRET
                         DUPRET2
                         ELGCDE
                         ELGCDEN
                         ENLSMPL
                         ENR_DMIS
                         FLAG_DUP
                         FSN
                         INRECNO
                         LITHO
                         MAILING
                         MAILTYP
                         MISC
                         PCMCODE
                         PG
                         PRVCDE
                         RADRDMIS
                         RADRREG
                         RCLINIC
                         RECTYP
                         RETCOUNT
                         RETPROC
                         SADRREG
                         SAGE
                         SCANDATE
                         SCSINEL
                         SERIAL
                         SSN
                         SSNSMPL
                         STATUS
                         SUPREG
                         TSTBATCH
                         UADRREG
                         UCLINIC
                         UDMIS
                         ZACFLAG
```

```
                        ZBENE
                        ZDZIP
                        ZMELIG
                        ZPCM
                        ZRCATCH
                        ZRZIP
                        ZUCATCH);
   WHERE FNSTATUS=11;
RUN;

TITLE1 "1999 DOD Health Survey Post-Stratification (8687-100)";
TITLE2 "Program Name: SANITIZC.SAS";
TITLE3 "Program Outputs: SANITIZC.SD2";
TITLE4 "1999 Child Data - Sanitized";

PROC CONTENTS DATA=OUT.SANITIZC POSITION; RUN;
```

5. CONSTRUCTED VARIABLES FOR ANALYSIS

```
*******************************************************************************
* PROGRAM:   CONVARC2.SAS
* WRITTEN:   5/23/2000 BY NATALIE JUSTH
* PURPOSE:   TO CREATE 5 INDEPENDENT VARIABLES: XENRLLMT, XENR_PCM, XINS_COV
*            XQENROLL,  XBNFGRP
*            1 INDEPENDENT VARIABLE ALREADY CREATED FROM DEERS-BFGROUPP
*            TO CREATE 24 DEPENDENT VARIABLES: KDISENRL, KBGPRB1,
*            KBGPRB2, KMILWAT1, KCIVWAT1, KMILOFFC, KCIVOFFC, KMILOP99,
*            KCIVOP99, KCIVINS,
* INPUT:     J:\8687\DATA1999\CFINAL\SANITIZC.SD2
* OUTPUT:    J:\8687\DATA1999\CFINAL\CONSVAR2.SD2
*******************************************************************************
*;

LIBNAME IN 'J:\8687\DATA1999\CFINAL';
LIBNAME LIBRARY 'J:\8687\data1999\CFINAL\fmtlib';
OPTIONS PS=79 LS=132 ERRORS=2 CENTER ;
TITLE1 '1999 Health Care Survey of DoD Beneficiaries Study - Child Form C';
TITLE2 'CREATE CONSTRUCTED & OUTCOME MEASURE VARIABLES';

DATA IN.CONVARC2 (KEEP = XENRLLMT XENR_PCM XINS_COV XREGION ENGROUPP
                         BFGROUPP XBNFGRP KDISENRL
                         KMILWAT1 KCIVWAT1 KMILOFFC
                         KCIVOFFC KBGPRB1 KBGPRB2
                         KMILOP99 KCIVOP99
                         MPRID KCIVINS)
       CONSVAR2;
   SET IN.SANITIZC;


  LABEL
       XENRLLMT    =  "Enrollment in TRICARE Prime"
       XENR_PCM    =  "Enrollment by PCM type"
       XINS_COV    =  "Insurance Coverage"
       XREGION     =  "Health Care regions "
       XBNFGRP     =  "Constructed Beneficiary Group"
       KDISENRL    =  "Intention to disenroll      "
       KMILWAT1    =  "Wait <4 wks for well patient visit-Mil"
       KCIVWAT1    =  "Wait <4 wks for well patient visit-Civ"
       KMILOFFC    =  "Office wait of 30 min or more-Mil"
       KCIVOFFC    =  "Office wait of 30 min or more-Civ"
       KBGPRB1     =  "Big problem getting referrals to spclst"
       KBGPRB2     =  "Big problem getting neccessary care"
       KMILOP99    =  "Outpatient visits to Military facility"
       KCIVOP99    =  "Outpatient visits to Civilian facility"
       KCIVINS     =  "Beneficiary covered by civilian insurance"
       ;


  FORMAT
        XENRLLMT       ENROLL.
        ENGROUPP       ENLSMP.
        XENR_PCM       PCM.
        XINS_COV       INSURE.
        XREGION        CREG.
        XBNFGRP        XBGC_S.
        KDISENRL       HAYNN.
        KMILWAT1       HAYNN.
        KCIVWAT1       HAYNN.
        KMILOFFC       HAYNN.
        KCIVOFFC       HAYNN.
        KBGPRB1        HAYNN.
        KBGPRB2        HAYNN.
        KMILOP99       CTIMES.
        KCIVOP99       CTIMES.
        BFGROUPP       XBGC_S.
        CACSMPL        CAC.
        KCIVINS        HAYNN2_.
        ;
```

```
/* CREATE INDEPENDENT VARIABLES */


/* XENRLLMT--ENROLLMENT STATUS */
 IF ENGROUPP IN ( 1, 2)THEN XENRLLMT = 1;       /*   Enrolled  */
 ELSE IF ENGROUPP =3 THEN XENRLLMT = 2;         /*   Not Enrolled */

/* XENR_PCM--ENROLLMENT BY PCM TYPE */
 IF ENGROUPP = 1 THEN XENR_PCM = 1;        /*   Enrolled - mil PCM */
 ELSE IF ENGROUPP = 2 THEN XENR_PCM = 2;   /*   Enrolled - civ PCM */
 ELSE IF ENGROUPP = 3 THEN XENR_PCM = 3;   /*   Not Enrolled       */

/* XINS_COV--INSURANCE COVERAGE */
  IF C99C01 = 1 THEN XINS_COV = 1;                          /* Prime */
  ELSE IF C99C01 = 3 THEN XINS_COV = 2;                     /* Standard/Extra */
  ELSE IF C99C01 IN (4,6,7) THEN XINS_COV = 3;              /* Other Insurance */

/* XQENROLL--ENROLLMENT ACCORDING TO QUESTIONNAIRE RESPONSES AND */
/* USING DEERS SAMPLING VALUES  */
/*      IF C99C01 = 1 THEN DO;
       IF C99C11 = 1 THEN XQENROLL = 1;      * Enrolled - mil PCM ;
       ELSE IF C99C11 = 2 THEN XQENROLL = 2; * Enrolled - civ PCM ;
      END;
      ELSE IF C99C11 = 3 THEN DO;
         IF ENGROUPP = 1 THEN XQENROLL = 1;      * Enrolled - Mil PCM ;
         ELSE IF ENGROUPP = 2 THEN XQENROLL = 2; * Enrolled - Civ PCM ;
      END;
     ELSE IF C99C02 NE 1 THEN XQENROLL=3;        * Not enrolled     ; */

/* XBNFGRP-Beneficiary Group that excludes those 65 and over-Active Duty
           and Family Members of Active Duty */
   XBNFGRP = BFGROUPP;

/* KDISENRL--INTENTION TO DISNEROLL */
   IF C99C59  IN (4, 5) THEN KDISENRL = 1;                  /* Yes */
      ELSE IF C99C59  IN (1, 2, 3, -5) THEN KDISENRL = 2;   /* No  */


/* KMILWAT1--WAIT LESS THAN 4 WEEKS FOR WELL PATIENT VISIT AT MIL FACILITIES
   KCIVWAT1--WAIT LESS THAN 4 WEEKS FOR WELL PATIENT VISIT AT CIV FACILITIES*/
   IF C99C42  = 1 THEN DO;                          /* Military */
      IF C99C27  IN (1, 2, 3) THEN KMILWAT1 = 1;        /*   Yes */
      ELSE IF C99C27 = 4 THEN KMILWAT1 = 2;             /*   No  */
   END;
      ELSE IF C99C42  = 2 THEN DO;                      /*   Civilian */
           IF C99C27  IN (1, 2, 3) THEN KCIVWAT1 = 1;      /*   Yes */
           ELSE IF C99C27 = 4 THEN KCIVWAT1 = 2;           /*   No  */
      END;

/* KMILOFFC--OFFICE WAIT OF 30 MINUTES OR MORE AT MILITARY FACILITES
   KCIVOFFC--OFFICE WAIT OF 30 MINUTES OR MORE AT CIVILIAN FACILITES */
   IF C99C42  = 1 THEN DO;                                  /* Military */
      IF C99C32  IN (3,4) THEN KMILOFFC = 1;                /*  Yes */
      ELSE IF C99C32  IN (1,2) THEN KMILOFFC = 2;           /*  No  */
   END;
      ELSE IF C99C42  = 2 THEN DO;                          /*   Civilian */
           IF C99C32  IN (3,4) THEN KCIVOFFC = 1;           /* Yes */
           ELSE IF C99C32  IN (1,2) THEN KCIVOFFC = 2;      /* No */
      END;

