APPENDIX E

STATISTICAL SPECIFICATIONS FOR CHILD REPORT CARDS

1. Create Variables for Child TRICARE Consumer Reports

```
*****************************************************************
*  Project:  DoD - 1999 Child Report Cards
*  Program:  STEP1C.SAS
*  Purpose:  Create dummy variables used in Child Report Card
*                   Create a Female dummy variable
*                   Create an Education dummy variable
*                   Create 3 super region dummies.
*
*            We require the most desired code to be the highest value.
*            Recode the dependent variables into:
*                   1 - the least desirable value
*                   2 - the 2nd least desirable value
*                   3 - the most desirable value
*                   . - missing
*
*            Create 7 variables GROUP1 - GROUP7;
*                  IF XINS_COV = 1 THEN GROUP1 = 1;
*                  IF XENR_PCM = 1   THEN GROUP2 = 1;
*                  IF XENR_PCM = 2        THEN GROUP3 = 1;
*                  IF XINS_COV IN (2,3) THEN GROUP4 = 1;
*                  IF AGEUND6 = 1         THEN GROUP5 = 1;
*                  IF AGE0612 = 1         THEN GROUP6 = 1;
*                  IF AGE1317 = 1         THEN GROUP7 = 1;
*
*  Modified: 1) Joan James, 7-13-2000: Adapted from Adult Report Card
*               STEP1C.SAS to reflect the requirements of the Child
*               Report Card.
*
*  Create 7 files for the 7 subgroups
*****************************************************************;
OPTIONS NOCENTER LS=124 PS=74 SOURCE SOURCE2 NOFMTERR NOOVP COMPRESS=YES;
LIBNAME   OUT       "DATA";
LIBNAME   IN1       "J:\8687\DATA1999\CFINAL";

TITLE1    'Program Saved as: STEP1C.SAS';

DATA OUT.ENTIRE;
   SET IN1.HCS99C_5(KEEP=
                FORM
                MPRID
                ZAGE_D1
                SUPREGA
                XREGION
                CACSMPL
                BFGROUPP
                ENGROUPP
                STRATUM
                XINS_COV
                XENR_PCM
                WRWT99
                C99C05
                C99C13
                C99C30
                C99C31
                C99C18
                C99C20
                C99C23
                C99C32
                C99C35
                C99C36
                C99C37
                C99C39
                C99C40
                C99C33
                C99C34
                C99C49
                C99C51
                C99C56
                C99C08
                C99C09
                C99C10
                C99C60A
```

```
                C99C60B
                C99C60C
                C99C60D
                C99C21
                C99C24
                C99C27
                C99C41
                C99C07
                C99C57
                C99D61
                C99E61
                C99D72
                C99D73
                C99E73
                C99E74
                C99D74A C99D74B C99D74C C99D74D C99D74E C99D74F
                C99E75A C99E75B C99E75C C99E75D C99E75E C99E75F
                RENAME=(STRATUM=XSTRATUM ZAGE_D1=ZAGE));
    FORMAT _ALL_;
    LENGTH STRATUM 3;
    STRATUM = XSTRATUM; DROP XSTRATUM;

    ******************************************************************
    * Compute parents highest education level based on form used.
    ******************************************************************;
    ARRAY FORM1LEV(6) C99D74A C99D74B C99D74C C99D74D C99D74E C99D74F;
    ARRAY FORM2LEV(6) C99E75A C99E75B C99E75C C99E75D C99E75E C99E75F;
    IF FORM = 1 THEN DO;
       DO I = 1 TO DIM(FORM1LEV);
          IF FORM1LEV(I) > 0 THEN SREDHIGH = FORM1LEV(I);
       END;
    END;
    ELSE IF FORM = 2 THEN DO;
       DO I = 1 TO DIM(FORM2LEV);
          IF FORM2LEV(I) > 0 THEN SREDHIGH = FORM2LEV(I);
       END;
    END;
    DROP I;

RUN;

*-------------------------------------------------------;
* Create AGE, FEMALE and GROUP (Beneficiary/Enrollment)
* subsets. Create the super region dummies.
*-------------------------------------------------------;
DATA OUT.ENTIRE;
   SET OUT.ENTIRE;

   LENGTH DEFAULT = 4;
   *---------------------------------------------------;
   * Create child AGE dummies
   *---------------------------------------------------;
   IF (ZAGE NE . AND ZAGE NE 255) THEN DO;
      AGEUND6 = 0;
      AGE0612 = 0;
      AGE1317 = 0;

      IF      (ZAGE < 6)         THEN AGEUND6 = 1;
      ELSE IF (6  <= ZAGE <= 12) THEN AGE0612 = 1;
      ELSE IF (13 <= ZAGE <= 17) THEN AGE1317 = 1;
   END;

   *---------------------------------------------------;
   * Create parent AGE dummies
   *---------------------------------------------------;
   IF      FORM = 1 THEN TEMPAGE = C99D72; DROP TEMPAGE;
   ELSE IF FORM = 2 THEN TEMPAGE = C99E73;
   if 1<=tempage<=8 then do;

   AGEUND18 = 0; AGE1824 = 0; AGE2534 = 0; AGE3544 = 0;
   AGE4554  = 0; AGE5564 = 0; AGE6574 = 0; AGE75UP = 0;

   IF      TEMPAGE = 1 THEN AGEUND18 = 1;
```

```
ELSE IF TEMPAGE = 2 THEN AGE1824  = 1;
ELSE IF TEMPAGE = 3 THEN AGE2534  = 1;
ELSE IF TEMPAGE = 4 THEN AGE3544  = 1;
ELSE IF TEMPAGE = 5 THEN AGE4554  = 1;
ELSE IF TEMPAGE = 6 THEN AGE5564  = 1;
ELSE IF TEMPAGE = 7 THEN AGE6574  = 1;
ELSE IF TEMPAGE = 8 THEN AGE75UP  = 1;
end;
*---------------------------------------------------;
* Create the FEMALE dummy variable;
*---------------------------------------------------;
IF      FORM = 1 THEN TEMPSEX = C99D73; DROP TEMPSEX;
ELSE IF FORM = 2 THEN TEMPSEX = C99E74;

IF TEMPSEX = 2 THEN
   FEMALE = 1;
ELSE
   FEMALE = 0;

*---------------------------------------------------;
* create the beneficiary group/enrollment group subsets;
*---------------------------------------------------;
GROUP1 = 0;
GROUP2 = 0;
GROUP3 = 0;
GROUP4 = 0;
GROUP5 = 0;
GROUP6 = 0;
GROUP7 = 0;
GROUP8 = 1;     * EVERYONE;

IF XINS_COV  = 1 THEN GROUP1 = 1;
IF XENR_PCM = 1   THEN GROUP2 = 1;
IF XENR_PCM = 2        THEN GROUP3 = 1;
IF XINS_COV IN (2,3) THEN GROUP4 = 1;
IF AGEUND6  = 1        THEN GROUP5 = 1;
IF AGE0612  = 1        THEN GROUP6 = 1;
IF AGE1317  = 1        THEN GROUP7 = 1;

*---------------------------------------------------;
* reorder the codes for C99C32;
* They are in reverse order;
*---------------------------------------------------;
IF C99C32 = 1      THEN R99032 = 4;
ELSE IF C99C32 = 2 THEN R99032 = 3;
ELSE IF C99C32 = 3 THEN R99032 = 2;
ELSE IF C99C32 = 4 THEN R99032 = 1;
ELSE IF C99C32 < 0 THEN R99032 = .;
   IF R99032 > 1   THEN R99032 = R99032 - 1;


*-------------------------------------------------------------;
* recode variables with Never, Sometimes, Usually and Always;
* recode Never & Sometimes (1 & 2) to 1.
* recode Usually (3) to 2.
* recode Always (4) to 3.
*-------------------------------------------------------------;
IF C99C18 = 1      THEN R99018 = 1;
ELSE IF C99C18 = 2 THEN R99018 = 1;
ELSE IF C99C18 = 3 THEN R99018 = 2;
ELSE IF C99C18 = 4 THEN R99018 = 3;
ELSE IF C99C18 < 0 THEN R99018 = .;

IF C99C20 = 1      THEN R99020 = 1;
ELSE IF C99C20 = 2 THEN R99020 = 1;
ELSE IF C99C20 = 3 THEN R99020 = 2;
ELSE IF C99C20 = 4 THEN R99020 = 3;
ELSE IF C99C20 < 0 THEN R99020 = .;

IF C99C23 = 1      THEN R99023 = 1;
ELSE IF C99C23 = 2 THEN R99023 = 1;
ELSE IF C99C23 = 3 THEN R99023 = 2;
ELSE IF C99C23 = 4 THEN R99023 = 3;
```

```
ELSE IF C99C23 < 0 THEN R99023 = .;

IF C99C35 = 1      THEN R99035 = 1;
ELSE IF C99C35 = 2 THEN R99035 = 1;
ELSE IF C99C35 = 3 THEN R99035 = 2;
ELSE IF C99C35 = 4 THEN R99035 = 3;
ELSE IF C99C35 < 0 THEN R99035 = .;

IF C99C36 = 1      THEN R99036 = 1;
ELSE IF C99C36 = 2 THEN R99036 = 1;
ELSE IF C99C36 = 3 THEN R99036 = 2;
ELSE IF C99C36 = 4 THEN R99036 = 3;
ELSE IF C99C36 < 0 THEN R99036 = .;

IF C99C37 = 1      THEN R99037 = 1;
ELSE IF C99C37 = 2 THEN R99037 = 1;
ELSE IF C99C37 = 3 THEN R99037 = 2;
ELSE IF C99C37 = 4 THEN R99037 = 3;
ELSE IF C99C37 < 0 THEN R99037 = .;

IF C99C39 = 1      THEN R99039 = 1;
ELSE IF C99C39 = 2 THEN R99039 = 1;
ELSE IF C99C39 = 3 THEN R99039 = 2;
ELSE IF C99C39 = 4 THEN R99039 = 3;
ELSE IF C99C39 < 0 THEN R99039 = .;

IF C99C40 = 1      THEN R99040 = 1;
ELSE IF C99C40 = 2 THEN R99040 = 1;
ELSE IF C99C40 = 3 THEN R99040 = 2;
ELSE IF C99C40 = 4 THEN R99040 = 3;
ELSE IF C99C40 < 0 THEN R99040 = .;

IF C99C33 = 1      THEN R99033 = 1;
ELSE IF C99C33 = 2 THEN R99033 = 1;
ELSE IF C99C33 = 3 THEN R99033 = 2;
ELSE IF C99C33 = 4 THEN R99033 = 3;
ELSE IF C99C33 < 0 THEN R99033 = .;

IF C99C34 = 1      THEN R99034 = 1;
ELSE IF C99C34 = 2 THEN R99034 = 1;
ELSE IF C99C34 = 3 THEN R99034 = 2;
ELSE IF C99C34 = 4 THEN R99034 = 3;
ELSE IF C99C34 < 0 THEN R99034 = .;

*-------------------------------------------------------------;
* Recode childs health status based on form
*-------------------------------------------------------------;
IF FORM = 1 THEN R99061 = C99D61;
ELSE R99061 = C99E61;
IF R99061 < 0 THEN R99061 = .;

*-------------------------------------------------------------;
* Recode B/S/N variables to one missing condition ".";
*-------------------------------------------------------------;
R99005  =  C99C05;  IF R99005  < 0 THEN R99005 = .;
R99013  =  C99C13;  IF R99013  < 0 THEN R99013 = .;
R99030  =  C99C30;  IF R99030  < 0 THEN R99030 = .;
R99031  =  C99C31;  IF R99031  < 0 THEN R99031 = .;
R99049  =  C99C49;  IF R99049  < 0 THEN R99049 = .;
R99051  =  C99C51;  IF R99051  < 0 THEN R99051 = .;
R99056  =  C99C56;  IF R99056  < 0 THEN R99056 = .;

*-------------------------------------------------------------;
* These variables are used for MPR composites and they are
* being included for INFORMATIONAL PURPOSES ONLY.
*-------------------------------------------------------------;
R99008  =  C99C08;  IF R99008  < 0 THEN R99008 = .;
R99009  =  C99C09;  IF R99009  < 0 THEN R99009 = .;
R99010  =  C99C10;  IF R99010  < 0 THEN R99010 = .;
R99060A = C99C60A;  IF R99060A < 0 THEN R99060A = .;
R99060B = C99C60B;  IF R99060B < 0 THEN R99060B = .;
R99060C = C99C60C;  IF R99060C < 0 THEN R99060C = .;
R99060D = C99C60D;  IF R99060D < 0 THEN R99060D = .;
```

```
   R99021  =  C99C21;  IF R99021  < 0 THEN R99021 = .;
   R99024  =  C99C24;  IF R99024  < 0 THEN R99024 = .;
   R99027  =  C99C27;  IF R99027  < 0 THEN R99027 = .;

   *-------------------------------------------------------------;
   * Recode the CAHPS rating variables.
   *-------------------------------------------------------------;
   R99041  =  C99C41;  IF R99041  < 0 THEN R99041 = .;
   R99007  =  C99C07;  IF R99007  < 0 THEN R99007 = .;
   R99057  =  C99C57;  IF R99057  < 0 THEN R99057 = .;

   *----------------------;
   * Create region dummies;
   *----------------------;
   IF SUPREGA NE . THEN DO;
      ARRAY REGDUMS (3) REG01 REG02 REG03;
      DO I = 1 TO DIM(REGDUMS);
         REGDUMS(I)=0;
      END;

      IF       SUPREGA = 1  THEN REG01 = 1;
      ELSE IF  SUPREGA = 2  THEN REG02 = 1;
      ELSE IF  SUPREGA = 3  THEN REG03 = 1;
   END;

* Sort the main file to reorder it by MPRID;
* Hopefully it will put the file into a;
* somewhat random order for printing;
 PROC SORT DATA=OUT.ENTIRE;
      BY MPRID;
 RUN;


* Print the contents of IN.ENTIRE;
 PROC CONTENTS DATA=OUT.ENTIRE;
      TITLE2 'Contents of ENTIRE';
 RUN;


* Print some of the recoded records;
PROC PRINT DATA=OUT.ENTIRE(obs=60);
     TITLE2 'Print of KEY variables';
     VAR
        MPRID
        ZAGE
        SUPREGA
        CACSMPL
        BFGROUPP
        SREDHIGH
        ENGROUPP
        STRATUM
        XENR_PCM
        XINS_COV
        WRWT99
       ;
RUN;


* Print some of the recoded records;
PROC PRINT DATA=OUT.ENTIRE(obs=60);
     TITLE2 'Print of AGE and SEX dummies';
     VAR
        ZAGE
        AGEUND6
        AGE0612
        AGE1317
        AGEUND18
        AGE1824
        AGE2534
        AGE3544
        AGE4554
        AGE5564
        AGE6574
```

```
        AGE75UP
        FEMALE
        GROUP1
        GROUP2
        GROUP3
        GROUP4
        GROUP5
        GROUP6
        GROUP7
        XENR_PCM
        XINS_COV
        ENGROUPP
        BFGROUPP
       ;
RUN;

PROC PRINT DATA=OUT.ENTIRE(OBS=60);
     TITLE2 'Print of recoded question variables';
     VAR
        C99C05  R99005
        C99C13  R99013
        C99C30  R99030
        C99C31  R99031
        C99C18  R99018
        C99C20  R99020
        C99C23  R99023
        C99C32  R99032
        C99C35  R99035
        C99C36  R99036
        C99C37  R99037
        C99C39  R99039
        C99C40  R99040
        C99C33  R99033
        C99C34  R99034
        C99C49  R99049
        C99C51  R99051
        C99C56  R99056
        C99C41  R99041
        C99C07  R99007
        C99C57  R99057
       ;
RUN;

PROC PRINT DATA=OUT.ENTIRE(OBS=60);
     TITLE2 'Print of recoded question variables';
     VAR
        FORM SREDHIGH
        C99D72
        C99D73
        C99E73
        C99E74
        C99D74A C99D74B C99D74C C99D74D C99D74E C99D74F
        C99E75A C99E75B C99E75C C99E75D C99E75E C99E75F
        C99D61  C99E61  R99061
        ;
RUN;


PROC PRINT DATA=OUT.ENTIRE(OBS=60);
      TITLE2 'Print of recoded REGION variables';
      VAR
        XREGION
        REG01
        REG02
        REG03
        ;
RUN;


* Create the 7 subgroups for processing by STEP2.SAS;
DATA OUT.GROUP1
      OUT.GROUP2
      OUT.GROUP3
```

```
      OUT.GROUP4
      OUT.GROUP5
      OUT.GROUP6
      OUT.GROUP7;

      SET OUT.ENTIRE;

      DROP
        C99C05
        C99C13
        C99C30
        C99C31
        C99C18
        C99C20
        C99C23
        C99C32
        C99C35
        C99C36
        C99C37
        C99C39
        C99C40
        C99C33
        C99C34
        C99C49
        C99C51
        C99C56
        C99C08
        C99C09
        C99C10
        C99C60A
        C99C60B
        C99C60C
        C99C60D
        C99C21
        C99C24
        C99C27
        C99C41
        C99C07
        C99C57
        ;
      IF GROUP1 = 1 THEN OUTPUT OUT.GROUP1;
      IF GROUP2 = 1 THEN OUTPUT OUT.GROUP2;
      IF GROUP3 = 1 THEN OUTPUT OUT.GROUP3;
      IF GROUP4 = 1 THEN OUTPUT OUT.GROUP4;
      IF GROUP5 = 1 THEN OUTPUT OUT.GROUP5;
      IF GROUP6 = 1 THEN OUTPUT OUT.GROUP6;
      IF GROUP7 = 1 THEN OUTPUT OUT.GROUP7;
RUN;
```