/* KBGPRB1--BIG PROBLEM GETTING REFERRALS TO SPECIALISTS */
   IF C99C13  =1 THEN KBGPRB1 =1;                             /* YES */
      ELSE IF C99C13  IN (2,3) THEN KBGPRB1 =2;               /* NO  */

/* KBGPRB2--BIG PROBLEM GETTING NECESSARY CARE */
   IF C99C30  =1 THEN KBGPRB2 =1;                           /* YES */
       ELSE IF C99C30  IN (2,3) THEN KBGPRB2 =2;            /* NO  */

/* KMILOP99--OUTPATIENT VISITS TO MILITARY FACILITY
   KCIVOP99--OUTPATIENT VISITS TO CIVILIAN FACILITY */
   IF C99C42  = 1 THEN KMILOP99=C99C29;
      ELSE KMILOP99 = 1 ;
   IF C99C42  = 2 THEN KCIVOP99=C99C29;
```

```
      ELSE KCIVOP99 = 1 ;


/* KCIVINS--IS BENEFICIARY COVERED BY CIVILIAN INSURANCE */
   IF C99C01=4 OR C99C58D=1 THEN KCIVINS=1;  /* YES  */
    ELSE KCIVINS=2;                          /* NO   */


 RUN;

/* CHECK NEW 1999 VARIABLES */
PROC FREQ DATA=CONSVAR2;
 TABLES XENRLLMT XENR_PCM XINS_COV BFGROUPP XBNFGRP XREGION
        KDISENRL KBGPRB1 KBGPRB2
        KMILWAT1 KCIVWAT1 KMILOFFC KCIVOFFC
        KMILOP99 KCIVOP99 KCIVINS
        / MISSING LIST;
 TITLE3 'ONE WAY FREQUENCIES ON ALL NEW 1999 CONSTRUCTED VARIABLES';
 RUN;


   PROC FREQ DATA=CONSVAR2;
    TABLES BFGROUPP*ENGROUPP*XENRLLMT
           BFGROUPP*ENGROUPP*XENR_PCM
           XENRLLMT*C99C01*XINS_COV
           XREGION*CACSMPL
           BFGROUPP*XBNFGRP
           C99C59 *KDISENRL
           C99C42 *C99C26 *KMILWAT1*KCIVWAT1
           C99C42 *C99C32 *KMILOFFC*KCIVOFFC
           C99C13 *KBGPRB1
           C99C30 *KBGPRB2
           C99C42 *C99C29 *KMILOP99
           C99C42 *C99C29 *KCIVOP99
           C99C01*C99C58D*KCIVINS
           / MISSING LIST;
     TITLE3 'CROSSTABS ON ALL NEW VARIABLES';
            RUN;

   PROC FREQ DATA=CONSVAR2;
      tables XREGION*CACSMPL
           / MISSING LIST;
      format _all_;
   run;

   PROC CONTENTS DATA =IN.CONVARC2;
   RUN;
```

6. MERGE CONSTRUCTED VARIABLES ONTO DATA FILE

```
*******************************************************************************
* PROGRAM:   MERGE.SAS
* WRITTEN:   5/23/00 BY NATALIE JUSTH
* PURPOSE:   TO MERGE FINAL FILES TOGETHER AND REORDER BY VARIABLE TYPE
*            To reorder variables within the record use a
*             LENGTH statement before the SET statement.
*             Make sure that MPRID is the first variable in the
*             record followed by:
*                            1)  other sampling variables
*                            2)  DEERS variables
*                            3)  Post-stratification vars
*                            4)  questionnaire responses
*                            5)  DRC variables
*                            6)  recoded questionnaire responses
*                            7)  coding scheme flags
*                            8)  constructed variables
*                            9)  weights  (NOT AVAILABLE FOR PRELIMINARY DATA)
* INPUT:     J:\8687\DATA1999\CFINAL\SANITIZC.SD2
* OUTPUT:    J:\8687\DATA1999\CFINAL\HCS99C_4.SD2
*******************************************************************************
*;

LIBNAME   IN       'J:\8687\DATA1999\CFINAL';
LIBNAME   LIBRARY  'J:\8687\DATA1999\CFINAL\FMTLIB';
OPTIONS PS=75 LS=111 ERRORS=2 COMPRESS=YES;

PROC SORT DATA=IN.SANITIZC OUT=SANITIZC;
BY MPRID;
RUN;


PROC SORT DATA=IN.CONVARC2 OUT=CONSVAR2;
BY MPRID;
RUN;


DATA HCS99C_4(DROP=C99C01_O C99C02_O C99C03_O C99C04_O C99C05_O C99C06_O C99C07_O C99C08_O
          C99C09_O C99C10_O C99C11_O C99C12_O C99C13_O C99C14_O C99C15_O C99C16_O
          C99C17_O C99C18_O C99C19_O C99C20_O C99C21_O C99C22_O C99C23_O C99C24_O
          C99C25_O C99C26_O C99C27_O C99C28_O C99C29_O C99C30_O C99C31_O C99C32_O
          C99C33_O C99C34_O C99C35_O C99C36_O C99C37_O C99C38_O C99C39_O C99C40_O
          C99C41_O C99C42_O C99C43_O C99C44_O C99C45_O C99C46_O C99C47_O C99C48_O
          C99C49_O C99C50_O C99C51_O C99C52_O C99C53_O C99C54_O C99C55_O C99C56_O
          C99C57_O C99C58AO C99C58BO C99C58CO C99C58DO C99C58EO C99C58FO C99C59_O
          C99C60AO C99C60BO C99C60CO C99C60DO C99D61_O C99D62_O C99D62AO C99D63_O
          C99D63AO C99D64_O C99D64AO C99D65_O C99D65AO C99D66_O C99D66AO C99D67_O
          C99D67AO C99D67BO C99D67CO C99D67DO C99D68_O C99D68AO C99D69_O C99D70_O
          C99D71AO C99D71BO C99D71CO C99D71DO C99D71EO C99D72_O C99D73_O C99D74AO
          C99D74BO C99D74CO C99D74DO C99D74EO C99D74FO C99D75_O C99E61_O C99E62_O
          C99E62AO C99E62BO C99E63_O C99E63AO C99E63BO C99E64_O C99E64AO C99E64BO
          C99E65_O C99E65AO C99E65BO C99E66_O C99E66AO C99E66BO C99E67_O C99E67AO
          C99E67BO C99E67CO C99E67DO C99E68_O C99E68AO C99E68BO C99E69_O C99E69AO
          C99E70_O C99E71_O C99E72AO C99E72BO C99E72CO C99E72DO C99E72EO C99E73_O
          C99E74_O C99E75AO C99E75BO C99E75CO C99E75DO C99E75EO C99E75FO C99E76_O);
   MERGE SANITIZC(in=hcsdb) CONSVAR2 ;
   BY MPRID;
   if hcsdb;


   FORMAT
   MISS_1    HAMISS.
   MISS_4    HAMISS.
   MISS_5    HAMISS.
   MISS_6    HAMISS.
   MISS_7    HAMISS.
   MISS_8    HAMISS.
   MISS_9    HAMISS.
   MISS_TOT  HAMISS.
   ZTYPE      $RECTYP.
   DDS        $DDSFMT.
   ZSTATUS    $STATUS.
   ZACV       $ZACV.
```

```
   ZPCMD      $ZPCMD.
   ZELIG      $ZELIG.
   BFGROUPP       XBGC_S.
   ENGROUPP       ENLSMP.
   AGEGROUP       AGESMPL.
   CACSMPL    CAC.
   XREGION    CREG.
   REGSMPL    CREGSMPL.
   CONUS      CONUSMHS.
   SUPREGP    SUPERREG.
   MSTATUS    $MSTATUS.
     RACE       $RACE.
     SEX        $SEX.
     SVC        $SVC.
     GROUP      $GROUP.
     MPCSMPL    MPCSMPL.
     SVCSMPL    SVCSMPL.
     SEXSMPL    HASEX.
     FNSTATUS   FNSTATS.
     SSEX       $SEX.
     ;

  LABEL NHFF = "Stratum sample size";
  LABEL C99E66A = "Because of condition lasting 12 months";
  LABEL C99E66B = "Expect condition to last at least 12 mos";
RUN;

 DATA IN.HCS99C_4;


   LENGTH

       MPRID       $ 8         /* ID                */
       MPCSMPL       5         /* sampling variable */
       SVCSMPL       5         /* sampling variable */
       SEXSMPL       5         /* sampling variable */
       STRATUM     $ 3         /* sampling variable */
       NHFF          8         /* sampling variable */
       FLAG_FIN      4         /* sampling variable */
       FNSTATUS      8         /* sampling variable */
       KEYCOUNT      8         /* sampling variable */
       BWT99         8         /* sampling variable */


       MSTATUS     $ 1         /* DEERS variable    */
       RACE        $ 1         /* DEERS variable    */
       SEX         $ 1         /* DEERS variable    */
       SVC         $ 1         /* DEERS variable    */
       GROUP       $ 1         /* DEERS variable    */
       DDS         $ 2         /* DEERS variable    */



       ZAGE_D1       8         /* post-stratification variable    */
       ZAGE_S30      8         /* post-stratification variable    */
       ZACV        $ 1         /* post-stratification variable    */
       ZCATCHID    $ 4         /* post-stratification variable    */
       ZELIG       $ 1         /* post-stratification variable    */
       ZENRID      $ 4         /* post-stratification variable    */
       ZPCMD       $ 3         /* post-stratification variable    */
       ZSTATUS     $ 1         /* post-stratification variable    */
       ZTYPE       $ 1         /* post-stratification variable    */
       BFGROUPP      3         /* post-stratification variable    */
       AGEGROUP      3         /* post-stratification variable    */
       ENGROUPP      3         /* post-stratification variable    */
       CACSMPL       8         /* post-stratification variable    */
       REGSMPL       5         /* post-stratification variable    */
       POSTSTR       8         /* post-stratification variable    */
       CELLP         8         /* post-stratification variable    */
       POPSIZE       8         /* post-stratification variable    */


       C99C01        4         /* questionnaire     */
```

```
C99C02        4         /* questionnaire     */
C99C03        4         /* questionnaire     */
C99C04        4         /* questionnaire     */
C99C05        4         /* questionnaire     */
C99C06        4         /* questionnaire     */
C99C07        4         /* questionnaire     */
C99C08        4         /* questionnaire     */
C99C09        4         /* questionnaire     */
C99C10        4         /* questionnaire     */
C99C11        4         /* questionnaire     */
C99C12        4         /* questionnaire     */
C99C13        4         /* questionnaire     */
C99C14        4         /* questionnaire     */
C99C15        4         /* questionnaire     */
C99C16        4         /* questionnaire     */
C99C17        4         /* questionnaire     */
C99C18        4         /* questionnaire     */
C99C19        4         /* questionnaire     */
C99C20        4         /* questionnaire     */
C99C21        4         /* questionnaire     */
C99C22        4         /* questionnaire     */
C99C23        4         /* questionnaire     */
C99C24        4         /* questionnaire     */
C99C25        4         /* questionnaire     */
C99C26        4         /* questionnaire     */
C99C27        4         /* questionnaire     */
C99C28        4         /* questionnaire     */
GRID28      $ 3         /* questionnaire     */
C99C29        4         /* questionnaire     */
C99C30        4         /* questionnaire     */
C99C31        4         /* questionnaire     */
C99C32        4         /* questionnaire     */
C99C33        4         /* questionnaire     */
C99C34        4         /* questionnaire     */
C99C35        4         /* questionnaire     */
C99C36        4         /* questionnaire     */
C99C37        4         /* questionnaire     */
C99C38        4         /* questionnaire     */
C99C39        4         /* questionnaire     */
C99C40        4         /* questionnaire     */
C99C41        4         /* questionnaire     */
C99C42        4         /* questionnaire     */
C99C43        4         /* questionnaire     */
C99C44        4         /* questionnaire     */
C99C45        4         /* questionnaire     */
C99C46        4         /* questionnaire     */
C99C47        4         /* questionnaire     */
C99C48        4         /* questionnaire     */
C99C49        4         /* questionnaire     */
C99C50        4         /* questionnaire     */
C99C51        4         /* questionnaire     */
C99C52        4         /* questionnaire     */
C99C53        4         /* questionnaire     */
C99C54        4         /* questionnaire     */
C99C55        4         /* questionnaire     */
C99C56        4         /* questionnaire     */
C99C57        4         /* questionnaire     */
C99C58A       4         /* questionnaire     */
C99C58B       4         /* questionnaire     */
C99C58C       4         /* questionnaire     */
C99C58D       4         /* questionnaire     */
C99C58E       4         /* questionnaire     */
C99C58F       4         /* questionnaire     */
C99C59        4         /* questionnaire     */
C99C60A       4         /* questionnaire     */
C99C60B       4         /* questionnaire     */
C99C60C       4         /* questionnaire     */
C99C60D       4         /* questionnaire     */
C99D61        4         /* questionnaire     */
C99D62        4         /* questionnaire     */
C99D62A       4         /* questionnaire     */
C99D63        4         /* questionnaire     */
C99D63A       4         /* questionnaire     */
```

```
C99D64        4         /* questionnaire     */
C99D64A       4         /* questionnaire     */
C99D65        4         /* questionnaire     */
C99D65A       4         /* questionnaire     */
C99D66        4         /* questionnaire     */
C99D66A       4         /* questionnaire     */
C99D67        4         /* questionnaire     */
C99D67A       4         /* questionnaire     */
C99D67B       4         /* questionnaire     */
C99D67C       4         /* questionnaire     */
C99D67D       4         /* questionnaire     */
C99D68        4         /* questionnaire     */
C99D68A       4         /* questionnaire     */
C99D69        4         /* questionnaire     */
C99D70        4         /* questionnaire     */
C99D71A       4         /* questionnaire     */
C99D71B       4         /* questionnaire     */
C99D71C       4         /* questionnaire     */
C99D71D       4         /* questionnaire     */
C99D71E       4         /* questionnaire     */
C99D72        4         /* questionnaire     */
C99D73        4         /* questionnaire     */
C99D74A       4         /* questionnaire     */
C99D74B       4         /* questionnaire     */
C99D74C       4         /* questionnaire     */
C99D74D       4         /* questionnaire     */
C99D74E       4         /* questionnaire     */
C99D74F       4         /* questionnaire     */
C99D75        4         /* questionnaire     */
C99E61        4         /* questionnaire     */
C99E62        4         /* questionnaire     */
C99E62A       4         /* questionnaire     */
C99E62B       4         /* questionnaire     */
C99E63        4         /* questionnaire     */
C99E63A       4         /* questionnaire     */
C99E63B       4         /* questionnaire     */
C99E64        4         /* questionnaire     */
C99E64A       4         /* questionnaire     */
C99E64B       4         /* questionnaire     */
C99E65        4         /* questionnaire     */
C99E65A       4         /* questionnaire     */
C99E65B       4         /* questionnaire     */
C99E66        4         /* questionnaire     */
C99E66A       4         /* questionnaire     */
C99E66B       4         /* questionnaire     */
C99E67        4         /* questionnaire     */
C99E67A       4         /* questionnaire     */
C99E67B       4         /* questionnaire     */
C99E67C       4         /* questionnaire     */
C99E67D       4         /* questionnaire     */
C99E68        4         /* questionnaire     */
C99E68A       4         /* questionnaire     */
C99E68B       4         /* questionnaire     */
GRID68      $ 2         /* questionnaire     */
C99E69        4         /* questionnaire     */
C99E69A       4         /* questionnaire     */
GRID69      $ 2         /* questionnaire     */
C99E70        4         /* questionnaire     */
C99E71        4         /* questionnaire     */
C99E72A       4         /* questionnaire     */
C99E72B       4         /* questionnaire     */
C99E72C       4         /* questionnaire     */
C99E72D       4         /* questionnaire     */
C99E72E       4         /* questionnaire     */
C99E73        4         /* questionnaire     */
C99E74        4         /* questionnaire     */
SSEX        $ 1         /* questionnaire     */
C99E75A       4         /* questionnaire     */
C99E75B       4         /* questionnaire     */
C99E75C       4         /* questionnaire     */
C99E75D       4         /* questionnaire     */
C99E75E       4         /* questionnaire     */
C99E75F       4         /* questionnaire     */
```

```
C99E76        4         /* questionnaire     */


REFUSE        4         /* DRC variable      */
DUPFLAG     $ 3         /* DRC variable      */
BLKREAS       4         /* DRC variable      */
FORM          8         /* DRC variable      */