2. CALCULATE CAHPS ADJUSTED SCORES

```
/*****************************************************************
/*  Project:  DoD - 1999 Child Report Cards
/*  Program:  STEP2C.SAS
/*  Purpose:  Draft Child Report Card
/*  Requires program STEP1C.SAS to have been run
/*  Programming specifications for Child report card
/*  The Child report card contains a large number of
/*  risk-adjusted scores.  Some scores are
/*  calculated from responses to individual survey questions.
/*  Composite scores are calculated by
/*  combining scores from individual questions.
/*  The scores then are compared with external civilian
/*  benchmarks.  The programming tasks involved in building
/*  the report card are:
/*       1)   preparing data for analyses
/*       2)   estimating risk adjustment models
/*       3)   calculating risk-adjusted values and variances
/*       4)   calculating benchmarks
/*       5)   comparing risk-adjusted values to benchmarks
/*            and hypothesis testing
/*
/* SUBGROUPS
/* ____________________________________________________________________
/*    Seven subgroups          Definitions
/* ____________________________________________________________________
/* 1. Prime enrollees          XINS_COV IN(1,2,6)
/* 2. Enrollees w/mil PCM      XENR_PCM IN(2,6)
/* 3. Enrollees w/civ PCM      XENR_PCM = 3
/* 4. Nonenrollees             XINS_COV IN(3,4,5)
/* 5. Active duty              AGEUND6=1
/* 6. Active duty dependents   AGE0612=1
/* 7. Retirees and dependents  AGE1317=1
/*
/*  PREV PGM:  STEP1C.SAS
/*  NEXT PGM:  ALLSCORE.SAS
/*****************************************************************/
OPTIONS NOCENTER LS=132 PS=78 SOURCE NOOVP COMPRESS=YES;
*OPTIONS NOCENTER LS=132 PS=78 SOURCE NOOVP MPRINT MLOGIC SYMBOLGEN STIMER;
LIBNAME   IN1      "DATA";
LIBNAME   IN2      "J:\8687\DATA1999\CFINAL";
LIBNAME   OUT      "DATA";
LIBNAME   OUT2     "DATA\CHILDHATFILES";

*--------------------------------------;
*-       set the parameters here      -;
*--------------------------------------;

* set the number of Dependent variables to process;
* One does not need to start at 1, but the max must be >= min;
%LET MIN_VAR =  1;
%LET MAX_VAR = 21;

* set the number of subgroups to process;
%LET MIN_GRP = 1;
%LET MAX_GRP = 7;

* I expect these to remain the same for;
* a particular dependent variable run;
%LET WGT      = WRWT99;
%LET IND_VAR1 = R99061;
%LET IND_VAR2 =   ;  *FEMALE;
%LET IND_VAR3 =   ;  *SREDHIGH;
%LET DEBUGFLG = 0;   *Set to 1 if you want extra printout;


%LET TITL1 = Prime enrollees;
%LET TITL2 = Enrollees w/military PCM;
%LET TITL3 = Enrollees w/civilian PCM;
%LET TITL4 = Nonenrollees;
%LET TITL5 = Under Age 6;
%LET TITL6 = Age 6-12;
%LET TITL7 = Age 13-17;
```

```
%LET TITL8 = All major groups;


%* GETTING NEEDED CARE;
%LET DEPVAR1 = R99005;
%LET DEPVAR2 = R99013;
%LET DEPVAR3 = R99030;
%LET DEPVAR4 = R99031;

%* GETTING NEEDED CARE QUICKLY;
%LET DEPVAR5 = R99018;
%LET DEPVAR6 = R99020;
%LET DEPVAR7 = R99023;
%LET DEPVAR8 = R99032;

%* HOW WELL DOCTORS COMMUNICATE;
%LET DEPVAR9  = R99035;
%LET DEPVAR10 = R99036;
%LET DEPVAR11 = R99037;
%LET DEPVAR12 = R99039;
%LET DEPVAR13 = R99040;

%* COURTEOUS AND HELPFUL OFFICE STAFF;
%LET DEPVAR14 = R99033;
%LET DEPVAR15 = R99034;

%* CUSTOMER SERVICE;
%LET DEPVAR16 = R99049;
%LET DEPVAR17 = R99051;
%LET DEPVAR18 = R99056;

%* RATING ALL HEALTH CARE: 0 - 10;
%LET DEPVAR19 = R99041;

%* RATING OF HEALTH PLAN: 0 - 10;
%LET DEPVAR20 = R99057;

%* RATING OF PERSONAL DR: 0 - 10;
%LET DEPVAR21 = R99007;

%MACRO SCORE;
**************************************;
* use this macro for all groups;
* super region variables are to be used     ;
*************************************;
%PUT ****************************************;
%PUT STARTING MACRO SCORE;
%PUT "GROUP    = " GROUP&IGRP;
%PUT "TITLE    = " &&DEPVAR&IVAR  &&TITL&IGRP;
%PUT "DEP_VAR  = " &&DEPVAR&IVAR;
%PUT "IND_VAR1 = " &IND_VAR1;
%PUT "IND_VAR2 = " &IND_VAR2;
%PUT "IND_VAR3 = " &IND_VAR3;
%PUT "WGT      = " &WGT;
%PUT ****************************************;

*---------------------------------------------------;
* If the current group is 1 use the skeleton files;
* else used the previous groups output file;
* The mrgfile is added to by each subgroup;
*---------------------------------------------------;
%LET RMRGFILE = OUT.R_&&DEPVAR&IVAR;
%IF "&IGRP" = "1" %THEN %LET RMRGFILE = IN2.SKELREG;

* run regression using the region level variables;
* output a BETA file (1 record) and the subgroup;
* file with residuals attached (many records);
PROC REG DATA = GROUP&IGRP OUTEST=BETAS;
     TITLE2 "Regression Model for GROUP&igrp for regions";
     TITLE3 "Beneficiary group&igrp:  &&TITL&IGRP";
     WEIGHT &WGT;
     %INCLUDE 'REGRSREG.INC';
     OUTPUT OUT = OUT2.H&IGRP&&DEPVAR&IVAR(KEEP=MPRID &WGT STRATUM
```

```
                        PRED&IGRP RESID&IGRP SUPREGA &&DEPVAR&IVAR)
              P = PRED&IGRP
              R = RESID&IGRP;

RUN;

* print of HCSDB file with the residuals and predicted values;
%IF &DEBUGFLG > 0  %THEN %DO;
    PROC PRINT DATA=OUT2.H&IGRP&&DEPVAR&IVAR (OBS=70);
         TITLE2 "OUT2.H&IGRP&&DEPVAR&IVAR:  file with predicted values and the RESID&IGRP";
         TITLE3 "Beneficiary group&igrp:  &&TITL&IGRP";
         VAR MPRID SUPREGA &&DEPVAR&IVAR RESID&IGRP PRED&IGRP;
    RUN;

    PROC PRINT DATA=BETAS;
         TITLE2 "BETAS:  file with coefficients";
         TITLE3 "Beneficiary group&igrp:  &&TITL&IGRP";
    RUN;
%END;


*-----------------------------------------;
*----- get the standard err/variance ----;
*-----------------------------------------;
%LET DEP = &&DEPVAR&IVAR;
%R_SUDAAN(OUT2.H&IGRP&&DEPVAR&IVAR);



* calculate prelim adjusted scores for the risk-adjusters;
* merge adjuster means with the adjuster coefficients;
* then sum their products. Finally add in the intercept;
 DATA ADJUST;
    SET MEANFILE;
    IF _N_ = 1 THEN SET BETAS(DROP = _TYPE_);
    %INCLUDE 'RISKARRY.INC';
    %INCLUDE 'RISKMEAN.INC';
    DO I = 1 TO DIM(COEFFS);
       IF COEFFS(I) = . THEN COEFFS(I) = 0;
       IF MEANS(I) = .  THEN MEANS(I)  = 0;
       ADJUST + ( COEFFS(I) * MEANS(I) );
    END;
    ADJUST = ADJUST + INTERCEP;
 RUN;


* add the region coefficients to the adjusted value from above;
* output one record per region with the region;
* level adjusted scores;
 DATA COEFFREG(KEEP=SUPREGA NEWADJST);
    SET ADJUST;
    %INCLUDE 'REGARRAY.INC';
    LENGTH NAME $8;
    DO I=1 TO DIM(REGRHS);
       CALL VNAME(REGRHS(I),NAME);
       SUPREGA=INPUT(SUBSTR(NAME,4,2),2.);
       IF REGRHS(I) = . THEN REGRHS(I) = 0;
       NEWADJST=ADJUST + REGRHS(I);
       OUTPUT;
    END;
RUN;


* sum of wgts for each region;
PROC MEANS DATA=GROUP&IGRP NWAY NOPRINT ;
  CLASS SUPREGA;
  VAR   &WGT;
  OUTPUT OUT=REG_WGTS (DROP = _TYPE_ _FREQ_) N=REGCNT&IGRP SUM=REGWGT&IGRP;
RUN;


* merge the COEFFREG file with the region;
* adjusted scores to the region level total weight;
```

```
* merge by the region.  Creates a region level;
* file with the total sample weight of the region;
DATA COEFFREG;
      MERGE COEFFREG(IN=IN1)
            REG_WGTS(IN=IN2   KEEP=SUPREGA REGCNT&IGRP REGWGT&IGRP);
      BY SUPREGA;
      IF IN1;
RUN;


%IF &DEBUGFLG > 0  %THEN %DO;
    PROC PRINT DATA=MEANFILE;
         TITLE2 'Print of MEANFILE';
         TITLE3 "Beneficiary group&igrp:  &&TITL&IGRP";
    RUN;

    PROC PRINT DATA=ADJUST;
         TITLE2 'Print of ADJUST';
         TITLE3 "Beneficiary group&igrp:  &&TITL&IGRP";
    RUN;

    PROC PRINT DATA=COEFFREG;
         TITLE2 'Print of COEFFREG: Region Adjusted Scores';
         TITLE3 "Beneficiary group&igrp:  &&TITL&IGRP";
    RUN;

    PROC PRINT DATA=REG_WGTS;
         TITLE2 'Print of REG_WGTS: Region Area Sum of WGTS';
         TITLE3 "Beneficiary group&igrp:  &&TITL&IGRP";
    RUN;

    PROC PRINT DATA=COEFFREG;
         TITLE2 'Print of COEFFREG: Regions Adjusted Scores - with sum of wgts and region';
         TITLE3 "Beneficiary group&igrp:  &&TITL&IGRP";
    RUN;
%END;


* Calculate region level adjusted scores from the;
* region level adjusted scores in COEFFREG;
PROC MEANS DATA=COEFFREG NWAY NOPRINT;
  WEIGHT REGWGT&IGRP;
  CLASS  SUPREGA;
  VAR    NEWADJST;
  OUTPUT OUT=REGFILE1 (DROP = _TYPE_ _FREQ_) MEAN=ADJ&IGRP;
RUN;


%IF &DEBUGFLG > 0  %THEN %DO;
   PROC PRINT DATA=REGFILE1;
        TITLE2 'Print of REGFILE1: Region Scores';
        TITLE3 "Beneficiary group&igrp:  &&TITL&IGRP";
   RUN;
%END;

* merge the previous groups region results (if any);
* with the region level std errs and the region;
* level results from catchment results collapsed to region;
DATA OUT.R_&&DEPVAR&IVAR;
     MERGE &RMRGFILE(IN=INS)
           R&IGRP&&DEPVAR&IVAR
           REG_WGTS(KEEP = REGCNT&IGRP REGWGT&IGRP SUPREGA)
           REGFILE1(KEEP = ADJ&IGRP SUPREGA);
     BY SUPREGA;
     DEPENDNT = "&&DEPVAR&IVAR";
     IF INS;
RUN;

* merge the previous groups regional results (if any);
* with the region level std err and the region;
* level results from the current group/dependent var;
DATA OUT.R_&&DEPVAR&IVAR;
    MERGE OUT.R_&&DEPVAR&IVAR(IN=INS)
```

```
          R&IGRP&&DEPVAR&IVAR    /*KRR - removed perm dataset ref to OUT2 */
          REG_WGTS
          REGFILE1;
    BY SUPREGA;
    DEPENDNT = "&&DEPVAR&IVAR";
    IF INS;
RUN;

PROC PRINT DATA=OUT.R_&&DEPVAR&IVAR;
     TITLE2 "Print of SUPER REGION variables in &&DEPVAR&IVAR";
     TITLE3 "Beneficiary group&igrp:  &&TITL&IGRP";
RUN;

%MEND SCORE;

%MACRO MAKE_INC;
****************************************;
* creates include files for later Procs;
* Needs to be run each time.  Called   ;
* in the outer (beneficiary loop).     ;
* I chose this method because it was   ;
* clearer(to me at least).             ;
* This macro needs to be run once per  ;
* Dep var per subgroup.                ;
****************************************;

* Drop records where the dependent var is missing;
* Drop records with missing catchment or region values;
  DATA GROUP&IGRP;
       SET IN1.GROUP&IGRP;
       IF &&DEPVAR&IVAR NOT = .;
  RUN;

 DATA _NULL_;
      SET GROUP&IGRP END = EOF;
      IF &&DEPVAR&IVAR NOT = .;

      ARRAY AGECNT(8) 8 aCNT1 - aCNT8;
      RETAIN AGECNT 0;
      RETAIN CNT 0;
      ARRAY AGENAM(8)  $8 AGENAM1 - AGENAM8;
      ARRAY AGENAMX(8) $8 AGENAMX1 - AGENAMX8;
      RETAIN AGENAM;
      RETAIN AGENAMX;
      ARRAY REGCNT(3)   8 REGCNT01 - REGCNT3;
      RETAIN CATCNT 0;
      RETAIN REGCNT 0;

      * create a name array for the parent age dummies;
      IF _N_ = 1 THEN DO;
         AGENAM(1) = "AGEUND18";
         AGENAM(2) = "AGE1824";
         AGENAM(3) = "AGE2534";
         AGENAM(4) = "AGE3544";
         AGENAM(5) = "AGE4554";
         AGENAM(6) = "AGE5564";
         AGENAM(7) = "AGE6574";
         AGENAM(8) = "AGE75UP";
      END;

      * total record count;
      CNT + 1;

      * count records in each age group;
      * we will use only age groups with more;
      * than 2 obs;
      IF AGEUND18 = 1 THEN AGECNT(1) + 1;
      IF AGE1824  = 1 THEN AGECNT(2) + 1;
      IF AGE2534  = 1 THEN AGECNT(3) + 1;
      IF AGE3544  = 1 THEN AGECNT(4) + 1;
      IF AGE4554  = 1 THEN AGECNT(5) + 1;
      IF AGE5564  = 1 THEN AGECNT(6) + 1;
      IF AGE6574  = 1 THEN AGECNT(7) + 1;
```

```
      IF AGE75UP  = 1 THEN AGECNT(8) + 1;

      * count records in each SUPREGA group;
      * we will only use SUPER REGIONS ;
      * with more than than 2 obs;
      * I am using the region value as the subscript;
      * to make the code simpler and more readable;
      IF SUPREGA >= 1 AND SUPREGA <= 3  THEN DO;
         REGCNT(SUPREGA) = REGCNT(SUPREGA) + 1;
      END;

      IF EOF THEN GOTO ENDFILE;
      RETURN;


ENDFILE:
     * create a title common to all procs in the current group;
     TITLE  " &&DEPVAR&IVAR  &&TITL&IGRP";

     * display counts in the log;
     %IF &DEBUGFLG > 0  %THEN %DO;
        PUT ' ';
        PUT 'AT EOF:';
        PUT "TOTAL CNT = "    CNT;
        PUT AGENAM(1) "  " AGECNT(1)=;
        PUT AGENAM(2) "  " AGECNT(2)=;
        PUT AGENAM(3) "  " AGECNT(3)=;
        PUT AGENAM(4) "  " AGECNT(4)=;
        PUT AGENAM(5) "  " AGECNT(5)=;
        PUT AGENAM(6) "  " AGECNT(6)=;
        PUT AGENAM(7) "  " AGECNT(7)=;
        PUT AGENAM(8) "  " AGECNT(8)=;
        PUT " ";

        DO I = 1 TO 3;
           IF(REGCNT(I) > 0) THEN DO;
              PUT  'REG' I Z2. REGCNT(I) 6.;
           END;
        END;
        PUT ' ';


      %END;    *** of debug test;


     *-------------------------------------------------------------;
     * This include is for the regression using regions;
     * in this case we drop the last REGION;
     FILE 'REGRSREG.INC';
     PUT @6  "MODEL  &&DEPVAR&IVAR = ";
     IF "&IND_VAR1" NE "" THEN PUT @12 "&IND_VAR1"; /* KRR - only output when present */
     IF "&IND_VAR2" NE "" THEN PUT @12 "&IND_VAR2"; /* KRR - only output when present */
     IF "&IND_VAR3" NE "" THEN PUT @12 "&IND_VAR3"; /* KRR - only output when present */

     CNT2 = 0;
     * setup an array of those age groups that have > 1 obs;
     DO I = 1 TO 8;
        IF AGECNT(I) > 1 THEN DO;
           CNT2 +1;
           AGENAMX(CNT2) = AGENAM(I);
        END;
     END;

     * now drop the last category to create;
     * an omitted category which is required;
     * to solve the regression properly;
     DO I = 1 TO CNT2-1;
        PUT @12 AGENAMX(I);
     END;

     * ditto for the catchment areas with > 0 obs;
     * in this case we drop the the first USABLE category;
     * this is not consistent with the catchment area code;
```

```
* but this is the method that Portia used;
FIRST = 0;
DO I = 1 TO 3;   * skip the 1st region with 1+ obs;
   IF REGCNT(I) > 0 THEN DO;
      IF FIRST = 1 THEN PUT @12 'REG' I Z2.;
      FIRST = 1;
   END;
END;
PUT @11 ';';


*-------------------------------------------------------;
* now create the complete var statement;
* for the Proc MEANS used to replace the;
* independent variables missing values;
* we assume the age groups will always be used;
* These are also called the RISK FACTORS;
FILE 'RISKVARS.INC';
PUT @10  "VAR";
DO I = 1 TO CNT2;
   PUT @12 AGENAMX(I);
END;

* not all the other dependent variables will be used;
* only write them out if they are not null;
CNT3 = 0;
IF "&IND_VAR1" NE "" THEN DO;
    CNT3 + 1;
    PUT @12 "&IND_VAR1";
END;

IF "&IND_VAR2" NE "" THEN DO;
    CNT3 + 1;
    PUT @12 "&IND_VAR2";
END;

IF "&IND_VAR3" NE "" THEN DO;
    CNT3 + 1;
    PUT @12 "&IND_VAR3";
END;
PUT @11 ';';


*-------------------------------------------------------;
* create an ARRAY statement of the desired risk factors;
* called adjusters in the specs and in the code;
FILE 'RISKARRY.INC';
PUT @10  "ARRAY COEFFS(*) $8";
DO I = 1 TO CNT2;
   PUT @12 AGENAMX(I);
END;

CNT3 = 0;
IF "&IND_VAR1" NE "" THEN DO;
    CNT3 + 1;
    PUT @12 "&IND_VAR1";
END;

IF "&IND_VAR2" NE "" THEN DO;
    CNT3 + 1;
    PUT @12 "&IND_VAR2";
END;

IF "&IND_VAR3" NE "" THEN DO;
    CNT3 + 1;
    PUT @12 "&IND_VAR3";
END;
PUT @11 ';';


*-------------------------------------------------------;
* create an ARRAY of mean names for the output;
* from a proc MEANS of the Risk Factors in RISKARRY;
```

```
     FILE 'RISKMEAN.INC';
     IND_CNT = CNT2 + CNT3;
     PUT @6 "ARRAY MEANS(*) $8";
     DO I = 1 TO IND_CNT;
        PUT @12 "MEAN" I Z2.;
     END;
     PUT @11 ';';

*   ----------------------------------------------------;
*   create the equivalent of the following statement;
*   OUTPUT OUT=MEANFILE(DROP = _TYPE_) MEAN=MEAN1-MEAN&MEAN_CNT;
     FILE 'MEANFILE.INC';
     PUT @6 "OUTPUT OUT=MEANFILE(DROP = _TYPE_) MEAN = ";
     DO I = 1 TO IND_CNT;
        PUT @12 "MEAN" I Z2.;
     END;
     PUT @11 ';';

     *-------------------------------------------------------;
     * create a super region area array;
     * with at least ONE obs;
     FILE 'REGARRAY.INC';
     PUT @10  "ARRAY REGRHS(*) $8";
     DO I = 1 TO 3;
        IF REGCNT(I) > 0 THEN DO; *** ems 7/12/00 changed "> 1" to "> 0";
           PUT  @16 'REG' I Z2.;
        END;
     END;
     PUT @11 ';';
 RUN;

 * Create the means of the adjuster variables;
 * They will be used to replace missing adjuster variables;
 * calculate weighted means;
 PROC MEANS DATA=GROUP&IGRP;
   WEIGHT &WGT;
   %INCLUDE 'RISKVARS.INC';
   %INCLUDE 'MEANFILE.INC';
 RUN;


%IF &DEBUGFLG > 0  %THEN %DO;
   PROC PRINT DATA=MEANFILE;
        TITLE2 "Print of MEANFILE for Risk Adjuster variables";
        TITLE3 "Beneficiary group&igrp:  &&TITL&IGRP";
   RUN;
%END;

DATA GROUP&IGRP;
     SET GROUP&IGRP;
     IF _N_ = 1 THEN SET MEANFILE;
     %INCLUDE 'RISKARRY.INC';
     %INCLUDE 'RISKMEAN.INC';
     DO I = 1 TO DIM(COEFFS);
        IF COEFFS(I) = . THEN DO;
           COEFFS(I) = MEANS(I);
        END;
     END;
RUN;

%MEND MAKE_INC;



%MACRO R_SUDAAN(INFILE);
*****************************************************;
* use this macro to create standard err (variances);
* FOR: REGIONS                                      ;
*****************************************************;
%PUT ****************************************;
%PUT STARTING MACRO R_SUDAAN (SUPER REGIONS);
%PUT ****************************************;
```

```
DATA &INFILE;
   SET &INFILE;
   IF SUPREGA > 0;
RUN;


* Sort data by STRATUM;
PROC SORT DATA=&INFILE;
   BY STRATUM;
RUN;

%IF &DEBUGFLG > 5  %THEN %DO;
   PROC PRINT DATA=&INFILE(OBS=5);
        TITLE2 'Print of the input file to SUDAAN (SUPER REGION)';
        TITLE3 "Beneficiary group&igrp:  &&TITL&IGRP";
   RUN;
%END;


* Calculate values for super regions;
PROC DESCRIPT DATA=&INFILE DESIGN=STRWR NOPRINT;
   WEIGHT &WGT;
   SETENV DECWIDTH=4;
   NEST STRATUM / missunit;
   VAR RESID&IGRP;
   TABLES SUPREGA;
   SUBGROUP SUPREGA;
   LEVELS 3;
   OUTPUT  SEMEAN
         / TABLECELL=DEFAULT
           FILENAME=RS&DEP;
   RUN;

   DATA R&IGRP&&DEPVAR&IVAR;
        SET RS&DEP;
        KEEP SUPREGA SEMEAN;
        IF SEMEAN NE .;
        RENAME SEMEAN = SEMEAN&IGRP;
   RUN;


   PROC PRINT DATA=R&IGRP&&DEPVAR&IVAR;
      TITLE2 "Print REGION DESCRIPT DATA=R&IGRP&&DEPVAR&IVAR";
      TITLE3 "Beneficiary group&igrp:  &&TITL&IGRP";
   RUN;

%MEND  R_SUDAAN;


%***********************************;
%*  call the macros;
%***********************************;

%MACRO MAINLOOP(MIN_VAR,MAX_VAR,MIN_GRP,MAX_GRP);
    %* loop over the set of dependent variables;
       %DO IVAR = &MIN_VAR %TO &MAX_VAR;
          %DO IGRP = &MIN_GRP %TO &MAX_GRP;
             %MAKE_INC;
             %SCORE;
          %END;
       %END;

%MEND;

%MAINLOOP(&MIN_VAR,&MAX_VAR,&MIN_GRP,&MAX_GRP);
_
```

```
/*****************************************************************
/*  Project:  DoD - 1999 Adult Report Cards
/*  Program:  ALLSCORE.SAS
/*  Purpose:  Draft Adult Report Card
/*  Requires program STEP1.SAS to have been run
/*  Requires program STEP2.SAS to have been run
/*  Programming specifications for adult report card
/*
/*  PREV PGM:  STEP1.SAS
/*  PREV PGM:  STEP2.SAS
/*  NEXT PGM:  COMPOSIT.SAS

/*MODIFIED: SAME FORMAT AS THE OTHER


/*****************************************************************/
OPTIONS NOCENTER LS=132 PS=78 SOURCE NOOVP COMPRESS=YES;
*OPTIONS NOCENTER LS=132 PS=78 SOURCE NOOVP MPRINT MLOGIC SYMBOLGEN STIMER;

LIBNAME   IN   "DATA";
LIBNAME   IN2  "DATA\CHILDHATFILES";

%LET MIN_VAR = 1;
%LET MAX_VAR = 21;

*set the number of subroups to process;

%LET MIN_GRP = 1;
%LET MAX_GRP = 7;

%LET WGT     = WRWT99;
%LET DEBUGFLG = 0;
%LET IND_VAR1 = R99061;
%LET IND_VAR2 = FEMALE;
%LET IND_VAR3 = SREDHIGH;

%* GETTING NEEDED CARE;
%LET DEPVAR1 = R99005;
%LET DEPVAR2 = R99013;
%LET DEPVAR3 = R99030;
%LET DEPVAR4 = R99031;

%* GETTING NEEDED CARE QUICKLY;
%LET DEPVAR5 = R99018;
%LET DEPVAR6 = R99020;
%LET DEPVAR7 = R99023;
%LET DEPVAR8 = R99032;

%* HOW WELL DOCTORS COMMUNICATE;
%LET DEPVAR9  = R99035;
%LET DEPVAR10 = R99036;
%LET DEPVAR11 = R99037;
%LET DEPVAR12 = R99039;
%LET DEPVAR13 = R99040;

%* COURTEOUS AND HELPFUL OFFICE STAFF;
%LET DEPVAR14 = R99033;
%LET DEPVAR15 = R99034;

%* CUSTOMER SERVICE;
%LET DEPVAR16 = R99049;
%LET DEPVAR17 = R99051;
%LET DEPVAR18 = R99056;

%* RATING ALL HEALTH CARE: 0 - 10;
%LET DEPVAR19 = R99041;

%* RATING OF HEALTH PLAN: 0 - 10;
%LET DEPVAR20 = R99057;

%* RATING OF PERSONAL DR: 0 - 10;
%LET DEPVAR21 = R99007;
```

```
%MACRO ALLSCORE;
*************************************;
* create an all beneficiares adjusted ;
* score for super region level  ;
* this is done by combining groups    ;
* 5, 6, & 7                           ;
*************************************;
%PUT STARTING MACRO ALLSCORE;


*---------------------------;
* merge groups 5, 6 & 7     ;
* 5. Age Under 6            ;
* 6. Age 6-12               ;