N1            4         /* CS flag variable     */
N2            4         /* CS flag variable     */
N3            4         /* CS flag variable     */
N4            4         /* CS flag variable     */
N5            4         /* CS flag variable     */
N6            8         /* CS flag variable     */
N7            4         /* CS flag variable     */
N8            4         /* CS flag variable     */
N9            4         /* CS flag variable     */
N10           4         /* CS flag variable     */
N11           4         /* CS flag variable     */
N12           4         /* CS flag variable     */
N13           4         /* CS flag variable     */
N14           4         /* CS flag variable     */
N15           4         /* CS flag variable     */
N16           4         /* CS flag variable     */
N17           4         /* CS flag variable     */
N18           4         /* CS flag variable     */
N19           4         /* CS flag variable     */
N20           4         /* CS flag variable     */
N21           4         /* CS flag variable     */
N22           4         /* CS flag variable     */
N23           4         /* CS flag variable     */
N24           4         /* CS flag variable     */
N25           4         /* CS flag variable     */
N26           4         /* CS flag variable     */
N26A          4         /* CS flag variable     */
N27           4         /* CS flag variable     */
N27A          4         /* CS flag variable     */
N28           4         /* CS flag variable     */
N28A          4         /* CS flag variable     */
N29           4         /* CS flag variable     */
N29A          4         /* CS flag variable     */
N30           4         /* CS flag variable     */
N30A          4         /* CS flag variable     */
N31           4         /* CS flag variable     */
N31A          4         /* CS flag variable     */
N32           4         /* CS flag variable     */
N32A          4         /* CS flag variable     */
N33           4         /* CS flag variable     */
N33A          4         /* CS flag variable     */
MISS_1        8         /* CS Count             */
MISS_4        8         /* CS Count             */
MISS_5        8         /* CS Count             */
MISS_6        8         /* CS Count             */
MISS_7        8         /* CS Count             */
MISS_8        8         /* CS Count             */
MISS_9        8         /* CS Count             */
MISS_TOT      8         /* CS Count             */
XREGION       3         /* constructed       */
CONUS         3         /* constructed       */
SUPREGP       8         /* constructed       */
XENRLLMT      8         /* constructed       */
XENR_PCM      8         /* constructed       */
XINS_COV      8         /* constructed       */
KDISENRL      8         /* constructed       */
KMILWAT1      8         /* constructed       */
KCIVWAT1      8         /* constructed       */
KMILOFFC      8         /* constructed       */
KCIVOFFC      8         /* constructed       */
KBGPRB1       8         /* constructed       */
KBGPRB2       8         /* constructed       */
KMILOP99      8         /* constructed       */
KCIVOP99      8         /* constructed       */
KCIVINS       8         /* constructed       */
```

```
     SET HCS99C_4;

     RUN;

PROC CONTENTS DATA=IN.HCS99C_4 POSITION;
RUN;
```

7. RESPONSE RATE CALCULATIONS

```
*****************************************************************************
*
* PROGRAM: TABLE02.SAS
* TASK:    1999 DOD HEALTH CARE SURVEY ANALYSIS (8676-002)
* PURPOSE: BUILD TABLE 2: RESPONSE RATES BY DOMAIN SUMMARY
*          1999 DOD HEALTH CARE SURVEY FILE.
* WRITTEN: 11/09/99 BY KEITH RATHBUN
*
* INCLUDES: 1) TABLE02.IN1
*           2) TABLE02.IN2
*
*****************************************************************************
*;
LIBNAME IN   "D:\KEITH\DOD99-8676\DATA";
LIBNAME OUT  "D:\KEITH\DOD99-8676\DATA";
OPTIONS PS=79 LS=132 COMPRESS=YES ERRORS=0 NOCENTER;

%MACRO PROCESS(INPT,FORM,LIB);
LIBNAME LIBRARY &LIB;
*********************************************************************
* Process OVERALL Summary of response rates
*********************************************************************
;
DATA _NULL_;
   SET IN.&INPT END=FINISHED;
   IF _N_ = 1 THEN DO;
      SN    = 0;
      SN1   = 0;
      SN11  = 0;
      SN12  = 0;
      SN2   = 0;
      SN3   = 0;
      SN4   = 0;
      SN41  = 0;
      SN42  = 0;
      WN   = 0;
      WN1  = 0;
      WN11 = 0;
      WN12 = 0;
      WN2  = 0;
      WN3  = 0;
      WN4  = 0;
      WN41 = 0;
      WN42 = 0;
   END;
****************************************************
* Accumulate group 1 weighted and unweighted counts
****************************************************
;
SN + 1;
WN + BWT99;
IF FNSTATUS IN(11,12) THEN DO;
   SN1 + 1;
   WN1 + BWT99;
   IF FNSTATUS = 11 THEN DO;
      SN11 + 1;
      WN11 + BWT99;
   END;
   ELSE DO;
      SN12 + 1;
      WN12 + BWT99;
   END;
END;
****************************************************
* Accumulate group 2 weighted and unweighted counts
****************************************************
;
ELSE IF FNSTATUS = 20 THEN DO;
   SN2 + 1;
   WN2 + BWT99;
END;
****************************************************
```

```
* Accumulate group 3 weighted and unweighted counts
****************************************************
;
ELSE IF FNSTATUS = 30 THEN DO;
   SN3 + 1;
   WN3 + BWT99;
END;
****************************************************
* Accumulate group 4 weighted and unweighted counts
****************************************************
;
ELSE IF FNSTATUS IN(41,42) THEN DO;
   SN4 + 1;
   WN4 + BWT99;
   IF FNSTATUS = 42 THEN DO;
      SN42 + 1;
      WN42 + BWT99;
   END;
   ELSE DO;
      SN41 + 1;
      WN41 + BWT99;
   END;
END;

DROP I;
RETAIN
   SN
   SN1
   SN11
   SN12
   SN2
   SN3
   SN4
   SN41
   SN42
   WN
   WN1
   WN11
   WN12
   WN2
   WN3
   WN4
   WN41
   WN42
  ;

   IF FINISHED THEN GO TO FINISHED;

   RETURN;

FINISHED:
   FILE "D:\KEITH\DOD99-8676\RES-RATE\TABLE02&FORM..OUT" LRECL=132;
   PUT; PUT;
   PUT @001 "TABLE 2: OVERALL RESPONSE RATES SUMMARY";
   PUT @001 "WRITTEN BY KEITH RATHBUN, 02-03-2000, TASK: 8676-002";
   PUT;
   PUT "SUMMARY OF GROUP COUNTS: FORM &FORM";
   PUT;
   PUT @060 "UNWEIGHTED COUNT"
       @100 "WEIGHTED COUNT"
       ;
   PUT @050 'FLR'
       @060 'FCR'
       @070 'FRR'
       @090 'FLR'
       @100 'FCR'
       @110 'FRR'
       ;
   %INCLUDE "D:\KEITH\DOD99-8676\SAS\TABLE02.IN2";
RUN;
%MEND PROCESS;

%PROCESS(KEEPERSC,C,"D:\KEITH\DOD99-8676\DATA\FMTLIBC");
```

```
RUN;

*********************************************************************
* Process Single Domain where domain1 is the variable of interest
*********************************************************************
;

%MACRO PROCESS1(DOMAIN1,INPT,FORM,LIB);

LIBNAME LIBRARY &LIB;

PROC SORT DATA=OUT.&INPT OUT=&INPT TAGSORT; BY &DOMAIN1; RUN;

DATA _NULL_;
   SET &INPT;
   BY &DOMAIN1;
   FILE "D:\KEITH\DOD99-8676\RES-RATE\&DOMAIN1..OUT" LRECL=132;
   LENGTH VARNAME1 $8;
   LENGTH VARIABLE $30;
   CALL VNAME(&DOMAIN1,VARNAME1);
   VARIABLE = VARNAME1;
   %INCLUDE 'D:\KEITH\DOD99-8676\SAS\TABLE02.IN1';
   IF LAST.&DOMAIN1 THEN DO;
      PUT @001 &DOMAIN1 @;
      %INCLUDE 'D:\KEITH\DOD99-8676\SAS\TABLE02.IN2';
   END; * DOMAIN;
RUN;
%MEND PROCESS1;

**********************************************************************************
* Process Double Domain where domain1/domain2 are the variables of interest
**********************************************************************************
;
%MACRO PROCESS2(DOMAIN1,DOMAIN2,INPT,FORM,LIB);

LIBNAME LIBRARY &LIB;

PROC SORT DATA=OUT.&INPT OUT=&INPT TAGSORT; BY &DOMAIN1 &DOMAIN2; RUN;

DATA _NULL_;
   SET &INPT;
   BY &DOMAIN1 &DOMAIN2;
   FILE "D:\KEITH\DOD99-8676\RES-RATE\&DOMAIN1&DOMAIN2..OUT" LRECL=132;
   LENGTH VARNAME1 $8;
   LENGTH VARNAME2 $8;
   LENGTH VARIABLE $30;
   CALL VNAME(&DOMAIN1,VARNAME1);
   CALL VNAME(&DOMAIN2,VARNAME2);
   VARIABLE = VARNAME1 || " " || VARNAME2;
   %INCLUDE 'D:\KEITH\DOD99-8676\SAS\TABLE02.IN1';
   IF LAST.&DOMAIN2 THEN DO;
      PUT @001 &DOMAIN1 @;
      PUT @025 &DOMAIN2 @;
      %INCLUDE 'D:\KEITH\DOD99-8676\SAS\TABLE02.IN2';
      SN    = 0;
      SN1   = 0;
      SN11  = 0;
      SN12  = 0;
      SN2   = 0;
      SN3   = 0;
      SN4   = 0;
      SN41  = 0;
      SN42  = 0;
      WN   = 0;
      WN1  = 0;
      WN11 = 0;
      WN12 = 0;
      WN2  = 0;
      WN3  = 0;
      WN4  = 0;
      WN41 = 0;
      WN42 = 0;
   END; * DOMAIN;
```