* 7. Age 13-17              ;
*                           ;
*---------------------------;


DATA REGRESS8;

    SET IN2.H5&&DEPVAR&IVAR (RENAME=(RESID5=RESID8 PRED5=PRED8))
        IN2.H6&&DEPVAR&IVAR (RENAME=(RESID6=RESID8 PRED6=PRED8))
        IN2.H7&&DEPVAR&IVAR (RENAME=(RESID7=RESID8 PRED7=PRED8))
       ;
RUN;


*-----------------------------------------;
* get the standard err/variance using the  ;
* residuals from the regressions.          ;
*-----------------------------------------;
%R_SUDAAN(REGRESS8);


* Read the final output files created in group7;
* Create a total beneficiary adjusted score;
* by calculating the weighted average score;
* of the age groupings ;
* merge the std errors for group8 with the;
* newly created adjusted scores and weights;
* for the 3 super regions;
DATA IN.R8&&DEPVAR&IVAR;
       MERGE IN.R_&&DEPVAR&IVAR(IN=IN1)
             RSEMEAN8;
       BY SUPREGA;
       IF IN1;

       REGCNT8  = SUM(REGCNT5,REGCNT6,REGCNT7);
       REGWGT8  = SUM(REGWGT5,REGWGT6,REGWGT7);
       TOTSCORE = SUM(REGWGT5 * ADJ5,REGWGT6 * ADJ6,REGWGT7 * ADJ7);
       ADJ8 = TOTSCORE / REGWGT8;
       IF ADJ8 = . THEN SEMEAN8 = .;
       DROP TOTSCORE;
RUN;



PROC PRINT DATA=IN.R8&&DEPVAR&IVAR;
     TITLE2 "Print of the final REGION level output of R8&&DEPVAR&IVAR";
RUN;

%MEND ALLSCORE;



%MACRO R_SUDAAN(INFILE);
%PUT ****************************************;
%PUT STARTING MACRO R_SUDAAN (SUPER REGIONS);
%PUT ****************************************;

DATA &INFILE;
     SET &INFILE;
```

```
     KEEP MPRID &WGT STRATUM RESID8 PRED8 SUPREGA;
RUN;


%IF &DEBUGFLG > 5  %THEN %DO;
   PROC PRINT DATA=&INFILE(OBS=60);
        TITLE2 "Print of the input file to SUDAAN (REGION)";
        TITLE3 "Age groups: 5, 6 & 7";
   RUN;
%END;


* Sort data by STRATUM;
PROC SORT DATA=&INFILE;
   BY STRATUM;
RUN;


* Calculate values for regions;
PROC DESCRIPT DATA=&INFILE DESIGN=STRWR NOPRINT;
   WEIGHT   &WGT;
   SETENV   DECWIDTH=4;
   NEST     STRATUM / missunit;
   VAR      RESID8;
   TABLES   SUPREGA;
   SUBGROUP SUPREGA;
   LEVELS   16;
   OUTPUT   SEMEAN
          / TABLECELL=DEFAULT
            FILENAME=RSEMEAN8;
   RUN;

   DATA RSEMEAN8;
        SET    RSEMEAN8;
        KEEP   SUPREGA SEMEAN;
        IF     SEMEAN  NE .;
        IF     SUPREGA NE . and SUPREGA NE 0;
        RENAME SEMEAN = SEMEAN8;
   RUN;


   PROC PRINT DATA = RSEMEAN8;
      TITLE2 "Print REGION level output of DESCRIPT (RSEMEAN8)";
      TITLE3 "Age groups: 5, 6 & 7";
   RUN;

%MEND  R_SUDAAN;

%MACRO MAINLOOP(MIN_VAR,MAX_VAR);
    %* loop over the set of dependent variables;
       %DO IVAR = &MIN_VAR %TO &MAX_VAR;
           %ALLSCORE;
       %END;
%MEND MAINLOOP;

%MAINLOOP(&MIN_VAR,&MAX_VAR);
_
```

```
/*****************************************************************
/*  Project:  DoD - 1999 Child Report Cards
/*  Program:  COMPOSIT.SAS
/*  Purpose:  Draft Child Report Card
/*  Requires programs STEP1C, STEP2C.SAS, ALLSCORE.SAS to have been run
/*
/*  Modified: Keith Rathbun, 07/18/2000: Updated for child survey.
/*            Added processing for 5th dependent variable.  Update
/*            macro calls.
/*
/*****************************************************************/
OPTIONS NOCENTER LS=132 PS=78 SOURCE SOURCE2 MLOGIC MPRINT NOOVP COMPRESS=YES;
libname in  'j:\8687\programs\reportcards\cahps_child99\data';
libname in2 'j:\8687\programs\reportcards\cahps_child99\data\childhatfiles';
libname out 'j:\8687\programs\reportcards\cahps_child99\data';

%MACRO COMPOSIT (TYPE=,COMPOS=,VAR1=,VAR2=,VAR3=,VAR4=,VAR5=,QCOUNT=);

  DATA _NULL_;
  %IF "&TYPE" = "R" %THEN %DO;
      CALL SYMPUT ('BYVAR','SUPREGA');
  %END; %ELSE

 ****************************************;
 *   Create a Composite Score           ;
 ****************************************;
 DATA _NULL_;
     FILE 'FILES.INC';
     PUT @6 'SET';
     IF "&VAR1" NE '' THEN PUT @8 "IN.&TYPE.8&VAR1";
     IF "&VAR2" NE '' THEN PUT @8 "IN.&TYPE.8&VAR2";
     IF "&VAR3" NE '' THEN PUT @8 "IN.&TYPE.8&VAR3";
     IF "&VAR4" NE '' THEN PUT @8 "IN.&TYPE.8&VAR4";
     IF "&VAR5" NE '' THEN PUT @8 "IN.&TYPE.8&VAR5";
     PUT @8 ';';
 RUN;


 DATA COMPOS&COMPOS;
      LENGTH DEPENDNT $ 8;
      %INCLUDE 'FILES.INC';
      DEPENDNT = "&TYPE.COMPOS&COMPOS";
 RUN;

 PROC SORT DATA=COMPOS&COMPOS;
      BY &BYVAR;
 RUN;

 PROC PRINT DATA=COMPOS&COMPOS(OBS=60);
      TITLE "Print of COMPOS&COMPOS after sort";
 RUN;

 DATA COMPOS&COMPOS;
      SET COMPOS&COMPOS;
      BY &BYVAR;
        %IF "&TYPE" = "R" %THEN %DO;
       ARRAY N(*) REGCNT1 - REGCNT8;
       ARRAY W(*) REGWGT1 - REGWGT8;
       ARRAY TN(*) TOTCNT1 - TOTCNT8;
       ARRAY TW(*) TOTWGT1 - TOTWGT8;
   %END; %ELSE
   %IF "&TYPE" = "C" %THEN %DO;
       ARRAY N(*) CATCNT1 - CATCNT8;
       ARRAY W(*) CATWGT1 - CATWGT8;
       ARRAY TN(*) TOTCNT1 - TOTCNT8;
       ARRAY TW(*) TOTWGT1 - TOTWGT8;
   %END;
      ARRAY ADJ(*)    ADJ1 - ADJ8;
      ARRAY TOTADJ(*) TOTADJ1 - TOTADJ8;
      ARRAY AVGADJ(*) AVJADJ1 - AVJADJ8;
      RETAIN TOTADJ TN TW;
      RETAIN AVGADJ;
```

```
      IF FIRST.&BYVAR THEN DO;
         DO I = 1 TO DIM(TOTADJ);
            TOTADJ(I) = 0; TN(I)=0; TW(I)=0;
         END;
      END; DROP I;

      PUT ' ';
      PUT ' --- STARTING LOOP1: ' &BYVAR=;
      DO I = 1 TO DIM(TOTADJ);
         PUT I= ADJ(I)=;
         IF ADJ(I) NE . THEN DO;
            TOTADJ(I) = TOTADJ(I) + ADJ(I);
            TN(I)=TN(I)+N(I);
            TW(I)=TW(I)+W(I);
         END;
         PUT I= ADJ(I)= TOTADJ(I)=;
      END;


      PUT ' ';
      PUT ' --- STARTING LOOP2: ' &BYVAR=;
      IF LAST.&BYVAR THEN DO;
         DO I = 1 TO DIM(TOTADJ);
            PUT I= ADJ(I)= TOTADJ(I)= AVGADJ(I)=;
            AVGADJ(I) = TOTADJ(I)/&QCOUNT;
            adj(i)=avgadj(i);
            N(I)=TN(I)/&QCOUNT;
            W(I)=TW(I)/&QCOUNT;
         END;
         OUTPUT;
      END;

 RUN;



%do i=1 %to 8;
/* Collect Standard Errors and residuals from variables in composite */
/*%if &i=1|&i=2|&i=5|&i=8 %then %do;
%let byvar2=cacsmpl;
%let type2=c;
%end;
%else %do;
%let byvar2=xregion;
%let type2=r;
%end;*/
%if &type=R|(&i=1|&i=2|&i>4) %then %do;
%if &var1~= %then %do;
%let n=r_&var1;
%let m=s_&var1;

data s_&var1(rename=(semean&i=s_&var1));
set in.&type.8&var1(keep=semean&i &byvar);
proc sort; by &byvar;
data r_&var1;
%if &i~=8 %then %do;
set in2.h&i.&var1(rename=(resid&i=r_&var1));
%end;
%else %do;
set           in2.h5&var1(rename=(resid5=r_&var1))           in2.h6&var1(rename=(resid6=r_&var1))
in2.h7&var1(rename=(resid7=r_&var1));
%end;
proc sort data=r_&var1; by mprid;
%end;

%if &var2~= %then %do;
%let n=%str(&n r_&var2);
%let m=%str(&m s_&var2);
data s_&var2(rename=(semean&i=s_&var2));
set in.&type.8&var2(keep=semean&i &byvar);
proc sort; by &byvar;
data r_&var2;
%if &i~=8 %then %do;
```

```
set in2.h&i.&var2(rename=(resid&i=r_&var2));
%end;
%else %do;
set           in2.h5&var2(rename=(resid5=r_&var2))           in2.h6&var2(rename=(resid6=r_&var2))
in2.h7&var2(rename=(resid7=r_&var2));
%end;
proc sort data=r_&var2; by mprid;
%end;

%if &var3~= %then %do;
%let n=%str(&n r_&var3);
data s_&var3(rename=(semean&i=s_&var3));
set in.&type.8&var3(keep=semean&i &byvar);
proc sort; by &byvar;
data r_&var3;
%if &i~=8 %then %do;
set in2.h&i.&var3(rename=(resid&i=r_&var3));
%end;
%else %do;
set           in2.h5&var3(rename=(resid5=r_&var3))           in2.h6&var3(rename=(resid6=r_&var3))
in2.h7&var3(rename=(resid7=r_&var3));
%end;
proc sort data=r_&var3; by mprid;
%let m=%str(&m s_&var3); %end;

%if &var4~= %then %do;
%let n=%str(&n r_&var4);
data s_&var4(rename=(semean&i=s_&var4));
set in.&type.8&var4(keep=semean&i &byvar);
proc sort; by &byvar;
data r_&var4;
%if &i~=8 %then %do;
set in2.h&i.&var4(rename=(resid&i=r_&var4));
%end;
%else %do;
set           in2.h5&var4(rename=(resid5=r_&var4))           in2.h6&var4(rename=(resid6=r_&var4))
in2.h7&var4(rename=(resid7=r_&var4));
%end;
%let m=%str(&m s_&var4);
proc sort data=r_&var4; by mprid;
%end;

%if &var5~= %then %do;
%let n=%str(&n r_&var5);
data s_&var5(rename=(semean&i=s_&var5));
set in.&type.8&var5(keep=semean&i &byvar);
proc sort; by &byvar;
data r_&var5;
%if &i~=8 %then %do;
set in2.h&i.&var5(rename=(resid&i=r_&var5));
%end;
%else %do;
set           in2.h5&var5(rename=(resid5=r_&var5))           in2.h6&var5(rename=(resid6=r_&var5))
in2.h7&var5(rename=(resid7=r_&var5));
%end;
%let m=%str(&m s_&var5);
proc sort data=r_&var5; by mprid;
%end;

/* Merge residual files and estimate correlations */
data infile;
merge &n; by mprid;
proc sort; by &byvar;
proc corr  outp=outf noprint;
by &byvar;
var &n;
weight wrwt99;
data outf;
set outf; by &byvar;
where _type_='CORR';
/*  sum standard error of a row variable times correlation times standard error of each column
variable, then sum sums and take square root, divide by number of variables */
data final;
```

```
merge &m outf; by &byvar;
data final;
set final; by &byvar;
array r_val &n;
array s_val &m;
sde=0;
do i=1 to dim(s_val);
%do j=1 %to &qcount;
if _name_="R_&&var&j" then
sde=sum(sde,r_val(i)*s_&&var&j*s_val(i));
%end;
end;
data sefin&compos._&i;
set final; by &byvar;
if first.&byvar then tv=0;
 tv+sde;
if last.&byvar then do;
sde&i=(tv**.5)/&qcount;
output;
end;
%if &i=1 %then %do;
data sefin&compos;
set sefin&compos._1(keep=&byvar sde&i); by &byvar;
rename sde&i=semean&i;
run;
%end;
%else %do;
data sefin&compos;
merge sefin&compos sefin&compos._&i(keep=&byvar sde&i); by &byvar;
rename sde&i=semean&i;
run;
%end;


%end;
%end;

data out.&type.compos&compos;
merge compos&compos sefin&compos; by &byvar;
run;
PROC PRINT DATA=OUT.&TYPE.COMPOS&COMPOS;
      TITLE1 COMPTITL;
 RUN;
%MEND COMPOSIT;

*--------------------------------------;
*-       set the parameters here      -;
*--------------------------------------;
************************************;
*  call the macro for each composite;
************************************;
%COMPOSIT (type=R,compos=1,var1=R99005,var2=R99013,var3=R99030,var4=R99031,qcount=4);
%COMPOSIT (type=R,compos=2,var1=R99018,var2=R99020,var3=R99023,var4=R99032,qcount=4);
%COMPOSIT (type=R,compos=3,var1=R99035,var2=R99036,var3=R99037,var4=R99039,var5=R99040,qcount=5);
%COMPOSIT (type=R,compos=4,var1=R99033,var2=R99034,qcount=2);
%COMPOSIT (type=R,compos=5,var1=R99049,var2=R99051,var3=R99056,qcount=3);
```