```
RUN;
%MEND PROCESS2;

*************************************************************************
* Process Triple Domain where domain1-3 are the variables of interest
*************************************************************************
;
%MACRO PROCESS3(DOMAIN1,DOMAIN2,DOMAIN3,INPT,FORM,LIB);

LIBNAME LIBRARY &LIB;

PROC SORT DATA=OUT.&INPT OUT=&INPT TAGSORT; BY &DOMAIN1 &DOMAIN2 &DOMAIN3; RUN;

DATA _NULL_;
   SET &INPT;
   BY &DOMAIN1 &DOMAIN2 &DOMAIN3;
   FILE "D:\KEITH\DOD99-8676\RES-RATE\&DOMAIN1&DOMAIN2&DOMAIN3..OUT" LRECL=132;
   LENGTH VARNAME1 $8;
   LENGTH VARNAME2 $8;
   LENGTH VARNAME3 $8;
   LENGTH VARIABLE $30;
   CALL VNAME(&DOMAIN1,VARNAME1);
   CALL VNAME(&DOMAIN2,VARNAME2);
   CALL VNAME(&DOMAIN3,VARNAME3);
   VARIABLE = VARNAME1 || " " || VARNAME2 || " " || VARNAME3;
   %INCLUDE 'D:\KEITH\DOD99-8676\SAS\TABLE02.IN1';
   IF LAST.&DOMAIN3 THEN DO;
      PUT @001 &DOMAIN1 @;
      PUT @018 &DOMAIN2 @;
      PUT @038 &DOMAIN3 @;
      %INCLUDE 'D:\KEITH\DOD99-8676\SAS\TABLE02.IN2';
      SN    = 0;
      SN1   = 0;
      SN11  = 0;
      SN12  = 0;
      SN2   = 0;
      SN3   = 0;
      SN4   = 0;
      SN41  = 0;
      SN42  = 0;
      WN   = 0;
      WN1  = 0;
      WN11 = 0;
      WN12 = 0;
      WN2  = 0;
      WN3  = 0;
      WN4  = 0;
      WN41 = 0;
      WN42 = 0;
   END; * DOMAIN;
RUN;
%MEND PROCESS3;

****************************************************
* PROCESS SINGLE DOMAIN TABULATION - FORM C
****************************************************
;
%PROCESS1(SUPREGP,  KEEPERSC, "FORM C",
         "D:\KEITH\DOD99-8676\DATA\FMTLIBC");
%PROCESS1(ENGROUPP, KEEPERSC, "FORM C",
         "D:\KEITH\DOD99-8676\DATA\FMTLIBC");
%PROCESS1(AGEGROUP, KEEPERSC, "FORM C",
         "D:\KEITH\DOD99-8676\DATA\FMTLIBC");

****************************************************
* PROCESS DOUBLE DOMAIN TABULATION - FORM C
****************************************************
;
*%PROCESS2(XXXXXXXX,XXXXXXXX,KEEPERSC, "FORM C",
         "D:\KEITH\DOD99-8676\DATA\FMTLIBC");
*%PROCESS2(XXXXXXXX,XXXXXXXX,KEEPERSC, "FORM C",
         "D:\KEITH\DOD99-8676\DATA\FMTLIBC");
*%PROCESS2(XXXXXXXX,XXXXXXXX,KEEPERSC, "FORM C",
```

```
         "D:\KEITH\DOD99-8676\DATA\FMTLIBC");
*%PROCESS2(XXXXXXXX,XXXXXXXX,KEEPERSC, "FORM C",
         "D:\KEITH\DOD99-8676\DATA\FMTLIBC");
```

8. DEVELOPMENT OF WEIGHTS

```
*****************************************************************************

* PROGRAM: WEIGHT03.SAS
* TASK:    1999 DOD HEALTH CARE SURVEY ANALYSIS (8676-100)
* PURPOSE: BUILD AND ASSIGN ADJUSTED WEIGHTS.
*          WEIGHTS FOR DOD SURVEY.
*          DOD HEALTH CARE SURVEY FILE.
*          REQUESTED BY DON JANG.
* WRITTEN: 11/09/99 BY KEITH RATHBUN
*
* INPUTS:  1) KEEPERSC.SD2 - Reduced/Deduped 1999 DOD survey dataset - Form C
*
* OUTPUTS: 1) WEIGH03C.SD2 - Form C dataset with adjusted weight (AWT)
*
*****************************************************************************
*;
LIBNAME IN       "D:\KEITH\DOD99-8676\DATA";
LIBNAME OUT      "D:\KEITH\DOD99-8676\DATA";
OPTIONS PS=79 LS=132 COMPRESS=YES NOCENTER;

*****************************************************************************
* Calculate adjusted weight based on user-specified domain.
*****************************************************************************;
%MACRO PROCESS(DOMAIN,FORM,INPT);
PROC SORT DATA=IN.&INPT(KEEP= MPRID FNSTATUS BWT99 &DOMAIN POSTSTR)
          OUT=&INPT(KEEP= MPRID FNSTATUS BWT99 &DOMAIN POSTSTR)
          TAGSORT;
BY &DOMAIN;
RUN;

*****************************************************************************
* Calculate adjustment factor A1 for each cell
*****************************************************************************;
DATA CELLSA1  (KEEP=SUMBWT SUMG1-SUMG3 A1 CELLNAME CELLCNT)
     MPRIDSA1 (KEEP=CELLNAME MPRID FNSTATUS BWT99 &DOMAIN)
    ;
   SET &INPT;
   BY &DOMAIN;

   LENGTH CELLNAME $25;
   CELLNAME = PUT(&DOMAIN,5.);

   IF FIRST.&DOMAIN THEN DO;
      CELLCNT = 0;
      SUMBWT  = 0.0;
      SUMG1   = 0.0;
      SUMG2   = 0.0;
      SUMG3   = 0.0;
      A1 = 0.0;
   END;
   CELLCNT + 1;

   ****************************************************
   * Accumulate total weight sum
   ****************************************************;
   SUMBWT + BWT99;
   ****************************************************
   * Accumulate group 1 weight sum
   ****************************************************;
   IF FNSTATUS IN(11,12) THEN SUMG1 + BWT99;
   ****************************************************
   * Accumulate group 2 weight sum
   ****************************************************;
   ELSE IF FNSTATUS = 20 THEN SUMG2 + BWT99;
   ****************************************************
   * Accumulate group 3 weight sum
   ****************************************************;
   ELSE IF FNSTATUS = 30 THEN SUMG3 + BWT99;

   RETAIN SUMBWT SUMG1-SUMG3 A1 CELLNAME CELLCNT MPRID;

   IF LAST.&DOMAIN THEN DO;
```

```
      A1 = SUMBWT/(SUMG1 + SUMG2 + SUMG3);
      OUTPUT CELLSA1;
      CELLCNT = 0;
      SUMBWT  = 0.0;
      SUMG1   = 0.0;
      SUMG2   = 0.0;
      SUMG3   = 0.0;
   END; * DOMAIN;
   OUTPUT MPRIDSA1;
RUN;

***************************************************************************
* Calculate adjustment factor A2 for each cell
***************************************************************************;
DATA CELLSA2 (KEEP=CELLNAME CELLCNT A1 A2 NUMER DENOM);
   MERGE MPRIDSA1 CELLSA1;
   BY CELLNAME;

   IF FIRST.CELLNAME THEN DO;
      A2 = 0.0;
      NUMER = 0.0;
      DENOM = 0.0;
   END;
   RETAIN NUMER DENOM A2;

   IF FNSTATUS IN(11,12) OR FNSTATUS = 20 THEN NUMER + BWT99*A1;
   IF FNSTATUS  = 11 THEN DENOM + BWT99*A1;

   IF LAST.CELLNAME THEN DO;
      A2 = NUMER/DENOM;
      OUTPUT CELLSA2;
   END;
RUN;

***************************************************************************
* Calculate Adjusted Weight
***************************************************************************;
DATA WEIGHT03;
   MERGE CELLSA2 MPRIDSA1;
   BY CELLNAME;
   IF FNSTATUS = 11 THEN
      AWT = A1*A2*BWT99;
   ELSE IF FNSTATUS IN(12,20,41,42) THEN
      AWT = 0;
   ELSE IF FNSTATUS = 30 THEN
      AWT = A1*BWT99;
   LABEL AWT = 'Adjusted Weight Factor';
   KEEP MPRID AWT A1 A2;
RUN;

***************************************************************************
* Sort the original data
***************************************************************************;
PROC SORT DATA=&INPT OUT=&INPT TAGSORT; BY MPRID; RUN;

***************************************************************************
* Sort the adjusted weight (AWT) data
***************************************************************************;
PROC SORT DATA=WEIGHT03 TAGSORT; BY MPRID; RUN;

***************************************************************************
* Append adjusted weight (AWT) variable
***************************************************************************;
DATA OUT.WEIGH03&FORM;
   MERGE WEIGHT03 &INPT;
   BY MPRID;
RUN;

TITLE1 "1999 DOD SURVEY OF HEALTH BENES - FORM &FORM";
TITLE2 "PROGRAM: WEIGHT03.SAS, INPUT: KEEPERS&FORM..SD2, OUTPUT: WEIGH03&FORM..SD2";

TITLE3 'DOMAIN TABULATIONS - UNWEIGHTED WHERE FNSTATUS=11';
PROC FREQ DATA=OUT.WEIGH03&FORM;
```

```
WHERE FNSTATUS = 11;
TABLES &DOMAIN /MISSING LIST;
RUN;

TITLE3 'DOMAIN TABULATIONS - WEIGHTED BY AWT';
PROC FREQ DATA=OUT.WEIGH03&FORM;
WEIGHT AWT;
TABLES  &DOMAIN /MISSING LIST;
RUN;

TITLE3 'DOMAIN TABULATIONS - WEIGHTED BY BWT99';
PROC FREQ DATA=OUT.WEIGH03&FORM;
WEIGHT BWT99;
TABLES  &DOMAIN /MISSING LIST;
RUN;

TITLE3 'WEIGHT TABULATIONS - UNWEIGHTED WHERE FNSTATUS=11';
PROC SORT DATA=OUT.WEIGH03&FORM OUT=WEIGH03&FORM; BY &DOMAIN; RUN;

PROC MEANS DATA=WEIGH03&FORM NOPRINT;
BY &DOMAIN;
WHERE FNSTATUS = 11;
VAR A1 A2;
OUTPUT OUT=TEMP MEAN=A1MEAN A2MEAN N=NA1 NA2;
RUN;
PROC PRINT; RUN;

%MEND PROCESS;

***************************************************************************
* Calculate adjusted weight based on user-specified parameters.
***************************************************************************;
%PROCESS(STRATUM,C,KEEPERSC)

***************************************************************************
*
* PROGRAM: WEIGHT04.SAS
* TASK:    1999 DOD HEALTH CARE SURVEY ANALYSIS (8676-200)
* PURPOSE: BUILD AND ASSIGN FINAL WEIGHTS - POST STRATIFICATION - FORM A/C.
*          WEIGHTS FOR DOD SURVEY.
*          DOD HEALTH CARE SURVEY FILE.
*          REQUESTED BY DON JANG.
* WRITTEN: 12/30/99 BY KEITH RATHBUN
*
* INPUTS:   1) WEIGH03A.SD2 - Adjusted Weights file - Form A
*           2) WEIGH03C.SD2 - Adjusted Weights file - Form C
*
* OUTPUTS   1) WEIGH04A.SD2 - Final Weights file - Form A
*           2) WEIGH04C.SD2 - Final Weights file - Form C
*
***************************************************************************
*;
LIBNAME IN   "D:\KEITH\DOD99-8676\DATA";
LIBNAME OUT  "D:\KEITH\DOD99-8676\DATA";
LIBNAME LIBRARY  "D:\KEITH\DOD99-8676\DATA\FMTLIB";
OPTIONS PS=79 LS=132 COMPRESS=YES ERRORS=0 NOCENTER;

%MACRO PROCESS(DOMAIN=,FORM=);

*********************************************************************
* Sort the adjusted weights file by user-specified domains
*********************************************************************;
PROC SORT DATA=IN.WEIGH03&FORM
          OUT=WEIGHT03(KEEP=FNSTATUS MPRID AWT &DOMAIN)
         ;
     BY &DOMAIN;
RUN;

*********************************************************************
* Assign cell names and calculate the sum of AWT
*********************************************************************;
DATA CELLS  (KEEP=SUMAWT SUMFN11 &DOMAIN)
     MPRIDS (KEEP=MPRID FNSTATUS AWT &DOMAIN)
```

```
    ;
   SET WEIGHT03;
   BY &DOMAIN;

   IF FIRST.&DOMAIN THEN DO;
      SUMAWT   = 0.0;
      SUMFN11  = 0;
   END;

   ********************************************************************
   * Accumulate sum of adjusted weight
   ********************************************************************;
   SUMAWT + AWT;

   ********************************************************************
   * COUNT the FNSTATUS = 11 within each DOMAIN
   ********************************************************************;
   IF FNSTATUS = 11 THEN SUMFN11 + 1;

   RETAIN SUMAWT SUMFN11;

   IF LAST.&DOMAIN THEN DO;
      OUTPUT CELLS;
      SUMAWT  = 0.0;
      SUMFN11  = 0;
   END; * DOMAIN;
   OUTPUT MPRIDS;
RUN;

***********************************************************************
* Merge the population counts and calculate the adjusted population (AP)
***********************************************************************;
PROC SORT DATA=IN.COUNT&FORM OUT=COUNT&FORM; BY &DOMAIN; RUN;

DATA AP;
   MERGE COUNT&FORM CELLS;
   BY &DOMAIN;
   AP = POP/SUMAWT;
RUN;

***********************************************************************
* Merge the adjusted population and calculate the final weight (WRWT99)
***********************************************************************;
DATA WEIGHT04;
   MERGE AP(IN=IN1) MPRIDS(IN=IN2);
   BY &DOMAIN;

   IF IN2 THEN DO;
      WRWT99 = AP*AWT;
      OUTPUT;
   END;

   LABEL WRWT99   = 'Final Weight';
   LABEL AP       = 'Adjusted population';
   LABEL POP      = 'DEERS population by CELLNAME for weights';
   LABEL SUMFN11  = 'COUNT of FNSTATUS=11 within CELLNAME';

   KEEP FNSTATUS WRWT99 AWT AP MPRID POP SUMFN11 &DOMAIN;
RUN;

TITLE1 "1999 DOD SURVEY OF HEALTH BENES - FORM &FORM";
TITLE2 "PROGRAM: WEIGHT04.SAS, INPUT: WEIGH03&FORM..SD2, OUTPUT: WEIGH04&FORM..SD2";
TITLE3 "WEIGHT TABULATIONS - UNWEIGHTED";

PROC MEANS DATA=WEIGHT04 NOPRINT;
   VAR POP WRWT99 AP SUMFN11;
   BY &DOMAIN;
   OUTPUT OUT=STATS(KEEP=POSTSTR  DEERSPOP POPCNT APM FN11CNT)
                    SUM= DUMMY1   POPCNT  DUMMY2  DUMMY3
                    MEAN=DUMMY4   DUMMY5  APM     DUMMY6
                    MAX= DEERSPOP DUMMY7  DUMMY8 FN11CNT;
RUN;
```

```
PROC PRINT;
   SUM DEERSPOP POPCNT FN11CNT;
RUN;

*********************************************************************
* Sort the original data and append the final weight (WRWT99)
*********************************************************************;
PROC SORT DATA=IN.WEIGH03&FORM OUT=WEIGHT03 TAGSORT; BY MPRID; RUN;
PROC SORT DATA=WEIGHT04 TAGSORT; BY MPRID; RUN;

DATA OUT.WEIGH04&FORM;
   MERGE WEIGHT03 WEIGHT04;
   BY MPRID;
RUN;

%MEND;

*%PROCESS(DOMAIN=xxxxxxx,FORM=A);
%PROCESS(DOMAIN=POSTSTR,FORM=C);



***************************************************************************
*
* PROGRAM: WEIGHT05.SAS
* TASK:    1999 DOD HEALTH CARE SURVEY ANALYSIS (8676-200)
* PURPOSE: BUILD AND ASSIGN JK WEIGHTS - POST STRATIFICATION - FORM A/C.
*          WEIGHTS FOR DOD SURVEY.
*          DOD HEALTH CARE SURVEY FILE.
*          REQUESTED BY DON JANG.
* WRITTEN: 12/30/99 BY KEITH RATHBUN
*
* INPUTS:   1) WEIGH04A.SD2 - Final Weights file - Form A
*           2) WEIGH04C.SD2 - Final Weights file - Form C
*
* OUTPUTS   1) WEIGH05A.SD2 - JackKnife (JK) Weights file - Form A
*           2) WEIGH05C.SD2 - JackKnife (JK) Weights file - Form C
*
***************************************************************************
*;
LIBNAME IN   "D:\KEITH\DOD99-8676\DATA";
LIBNAME OUT  "D:\KEITH\DOD99-8676\DATA";
LIBNAME LIBRARY  "D:\KEITH\DOD99-8676\DATA\FMTLIBA";
OPTIONS PS=79 LS=132 COMPRESS=YES ERRORS=0 NOCENTER;