3. CONVERT CAHPS SCORES INTO WEB LAYOUT

```
*****************************************************************************
*
* PROGRAM:  LOADCAHC.SAS
* TASK:     1999 DOD HEALTH CARE SURVEY ANALYSIS (8687-200)
* PURPOSE:  Convert the CAHPS Scores Database into the WEB layout
*
* WRITTEN:  07/14/2000 BY KEITH RATHBUN
*
* INPUTS:   1) CAHPS Individual and Composite data sets with adjusted scores
*
* OUTPUT:   1) LOADCAHC.SD2 - Combined CAHPS Scores Database in WEB layout
*
* INCLUDES: 1) LOADCAHC.INC - Format definitions for CAHPS Individual
*              and composite data sets
*
* NOTES:
*
* 1) The following steps need to be run prior to this program:
*    - STEP1C.SAS - Recode questions and generate group files
*    - STEP2C.SAS - Calculate individual adjusted scores for group 1-7
*    - ALLSCORE.SAS - Calculate individual adjusted scores overall (group 8)
*    - COMPOSIT.SAS - Calculate composite adjusted scores for group 1-8
*
* 2) The output file (LOADCAHC.SD2) will be run through the
*    MAKEHTMC.SAS program to generate the WEB pages.
*
* 3) This program is a modified version of LOADCAHP.SAS adapted to meet
*    the requirements of the child report card.
*
*****************************************************************************
* Assign data libraries and options
*****************************************************************************;
LIBNAME IN      "J:\8687\PROGRAMS\REPORTCARDS\CAHPS_CHILD99\DATA";
LIBNAME OUT     "J:\8687\PROGRAMS\LOADWEB\CAHPS_CHILD99\DATA";
LIBNAME LIBRARY "J:\8687\DATA1999\CFINAL\FMTLIB";
OPTIONS PS=79 LS=132 COMPRESS=YES NOCENTER;

*****************************************************************************
* Load Format definitions for CAHPS Individual and composite data sets.
*****************************************************************************;
%INCLUDE "J:\8687\PROGRAMS\LOADWEB\LOADCAHC.INC";

*****************************************************************************
*****************************************************************************
*
* Process Macro Input Parameters:
*
* 1) QUESTION = Variable Question Name (DSN).
*    - For individual Questions it is the variable name
*    - For composite Questions it is called xCOMPOSn
*      where n = a predefined composite # and
*            x = R (Region) or C (Catchment)
* 2) TYPE = Type of Score (COMPOSITE or INDIVIDUAL)
* 3) REGCAT = Region/Catchment Area
*
*****************************************************************************
*****************************************************************************;
%MACRO PROCESS(QUESTION=,TYPE=,REGCAT=);
*****************************************************************************
* Assign value for BENTYPE composite year
*****************************************************************************;
%LET YEAR = "1999";

*****************************************************************************
* Assign prefix for weighted/unweighted count variables.
* Unweighted counts are REGCNTn or CATCNTn where n=group number.
* Weighted counts are REGWGTn or CATWGTn where n=group number.
*****************************************************************************;
%IF "&REGCAT" = "Region" %THEN %DO;
   %LET PREFIX = REG;
%END;
%ELSE %IF "&REGCAT" = "Catchment" %THEN %DO;
```

```
   %LET PREFIX = CAT;
%END;
%ELSE %DO;
   %PUT "ERROR: Invalid Type = &TYPE";
%END;

*****************************************************************************
*
* Convert the CAHPS individual Scores Record into WEB layout.
* There are 8 logical records (adjusted scores) per physical record:
*
* ____________________________________________
*    Adjusted Score           Definitions
*    Group Number
* ____________________________________________
* 1. Prime enrollees          XINS_COV IN (1,2,6)
* 2. Enrollees w/mil PCM      XENR_PCM IN (2,6)
* 3. Enrollees w/civ PCM      XENR_PCM = 3
* 4. Nonenrollees             XINS_COV IN (3,4,5)
* 5. Under Age 6              AGEUND6 = 1
* 6. 6-12 Years               AGE0612 = 1
* 7. 13-17 Years              AGE1317 = 1
* 8. All beneficiaries        All beneficiaries
*
*****************************************************************************;
DATA &QUESTION;
   SET IN.&QUESTION;

   LENGTH MAJGRP  $42;
   LENGTH ROWCAT  $30;
   LENGTH BENTYPE $75;
   LENGTH BENEFIT $50;

   ********************************************************************
   * For now, Initialize Significance test to zero.
   ********************************************************************;
   SIG = 0;

   ********************************************************************
   * Assign benefit and benefit type
   ********************************************************************;
   IF "&TYPE" = "INDIVIDUAL" THEN DO;
      IF DEPENDNT IN("R99041","R99007","R99057") THEN
         BENTYPE = PUT(&YEAR,$BENTYPF.);
      ELSE
         BENTYPE = PUT(DEPENDNT,$BENTYPF.);
      BENEFIT = PUT(DEPENDNT,$BENEF.);
   END;
   ELSE IF "&TYPE" = "COMPOSITE" THEN DO;
      BENTYPE = PUT(&YEAR,$BENTYPF.);
      BENEFIT = PUT(DEPENDNT,$BENEF.);
   END;
   ELSE PUT "ERROR: Invalid TYPE = &TYPE";

   ********************************************************************
   * Assign Super-Region to Major Group
   ********************************************************************;
   MAJGRP = PUT(SUPREGA,MAJGRPF.);

   ********************************************************************
   * Output score records for the eight row categories
   ********************************************************************;
   %DO I = 1 %TO 8;
      ROWCAT = PUT(&I,ROWCATF.);
      SCORE  = ADJ&I;
      SEMEAN = SEMEAN&I;
      N_OBS  = &PREFIX.CNT&I;
      N_WGT  = &PREFIX.WGT&I;
      OUTPUT;
   %END;

KEEP MAJGRP
     ROWCAT
```

```
     BENTYPE
     BENEFIT
     SCORE
     SEMEAN
     N_OBS
     N_WGT
     SIG
   ;
RUN;

%MEND;

*************************************************************************
* COMPOSITE # 1.
* GETTING NEEDED CARE VARIABLES.
*************************************************************************;
%PROCESS(QUESTION=RCOMPOS1,TYPE=COMPOSITE, REGCAT=Region);
%PROCESS(QUESTION=R8R99005,TYPE=INDIVIDUAL,REGCAT=Region);
%PROCESS(QUESTION=R8R99013,TYPE=INDIVIDUAL,REGCAT=Region);
%PROCESS(QUESTION=R8R99030,TYPE=INDIVIDUAL,REGCAT=Region);
%PROCESS(QUESTION=R8R99031,TYPE=INDIVIDUAL,REGCAT=Region);

*************************************************************************
* COMPOSITE # 2.
* GETTING CARE QUICKLY VARIABLES.
*************************************************************************;
%PROCESS(QUESTION=RCOMPOS2,TYPE=COMPOSITE, REGCAT=Region);
%PROCESS(QUESTION=R8R99018,TYPE=INDIVIDUAL,REGCAT=Region);
%PROCESS(QUESTION=R8R99020,TYPE=INDIVIDUAL,REGCAT=Region);
%PROCESS(QUESTION=R8R99023,TYPE=INDIVIDUAL,REGCAT=Region);
%PROCESS(QUESTION=R8R99032,TYPE=INDIVIDUAL,REGCAT=Region);

*************************************************************************
* COMPOSITE # 3.
* HOW WELL DOCTORS COMMUNICATE.
*************************************************************************;
%PROCESS(QUESTION=RCOMPOS3,TYPE=COMPOSITE, REGCAT=Region);
%PROCESS(QUESTION=R8R99035,TYPE=INDIVIDUAL,REGCAT=Region);
%PROCESS(QUESTION=R8R99036,TYPE=INDIVIDUAL,REGCAT=Region);
%PROCESS(QUESTION=R8R99037,TYPE=INDIVIDUAL,REGCAT=Region);
%PROCESS(QUESTION=R8R99039,TYPE=INDIVIDUAL,REGCAT=Region);
%PROCESS(QUESTION=R8R99040,TYPE=INDIVIDUAL,REGCAT=Region);

*************************************************************************
* COMPOSITE # 4.
* COURTEOUS AND HELPFUL OFFICE STAFF.
*************************************************************************;
%PROCESS(QUESTION=RCOMPOS4,TYPE=COMPOSITE, REGCAT=Region);
%PROCESS(QUESTION=R8R99033,TYPE=INDIVIDUAL,REGCAT=Region);
%PROCESS(QUESTION=R8R99034,TYPE=INDIVIDUAL,REGCAT=Region);

*************************************************************************
* COMPOSITE # 5.
* CUSTOMER SERVICE.
*************************************************************************;
%PROCESS(QUESTION=RCOMPOS5,TYPE=COMPOSITE, REGCAT=Region);
%PROCESS(QUESTION=R8R99049,TYPE=INDIVIDUAL,REGCAT=Region);
%PROCESS(QUESTION=R8R99051,TYPE=INDIVIDUAL,REGCAT=Region);
%PROCESS(QUESTION=R8R99056,TYPE=INDIVIDUAL,REGCAT=Region);

*************************************************************************
* INDIVIDUAL # 1.
* RATING OF ALL HEALTH CARE: 0 - 10.
*************************************************************************;
%PROCESS(QUESTION=R8R99041,TYPE=INDIVIDUAL,REGCAT=Region);

*************************************************************************
* INDIVIDUAL # 2.
* RATING OF HEALTH PLAN: 0 - 10.
*************************************************************************;
%PROCESS(QUESTION=R8R99057,TYPE=INDIVIDUAL,REGCAT=Region)

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