%MACRO PROCESS(DOMAIN1, DOMAIN2, FORM);

*********************************************************************
* Sort the final weights file by user-specified domains
*********************************************************************;
PROC SORT DATA=IN.WEIGH04&FORM
          OUT=WEIGHT04(KEEP=FNSTATUS MPRID BWT99 &DOMAIN1 &DOMAIN2)
         ;
     BY &DOMAIN1;
RUN;

*********************************************************************
* Append SUBSET index (I) to each observation
*********************************************************************;
DATA SUBSETS;
   SET WEIGHT04;
   BY &DOMAIN1;

   IF _N_ = 1 OR MOD(_N_-1,40) = 0 THEN SUBSET = 1;
   ELSE SUBSET + 1;

   RETAIN SUBSET;
   BBWT = BWT99*(40/39);
RUN;

*********************************************************************
*********************************************************************
* Generate JackKnife/replicated weights WRWT01-WRWT40
```

```
*********************************************************************
*********************************************************************;
%DO I = 1 %TO 40;

DATA SUBSET;
   SET SUBSETS;
   IF &I = SUBSET THEN DELETE; *Remove the current subset;
RUN;

**************************************************************************
* Calculate adjustment factor A1 for each cell
**************************************************************************;
DATA CELLSA1  (KEEP=SUMBBWT SUMG1-SUMG3 A1 CELLNAME CELLCNT)
     MPRIDSA1 (KEEP=CELLNAME MPRID FNSTATUS BBWT &DOMAIN1 &DOMAIN2)
    ;
   SET SUBSET;
   BY &DOMAIN1;

   LENGTH CELLNAME $25;
   CELLNAME = PUT(&DOMAIN1,5.);

   IF FIRST.&DOMAIN1 THEN DO;
      CELLCNT = 0;
      SUMBBWT = 0.0;
      SUMG1   = 0.0;
      SUMG2   = 0.0;
      SUMG3   = 0.0;
      A1 = 0.0;
   END;
   CELLCNT + 1;

   ****************************************************
   * Accumulate total weight sum
   ****************************************************;
   SUMBBWT + BBWT;
   ****************************************************
   * Accumulate group 1 weight sum
   ****************************************************;
   IF FNSTATUS IN(11,12) THEN SUMG1 + BBWT;
   ****************************************************
   * Accumulate group 2 weight sum
   ****************************************************;
   ELSE IF FNSTATUS = 20 THEN SUMG2 + BBWT;
   ****************************************************
   * Accumulate group 3 weight sum
   ****************************************************;
   ELSE IF FNSTATUS = 30 THEN SUMG3 + BBWT;

   RETAIN SUMBBWT SUMG1-SUMG3 A1 CELLNAME CELLCNT MPRID;

   IF LAST.&DOMAIN1 THEN DO;
      A1 = SUMBBWT/(SUMG1 + SUMG2 + SUMG3);
      OUTPUT CELLSA1;
      CELLCNT = 0;
      SUMBBWT = 0.0;
      SUMG1   = 0.0;
      SUMG2   = 0.0;
      SUMG3   = 0.0;
   END; * DOMAIN;
   OUTPUT MPRIDSA1;
RUN;

**************************************************************************
* Calculate adjustment factor A2 for each cell
**************************************************************************;
DATA CELLSA2 (KEEP=CELLNAME CELLCNT A1 A2 NUMER DENOM);
   MERGE MPRIDSA1 CELLSA1;
   BY CELLNAME;

   IF FIRST.CELLNAME THEN DO;
      A2 = 0.0;
      NUMER = 0.0;
      DENOM = 0.0;
```

```
   END;
   RETAIN NUMER DENOM A2;

   IF FNSTATUS IN(11,12) OR FNSTATUS = 20 THEN NUMER + BBWT*A1;
   IF FNSTATUS  = 11 THEN DENOM + BBWT*A1;

   IF LAST.CELLNAME THEN DO;
      A2 = NUMER/DENOM;
      OUTPUT CELLSA2;
   END;
RUN;

*************************************************************************
* Calculate Adjusted Weight
*************************************************************************;
DATA ADJWGT;
   MERGE CELLSA2 MPRIDSA1;
   BY CELLNAME;
   IF FNSTATUS = 11 THEN
      AWT = A1*A2*BBWT;
   ELSE IF FNSTATUS IN(12,20,41,42) THEN
      AWT = 0;
   ELSE IF FNSTATUS = 30 THEN
      AWT = A1*BBWT;
   KEEP MPRID FNSTATUS AWT BBWT &DOMAIN1 &DOMAIN2;
RUN;

**********************************************************************
* Begin final weight code
**********************************************************************
* Assign cell names and calculate the sum of AWT
**********************************************************************;

PROC SORT DATA=ADJWGT; BY &DOMAIN2; RUN;

DATA CELLS  (KEEP=SUMAWT &DOMAIN2)
     MPRIDS (KEEP=MPRID FNSTATUS AWT &DOMAIN1 &DOMAIN2)
    ;
   SET ADJWGT;
   BY &DOMAIN2;

   IF FIRST.&DOMAIN2 THEN DO;
      SUMAWT  = 0.0;
   END;

   **********************************************************************
   * Accumulate sum of adjusted weight
   **********************************************************************;
   SUMAWT + AWT;

   RETAIN SUMAWT;

   IF LAST.&DOMAIN2 THEN DO;
      OUTPUT CELLS;
      SUMAWT  = 0.0;
   END; * DOMAIN;
   OUTPUT MPRIDS;
RUN;

**********************************************************************
* Merge the population counts and calculate the adjusted population (AP)
**********************************************************************;
PROC SORT DATA=IN.COUNT&FORM OUT=COUNT&FORM; BY &DOMAIN2; RUN;

DATA AP;
   MERGE COUNT&FORM CELLS;
   BY &DOMAIN2;
   AP = POP/SUMAWT;
RUN;

**********************************************************************
* Merge the adjusted population and calculate JackKnife Weights
* (WRWT1-WRWT40)
```

```
*************************************************************************;
DATA SUBSET&I(KEEP=MPRID SUBSET JKWEIGHT);
   MERGE AP(IN=IN1) MPRIDS(IN=IN2);
   BY &DOMAIN2;

   SUBSET = &I;
   IF IN2 THEN DO;
      JKWEIGHT = AP*AWT;
      OUTPUT;
   END;

RUN;
PROC SORT DATA=SUBSET&I; BY MPRID; RUN;

*************************************************************************
*************************************************************************
* End of JackKnife/replicated weights WRWT01-WRWT40 assignments
*************************************************************************
*************************************************************************;
%END;

*************************************************************************
* Combine all of the JackKnife weight subsets by MPRID
*************************************************************************;
DATA ALLSETS;
   SET SUBSET1   SUBSET2   SUBSET3   SUBSET4   SUBSET5
       SUBSET6   SUBSET7   SUBSET8   SUBSET9   SUBSET10
       SUBSET11  SUBSET12  SUBSET13  SUBSET14  SUBSET15
       SUBSET16  SUBSET17  SUBSET18  SUBSET19  SUBSET20
       SUBSET21  SUBSET22  SUBSET23  SUBSET24  SUBSET25
       SUBSET26  SUBSET27  SUBSET28  SUBSET29  SUBSET30
       SUBSET31  SUBSET32  SUBSET33  SUBSET34  SUBSET35
       SUBSET36  SUBSET37  SUBSET38  SUBSET39  SUBSET40
   ;
   BY MPRID;
   ARRAY JKWT(40) WRWT1-WRWT40; RETAIN WRWT1-WRWT40;
   IF FIRST.MPRID THEN DO;
      DO I = 1 TO 40; DROP I;
         JKWT(I) = . ;
      END;
   END;
   JKWT(SUBSET) = JKWEIGHT;
   IF LAST.MPRID THEN OUTPUT;
   KEEP MPRID WRWT1-WRWT40;
RUN;

*************************************************************************
* Sort the original data, get the final weight (WRWT99), append the
* JackKnife/Replicated weights (WRWT1-WRWT40), and label variables.
*************************************************************************;
PROC SORT DATA=IN.WEIGH04&FORM
          OUT=WEIGHT04;
     BY MPRID;
RUN;

DATA OUT.WEIGH05&FORM;
   MERGE WEIGHT04 ALLSETS;
   BY MPRID;
   LABEL
      MPRID  = 'MPR ID Number'
      WRWT1  = 'Replicated/JackKnife Weight 1'
      WRWT2  = 'Replicated/JackKnife Weight 2'
      WRWT3  = 'Replicated/JackKnife Weight 3'
      WRWT4  = 'Replicated/JackKnife Weight 4'
      WRWT5  = 'Replicated/JackKnife Weight 5'
      WRWT6  = 'Replicated/JackKnife Weight 6'
      WRWT7  = 'Replicated/JackKnife Weight 7'
      WRWT8  = 'Replicated/JackKnife Weight 8'
      WRWT9  = 'Replicated/JackKnife Weight 9'
      WRWT10 = 'Replicated/JackKnife Weight 10'
      WRWT11 = 'Replicated/JackKnife Weight 11'
      WRWT12 = 'Replicated/JackKnife Weight 12'
      WRWT13 = 'Replicated/JackKnife Weight 13'
```