```
* INDIVIDUAL # 3.
* RATING OF PERSONAL DOCTOR: 0 - 10.
*************************************************************************;
%PROCESS(QUESTION=R8R99007,TYPE=INDIVIDUAL,REGCAT=Region);

*************************************************************************
*************************************************************************
* STACK up all of the files into one final output dataset.
*************************************************************************
*************************************************************************;
DATA OUT.LOADCAHC;
   SET R8R99005
       R8R99007
       R8R99013
       R8R99030
       R8R99031
       R8R99018
       R8R99020
       R8R99023
       R8R99032
       R8R99035
       R8R99036
       R8R99037
       R8R99039
       R8R99040
       R8R99041
       R8R99033
       R8R99034
       R8R99049
       R8R99051
       R8R99056
       R8R99057
       RCOMPOS1
       RCOMPOS2
       RCOMPOS3
       RCOMPOS4
       RCOMPOS5
    ;
    IF SCORE = . THEN DELETE;
RUN;

TITLE1 "1999 DOD Health Survey Scores/Report Cards (8687-200)";
TITLE2 "Program Name: LOADCAHC.SAS By Keith Rathbun";
TITLE3 "Program Inputs: CAHPS Individual and Composite data sets with adjusted scores";
TITLE4 "Program Outputs: LOADCAHC.SD2 - Combined CAHPS Scores Database in WEB layout";

PROC FREQ;
TABLES BENEFIT BENTYPE MAJGRP ROWCAT
      /MISSING LIST;
RUN;
_
```

4. CREATE FORMAT LIBRARY FOR 1999 CAHPS BENCHMARK DATA

```
*****************************************************************************
*
* PROGRAM: CHILD.SAS
* TASK:    DOD HEALTH CARE SURVEY ANALYSIS (8687-100)
* PURPOSE: GENERATE FORMAT LIBRARY FOR 1999 CHILD CAHPS BENCHMARK DATA
* WRITTEN: 05/30/2000 BY KEITH RATHBUN
*
* INPUT:   None
*
* OUTPUT:  1) FORMATS.SC2 - 1999 Child - CAHPS Benchmark format library
*
*****************************************************************************
*;
LIBNAME IN      "D:\KEITH\DOD99-8676\TEMP\CHILD_99_CAHPS_BENCHMARK";
LIBNAME LIBRARY "D:\KEITH\DOD99-8676\TEMP\CHILD_99_CAHPS_BENCHMARK";
OPTIONS NOCENTER LS=132 PS=79;

PROC FORMAT LIBRARY=LIBRARY;
value MODEL
                 1 = "HMO"
                 2 = "PPO"
                 3 = "POS"
                 4 = "PCCM"
                 5 = "Other"
   ;
value POPULAT
                 1 = "Adult Medicaid"
                 2 = "Adult Commercial"
                 3 = "Child Medicaid"
                 4 = "Child Commercial"
   ;
value CC01A
                 1 = "YES"
                 2 = "NO"
   ;
value CC03A
                 1 = "YES"
                 2 = "NO"
   ;
value CC04A
                 1 = "LESS THAN 12 MONTHS"
                 2 = "12 UP TO 24 MONTHS"
                 3 = "2 UP TO 5 YEARS"
                 4 = "5 UP TO 10 YEARS"
                 5 = "10 OR MORE YEARS"
   ;
value CC05A
                 1 = "YES"
                 2 = "NO"
   ;
value CC06A
                 1 = "A BIG PROBLEM"
                 2 = "A SMALL PROBLEM"
                 3 = "NOT A PROBLEM"
                 4 = "DIDN'T GET NEW DR"
   ;
value CC07A
                 1 = "YES"
                 2 = "NO"
   ;
value CC08A
                 1 = "NEVER"
                 2 = "SOMETIMES"
                 3 = "USUALLY"
                 4 = "ALWAYS"
   ;
value CC09A
                 0 = "Worst Personal MD or RN"
                 1 = "1"
                 2 = "2"
                 3 = "3"
                 4 = "4"
```

```
                 5 = "5"
                 6 = "6"
                 7 = "7"
                 8 = "8"
                 9 = "9"
                10 = "Best personal MD or RN"
                98 = "DOESN'T HAVE PERS DR"
   ;
value CC10A
                 1 = "YES"
                 2 = "NO"
   ;
value CC11A
                 1 = "A BIG PROBLEM"
                 2 = "A SMALL PROBLEM"
                 3 = "NOT A PROBLEM"
                 8 = "DIDN'T NEED TO SEE SPEC"
   ;
value CC12A
                 1 = "YES"
                 2 = "NO"
   ;
value CC13A
                 0 = "Worst specialist"
                 1 = "1"
                 2 = "2"
                 3 = "3"
                 4 = "4"
                 5 = "5"
                 6 = "6"
                 7 = "7"
                 8 = "8"
                 9 = "9"
                10 = "Best specialist"
                98 = "DIDN'T SEE A SPECIALIST"
   ;
value CC14A
                 1 = "YES"
                 2 = "NO"
                 8 = "DOESN'T HAVE PERS DR"
   ;
value CC15A
                 1 = "YES"
                 2 = "NO"
   ;
value CC16A
                 1 = "NEVER"
                 2 = "SOMETIMES"
                 3 = "USUALLY"
                 4 = "ALWAYS"
                 8 = "DIDN'T CALL"
   ;
value CC17A
                 1 = "YES"
                 2 = "NO"
   ;
value CC18A
                 1 = "NEVER"
                 2 = "SOMETIMES"
                 3 = "USUALLY"
                 4 = "ALWAYS"
                 5 = "DIDN'T NEED AN APPOINTMENT"
   ;
value CC19A
                 1 = "YES"
                 2 = "NO"
   ;
value CC20A
                 1 = "NEVER"
                 2 = "SOMETIMES"
                 3 = "USUALLY"
                 4 = "ALWAYS"
                 8 = "DIDN'T NEED CARE"
```

```
   ;
value CC21A
                 0 = "NONE"
   ;
value CC21B
                 0 = "None"
   ;
value CC22A
                 0 = "NONE"
                 1 = "1"
                 2 = "2"
                 3 = "3"
                 4 = "4"
                 5 = "5 TO 9"
                 6 = "10 OR MORE"
   ;
value CC23A
                 1 = "A BIG PROBLEM"
                 2 = "A SMALL PROBLEM"
                 3 = "NOT A PROBLEM"
                 4 = "HAD NO VISITS IN THE LAST 12 MONTHS"
   ;
value CC24A
                 1 = "A BIG PROBLEM"
                 2 = "A SMALL PROBLEM"
                 3 = "NOT A PROBLEM"
                 4 = "HAD NO VISITS"
   ;
value CC25A
                 1 = "ALWAYS"
                 2 = "USUALLY"
                 3 = "SOMETIMES"
                 4 = "NEVER"
                 5 = "I DON'T KNOW"
   ;
value CC26A
                 1 = "NEVER"
                 2 = "SOMETIMES"
                 3 = "USUALLY"
                 4 = "ALWAYS"
                 5 = "I DON'T KNOW"
                 6 = "HAD NO VISITS"
   ;
value CC27A
                 1 = "NEVER"
                 2 = "SOMETIMES"
                 3 = "USUALLY"
                 4 = "ALWAYS"
                 5 = "I DON'T KNOW"
                 6 = "HAD NO VISITS"
   ;
value CC28A
                 1 = "NEVER"
                 2 = "SOMETIMES"
                 3 = "USUALLY"
                 4 = "ALWAYS"
                 5 = "I DON'T KNOW"
                 6 = "HAD NO VISITS"
   ;
value CC29A
                 1 = "NEVER"
                 2 = "SOMETIMES"
                 3 = "USUALLY"
                 4 = "ALWAYS"
                 5 = "I DON'T KNOW"
                 6 = "HAD NO VISITS"
   ;
value CC30A
                 1 = "NEVER"
                 2 = "SOMETIMES"
                 3 = "USUALLY"
                 4 = "ALWAYS"
                 5 = "I DON'T KNOW"
```

```
                 6 = "HAD NO VISITS"
   ;
value CC31A
                 1 = "YES"
                 2 = "NO"
   ;
value CC32A
                 1 = "NEVER"
                 2 = "SOMETIMES"
                 3 = "USUALLY"
                 4 = "ALWAYS"
                 5 = "I DON'T KNOW"
                 6 = "HAD NO VISITS"
   ;
value CC33A
                 1 = "NEVER"
                 2 = "SOMETIMES"
                 3 = "USUALLY"
                 4 = "ALWAYS"
                 5 = "I DON'T KNOW"
                 6 = "HAD NO VISITS"
   ;
value CC34A
                 0 = "Worst health care"
                 1 = "1"
                 2 = "2"
                 3 = "3"
                 4 = "4"
                 5 = "5"
                 6 = "6"
                 7 = "7"
                 8 = "8"
                 9 = "9"
                10 = "Best health care"
                11 = "HAD NO VISITS"
   ;
value CC35A
                 1 = "YES"
                 2 = "NO"
   ;
value CC36A
                 1 = "A BIG PROBLEM"
                 2 = "A SMALL PROBLEM"
                 3 = "NOT A PROBLEM"
                 4 = "DIDN'T LOOK FOR INFO"
   ;
value CC37A
                 1 = "YES"
                 2 = "NO"
   ;
value CC38A
                 1 = "A BIG PROBLEM"
                 2 = "A SMALL PROBLEM"
                 3 = "NOT A PROBLEM"
                 4 = "DIDN'T CALL CUST SERV"
   ;
value CC39A
                 1 = "YES"
                 2 = "NO"
   ;
value CC40A
                 1 = "A BIG PROBLEM"
                 2 = "A SMALL PROBLEM"
                 3 = "NOT A PROBLEM"
                 4 = "DIDN'T HAVE PAPERWORK"
   ;
value CC41A
                 0 = "Worst health plan"
                 1 = "1"
                 2 = "2"
                 3 = "3"
                 4 = "4"
                 5 = "5"
```

```
                 6 = "6"
                 7 = "7"
                 8 = "8"
                 9 = "9"
                10 = "Best health plan"
   ;
value CC42A
                 1 = "EXCELLENT"
                 2 = "VERY GOOD"
                 3 = "GOOD"
                 4 = "FAIR"
                 5 = "POOR"
   ;
value CC43A
                 0 = "less than one year old"
   ;
value CC43B
                 0 = "Less than one year old"
   ;
value CC44A
                 1 = "MALE"
                 2 = "FEMALE"
   ;
value CC45A
                 1 = "HISPANIC OR LATINO"
                 2 = "NOT HISPANIC OR LATINO"
   ;
value CC46A
                 1 = "WHITE"
   ;
value CC46B
                 2 = "BLACK"
   ;
value CC46C
                 3 = "ASIAN"
   ;
value CC46D
                 4 = "NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER"
   ;
value CC46E
                 5 = "AMERICAN INDIAN OR OTHER ALASKA NATIVE"
   ;
value CC47A
                 0 = "Under 18"
                 1 = "18 TO 24"
                 2 = "25 TO 34"
                 3 = "35 TO 44"
                 4 = "45 TO 54"
                 5 = "55 TO 64"
                 6 = "65 TO 74"
                 7 = "75 OR OLDER"
   ;
value CC48A
                 1 = "MALE"
                 2 = "FEMALE"
   ;
value CC49A
                 1 = "8TH GRADE OR LESS"
                 2 = "SOME HIGH SCHOOL, BUT DID NOT GRADUATE"
                 3 = "HIGH SCHOOL GRADUATE OR GED"
                 4 = "SOME COLLEGE OR 2-YEAR DEGREE"
                 5 = "4-YEAR COLLEGE GRADUATE"
                 6 = "MORE THAN 4-YEAR COLLEGE DEGREE"
   ;
value CC50A
                 1 = "I AM THE POLICYHOLDER"
                 2 = "SPOUSE OR PARTNER OF POLICYHOLDER"
                 3 = "CHILD OF POLICYHOLDER"
                 4 = "OTHER FAMILY MEMBER"
                 5 = "FRIEND"
                 6 = "SOMEONE ELSE"
   ;
value CC51A
```

```
                 1 = "MOTHER OR FATHER"
                 2 = "GRANDPARENT"
                 3 = "AUNT OR UNCLE"
                 4 = "OLDER BROTHER OR SISTER"
                 5 = "OTHER RELATIVE"
                 6 = "LEGAL GUARDIAN"
   ;
value FILTER__
                 0 = "Not Selected"
                 1 = "Selected"
   ;
value CC16_3A
                 1 = "NEVER/SOMETIMES"
                 2 = "USUALLY"
                 3 = "ALWAYS"
   ;
value CC18_3A
                 1 = "NEVER/SOMETIMES"
                 2 = "USUALLY"
                 3 = "ALWAYS"
   ;
value CC20_3A
                 1 = "NEVER/SOMETIMES"
                 2 = "USUALLY"
                 3 = "ALWAYS"
   ;
value CC25_3A
                 1 = "ALWAYS/USUALLY"
                 2 = "SOMETIMES"
                 3 = "NEVER"
   ;
value CC28_3A
                 1 = "NEVER/SOMETIMES"
                 2 = "USUALLY"
                 3 = "ALWAYS"
   ;
value CC29_3A
                 1 = "NEVER/SOMETIMES"
                 2 = "USUALLY"
                 3 = "ALWAYS"
   ;
value CC30_3A
                 1 = "NEVER/SOMETIMES"
                 2 = "USUALLY"
                 3 = "ALWAYS"
   ;
value CC33_3A
                 1 = "NEVER/SOMETIMES"
                 2 = "USUALLY"
                 3 = "ALWAYS"
   ;
value CC26_3A
                 1 = "NEVER/SOMETIMES"
                 2 = "USUALLY"
                 3 = "ALWAYS"
   ;
value CC27_3A
                 1 = "NEVER/SOMETIMES"
                 2 = "USUALLY"
                 3 = "ALWAYS"
   ;
value CC09_3A
                 1 = "0-6"
                 2 = "7-8"
                 3 = "9-10"
   ;
value CC13_3A
                 1 = "0-6"
                 2 = "7-8"
                 3 = "9-10"
   ;
value CC34_3A
                 1 = "0-6"
```