```
          WRWT14 = 'Replicated/JackKnife Weight 14'
          WRWT15 = 'Replicated/JackKnife Weight 15'
          WRWT16 = 'Replicated/JackKnife Weight 16'
          WRWT17 = 'Replicated/JackKnife Weight 17'
          WRWT18 = 'Replicated/JackKnife Weight 18'
          WRWT19 = 'Replicated/JackKnife Weight 19'
          WRWT20 = 'Replicated/JackKnife Weight 20'
          WRWT21 = 'Replicated/JackKnife Weight 21'
          WRWT22 = 'Replicated/JackKnife Weight 22'
          WRWT23 = 'Replicated/JackKnife Weight 23'
          WRWT24 = 'Replicated/JackKnife Weight 24'
          WRWT25 = 'Replicated/JackKnife Weight 25'
          WRWT26 = 'Replicated/JackKnife Weight 26'
          WRWT27 = 'Replicated/JackKnife Weight 27'
          WRWT28 = 'Replicated/JackKnife Weight 28'
          WRWT29 = 'Replicated/JackKnife Weight 29'
          WRWT30 = 'Replicated/JackKnife Weight 30'
          WRWT31 = 'Replicated/JackKnife Weight 31'
          WRWT32 = 'Replicated/JackKnife Weight 32'
          WRWT33 = 'Replicated/JackKnife Weight 33'
          WRWT34 = 'Replicated/JackKnife Weight 34'
          WRWT35 = 'Replicated/JackKnife Weight 35'
          WRWT36 = 'Replicated/JackKnife Weight 36'
          WRWT37 = 'Replicated/JackKnife Weight 37'
          WRWT38 = 'Replicated/JackKnife Weight 38'
          WRWT39 = 'Replicated/JackKnife Weight 39'
          WRWT40 = 'Replicated/JackKnife Weight 40'
       ;
    RUN;

    TITLE1 "1999 DOD Health Survey Final/Replicated Weights (8676-200)";
    TITLE2 "Program Name: WEIGHT05.SAS By Keith Rathbun";
    TITLE3 "Program Output: WEIGH05&FORM..SD2";

    PROC CONTENTS DATA=OUT.WEIGH05&FORM;

    PROC MEANS DATA=OUT.WEIGH05&FORM;
    VAR  WRWT99 WRWT1-WRWT40;
    RUN;

    PROC SORT DATA=OUT.WEIGH05&FORM; BY &DOMAIN2; RUN;
    PROC MEANS DATA=OUT.WEIGH05&FORM NOPRINT;
       VAR POP WRWT1;
       BY &DOMAIN2;
       OUTPUT OUT=STATS(KEEP=&DOMAIN2 DEERSPOP POPCNT)
                        SUM= DUMMY1   POPCNT
                        MAX= DEERSPOP DUMMY2;
    RUN;
    PROC PRINT; RUN;

    PROC PRINT DATA=OUT.WEIGH05&FORM(OBS=50);
    VAR MPRID BWT99 AWT WRWT99 WRWT1-WRWT40;
    RUN;

    %MEND;

    %PROCESS(STRATUM, POSTSTR, C);

*****************************************************************************
*
* PROGRAM: FINAL02.SAS
* TASK:    1999 DOD HEALTH CARE SURVEY ANALYSIS (8687-100)
* PURPOSE: RECODE MISSING REPLICATED WEIGHTS TO ZERO.
*          DOD HEALTH CARE SURVEY FILE.
* WRITTEN: 04/10/2000 BY KEITH RATHBUN
*
* INPUTS:  1) WEIGH05A.SD2 - Replicated Weights file - Form A
*          2) WEIGH05C.SD2 - Replicated Weights file - Form C
*
* OUTPUTS: 1) FINAL02A.SD2 - Replicated Weights file with recodes - Form A
*          2) FINAL02C.SD2 - Replicated Weights file with recodes - Form C
*
*****************************************************************************
```

```
*;
LIBNAME IN      "D:\KEITH\DOD99-8676\DATA";
LIBNAME OUT     "D:\KEITH\DOD99-8676\DATA";
LIBNAME LIBRARY "D:\KEITH\DOD99-8676\DATA\FMTLIBA";
OPTIONS PS=79 LS=132 COMPRESS=YES ERRORS=2 NOCENTER;

%MACRO PROCESS(FORM);

PROC SORT DATA=OUT.WEIGH05&FORM; BY MPRID; RUN;
DATA OUT.FINAL02&FORM;
   SET OUT.WEIGH05&FORM;
   BY MPRID;

   ARRAY WGTS(40) WRWT1-WRWT40;
   DO I = 1 TO 40; DROP I;
      IF WGTS(I) EQ . THEN WGTS(I) = 0;
   END;

   KEEP MPRID BWT99 WRWT99 WRWT1-WRWT40;
RUN;

TITLE1 "1999 DOD SURVEY OF HEALTH BENES - FORM &FORM";
TITLE2 "PROGRAM: FINAL02.SAS, OUTPUT: FINAL02&FORM..SD2";

PROC CONTENTS DATA=OUT.FINAL02&FORM; RUN;

PROC MEANS DATA=OUT.FINAL02&FORM;
VAR BWT99 WRWT99 WRWT1-WRWT40;
RUN;

%MEND PROCESS;

*%PROCESS(A);
%PROCESS(C);
RUN;
```

9. FINAL MERGE

```
*************************************************************************
*
* PROGRAM: ADDWGTS.SAS
* TASK:    1999 DOD HEALTH CARE SURVEY ANALYSIS (8687-100)
* PURPOSE: MERGE THE FINAL WEIGHTS FILE WITH THE FINAL
*          QUESTIONNAIRE/SAMPLE FILE
*
* WRITTEN: 02/24/2000 BY KEITH RATHBUN
*
* INPUTS:  1) FINAL02A.SD2 - Final/Replicated Weights file - FORM A
*          2) FINAL02C.SD2 - Final/Replicated Weights file - FORM C
*          3) HCS99A_n.SD2 - Final FORM A Questionnaire/Sample File
*          4) HCS99C_n.SD2 - Final FORM C Questionnaire/Sample File
*             where n = final dataset suffix
*
* OUTPUTS: 1) HCS99A_n+1.SD2 - Final FORM A Questionnaire/Sample File
*             combined with Final/Replicated Weights file - FORM A
*          2) HCS99C_n+1.SD2 - Final FORM C Questionnaire/Sample File
*             combined with Final/Replicated Weights file - FORM C
*             where n+1 = final dataset suffix
*
*************************************************************************
;
LIBNAME IN  "D:\KEITH\DOD99-8676\DATA";
LIBNAME OUT "D:\KEITH\DOD99-8676\DATA";
LIBNAME LIBRARY "D:\KEITH\DOD99-8676\DATA\FMTLIBA";

OPTIONS PS=79 LS=132 COMPRESS=YES NOCENTER;

%MACRO PROCESS(DSNI_1=,DSNI_2=,DSNO=);
*************************************************************************
* Merge the final weights file with the final Questionnaire/Sample file
*************************************************************************;
PROC SORT DATA=IN.&DSNI_1 OUT=&DSNI_1; BY MPRID; RUN;
%IF "&DSNI_1" = "FINAL02C" %THEN %DO;
    PROC SORT DATA=IN.&DSNI_2 OUT=&DSNI_2(DROP=BWT99); BY MPRID; RUN;
%END;
%ELSE %DO;
    PROC SORT DATA=IN.&DSNI_2 OUT=&DSNI_2; BY MPRID; RUN;
%END;

DATA OUT.&DSNO;
   MERGE &DSNI_2(IN=IN2)
         &DSNI_1(IN=IN1);
   BY MPRID;
   IF IN1 AND IN2;
   LABEL KEYCOUNT = "# of Key Questions Answered";
RUN;

TITLE1 "1999 DOD Health Survey Post-Stratification (8687-100)";
TITLE2 "Program Name: ADDWGTS.SAS By Keith Rathbun";
TITLE3 "Program Inputs: &DSNI_1..SD2 -- &DSNI_2..SD2";
TITLE4 "Program Outputs: &DSNO..SD2";
PROC CONTENTS; RUN;

%MEND PROCESS;

%PROCESS(DSNI_1=FINAL02A, DSNI_2=HCS99A_4, DSNO=HCS99A_5);
*%PROCESS(DSNI_1=FINAL02C, DSNI_2=HCS99C_4, DSNO=HCS99C_5);
```