```
                 2 = "7-8"
                 3 = "9-10"
   ;
value CC41_3A
                 1 = "0-6"
                 2 = "7-8"
                 3 = "9-10"
   ;
value Q06_P2A
                 0 = "A SMALL PROBLEM/ A BIG PROBLEM"
                 1 = "NOT A PROBLEM"
   ;
value Q11_P2A
                 0 = "A SMALL PROBLEM/ A BIG PROBLEM"
                 1 = "NOT A PROBLEM"
   ;
value Q23_P2A
                 0 = "A SMALL PROBLEM/ A BIG PROBLEM"
                 1 = "NOT A PROBLEM"
   ;
value Q24_P2A
                 0 = "A SMALL PROBLEM/ A BIG PROBLEM"
                 1 = "NOT A PROBLEM"
   ;
value Q16_P2A
                 0 = "USUALLY/SOMETIMES/NEVER"
                 1 = "ALWAYS"
   ;
value Q18_P2A
                 0 = "USUALLY/SOMETIMES/NEVER"
                 1 = "ALWAYS"
   ;
value Q25_P2A
                 0 = "SOMETIMES/USUALLY/ALWAYS"
                 1 = "NEVER"
   ;
value Q28_P2A
                 0 = "USUALLY/SOMETIMES/NEVER"
                 1 = "ALWAYS"
   ;
value Q29_P2A
                 0 = "USUALLY/SOMETIMES/NEVER"
                 1 = "ALWAYS"
   ;
value Q30_P2A
                 0 = "USUALLY/SOMETIMES/NEVER"
                 1 = "ALWAYS"
   ;
value Q33_P2A
                 0 = "USUALLY/SOMETIMES/NEVER"
                 1 = "ALWAYS"
   ;
value Q26_P2A
                 0 = "USUALLY/SOMETIMES/NEVER"
                 1 = "ALWAYS"
   ;
value Q27_P2A
                 0 = "USUALLY/SOMETIMES/NEVER"
                 1 = "ALWAYS"
   ;
value Q36_P2A
                 0 = "A SMALL PROBLEM/ A BIG PROBLEM"
                 1 = "NOT A PROBLEM"
   ;
value Q38_P2A
                 0 = "A SMALL PROBLEM/ A BIG PROBLEM"
                 1 = "NOT A PROBLEM"
   ;
value Q40_P2A
                 0 = "A SMALL PROBLEM/ A BIG PROBLEM"
                 1 = "NOT A PROBLEM"
   ;
value BHEALTH
```

```
                 1 = "Fair/poor"
                 2 = "Good"
                 3 = "Very good"
                 4 = "Excellent"
   ;
value CC43X
                 0 = "less than one year old"
   ;
value CC43BX
                 0 = "Less than one year old"
   ;
value CC32_3A
                 1 = "NEVER/SOMETIMES"
                 2 = "USUALLY"
                 3 = "ALWAYS"
   ;
value Q32_P2A
                 0 = "USUALLY/SOMETIMES/NEVER"
                 1 = "ALWAYS"
   ;
RUN;

TITLE1 "1999 Child CAHPS Benchmark Data (8687-100)";
TITLE2 "Program Name: CHILD.SAS By Keith Rathbun";
TITLE3 "Purpose: Generate Format Library for 1999 Child CAHPS Benchmark Data";

PROC CONTENTS DATA=IN.CHILD; RUN;
PROC PRINT    DATA=IN.CHILD(OBS=20); RUN;
_
```

5. CALCULATE 1999 CAHPS BENCHMARK DATA FOR 1999 CHILD HCSDB

```
***************************************************************************
*
* PROGRAM:  BENCHC01.SAS
* TASK:     1999 DOD HEALTH CARE SURVEY ANALYSIS (8687-200)
* PURPOSE:  Extract 1999 Child CAHPS Questions
*
* WRITTEN:  07/14/2000 BY KEITH RATHBUN
*
* INPUTS:   1) CHILD.SD2 - 1999 Child CAHPS Questions
*
* OUTPUT:   1) BENCHC01.SD2 - 1999 Child CAHPS Questions Renamed to be
*              consistent with the 1999 MPR DOD Survey.
*
* NOTES:
*
* 1) This program will generate the input for BENCHC02.SAS.
*
***************************************************************************
* Assign data libraries and options
***************************************************************************;
LIBNAME IN      "j:\8687\data1999\child_99_cahps_benchmark";
LIBNAME OUT     "j:\8687\programs\benchmark";
OPTIONS PS=79 LS=132 COMPRESS=YES NOCENTER;

DATA OUT.BENCHC01;
   SET IN.CHILD;
   FORMAT _ALL_;

   C99C05   = CC06;
   C99C13   = CC11;
   C99C30   = CC23;
   C99C31   = CC24;
   C99C18   = CC16;
   C99C20   = CC18;
   C99C23   = CC20;
   C99C32   = CC25;
   C99C35   = CC28;
   C99C36   = CC29;
   C99C37   = CC30;
   C99C39   = CC32;
   C99C40   = CC33;
   C99C33   = CC26;
   C99C34   = CC27;
   C99C49   = CC36;
   C99C51   = CC38;
   C99C56   = CC40;
   C99D65   = CC35;
   C99C41   = CC34;
   C99C07   = CC09;
   C99C57   = CC41;
   AGEGROUP = CC47; * Parents age grouping;
   ZAGE     = Cage; * Childs Age;
   XSEXA    = CC44; * Childs Sex;
   SREDHIGH = CC49; * Parents Education Level;
   C99D61   = CC42; * Childs health status;

   LABEL C99C05   = "CC06 - CAHPS variable"
         C99C13   = "CC11 - CAHPS variable"
         C99C30   = "CC23 - CAHPS variable"
         C99C31   = "CC24 - CAHPS variable"
         C99C18   = "CC16 - CAHPS variable"
         C99C20   = "CC18 - CAHPS variable"
         C99C23   = "CC20 - CAHPS variable"
         C99C32   = "CC25 - CAHPS variable"
         C99C35   = "CC28 - CAHPS variable"
         C99C36   = "CC29 - CAHPS variable"
         C99C37   = "CC30 - CAHPS variable"
         C99C39   = "CC32 - CAHPS variable"
         C99C40   = "CC33 - CAHPS variable"
         C99C33   = "CC26 - CAHPS variable"
         C99C34   = "CC27 - CAHPS variable"
         C99C49   = "CC36 - CAHPS variable"
```

```
          C99C51   = "CC38 - CAHPS variable"
          C99C56   = "CC40 - CAHPS variable"
          C99D65   = "CC35 - CAHPS variable"
          C99C41   = "CC34 - CAHPS variable"
          C99C07   = "CC09 - CAHPS variable"
          C99C57   = "CC41 - CAHPS variable"
          AGEGROUP = "CC47 - CAHPS variable" /* Parents Age Grouping     */
          ZAGE     = "CC43 - CAHPS variable" /* Childs Age Variable      */
          XSEXA    = "CC44 - CAHPS variable" /* Childs Sex               */
          SREDHIGH = "CC49 - CAHPS variable" /* Parent Education Level   */
          C99D61   = "CC42 - CAHPS variable" /* Childs Health Status     */
         ;
    KEEP C99C05
         C99C13
         C99C30
         C99C31
         C99C18
         C99C20
         C99C23
         C99C32
         C99C35
         C99C36
         C99C37
         C99C39
         C99C40
         C99C33
         C99C34
         C99C49
         C99C51
         C99C56
         C99D65
         C99C41
         C99C07
         C99C57
         AGEGROUP
         ZAGE
         XSEXA
         SREDHIGH
         C99D61
         PRODUCT
;
RUN;

TITLE1 "Extract 1999 Child CAHPS Questions (8687-200)";
TITLE2 "Program Name: BENCHC01.SAS By Keith Rathbun";
TITLE3 "Program Input: CHILD.SD2";
TITLE4 "Program Output: BENCHC01.SD2";

PROC CONTENTS; RUN;

PROC FREQ;
TABLES _ALL_ /MISSING LIST;
RUN;
```

```
*************************************************************************
*
* PROGRAM:  BENCHC02.SAS
* TASK:     1999 DOD HEALTH CARE SURVEY ANALYSIS (8687-200)
* PURPOSE:  Recode 1999 Child CAHPS Questions
*
* WRITTEN:  07/17/2000 BY KEITH RATHBUN
*
* INPUT:    1) BENCHC01.SD2 - 1999 Child CAHPS Questions Renamed to be
*              consistent with the 1999 MPR DOD Survey.
*
* OUTPUT:   1) BENCHC02.SD2 - Recoded 1999 Child CAHPS Questions Renamed
*              to be consistent with the 1999 MPR DOD Survey.
*
* NOTES:
*
* 1) Run this program after BENCHC01.SAS.
* 2) This program will generate the input for BENCHC03.SAS.
*
*************************************************************************
* Assign data libraries and options
*************************************************************************;
LIBNAME IN      "j:\8687\programs\benchmark";
LIBNAME OUT     ".";
OPTIONS PS=79 LS=132 COMPRESS=YES NOCENTER;

DATA OUT.BENCHC02;
   SET IN.BENCHC01;

   ************************************************************************
   * Reorder the code for C99C32.
   * It is in reverse order.
   ************************************************************************;
   IF C99C32 = 1      THEN R99032 = 4;
   ELSE IF C99C32 = 2 THEN R99032 = 3;
   ELSE IF C99C32 = 3 THEN R99032 = 2;
   ELSE IF C99C32 = 4 THEN R99032 = 1;
   ELSE IF C99C32 < 0 THEN R99032 = .;
      IF C99C32 > 1   THEN R99032 = R99032 - 1;

   ************************************************************************
   * Recode variables with Never, Sometimes, Usually and Always.
   * Recode Never & Sometimes (1 & 2) to 1.
   * Recode Usually (3) to 2.
   * Recode Always (4) to 3.
   ************************************************************************;
   IF C99C18 = 1      THEN R99018 = 1;
   ELSE IF C99C18 = 2 THEN R99018 = 1;
   ELSE IF C99C18 = 3 THEN R99018 = 2;
   ELSE IF C99C18 = 4 THEN R99018 = 3;
   ELSE IF C99C18 < 0 THEN R99018 = .;

   IF C99C20 = 1      THEN R99020 = 1;
   ELSE IF C99C20 = 2 THEN R99020 = 1;
   ELSE IF C99C20 = 3 THEN R99020 = 2;
   ELSE IF C99C20 = 4 THEN R99020 = 3;
   ELSE IF C99C20 < 0 THEN R99020 = .;

   IF C99C23 = 1      THEN R99023 = 1;
   ELSE IF C99C23 = 2 THEN R99023 = 1;
   ELSE IF C99C23 = 3 THEN R99023 = 2;
   ELSE IF C99C23 = 4 THEN R99023 = 3;
   ELSE IF C99C23 < 0 THEN R99023 = .;

   IF C99C32 = 1      THEN R99032 = 1;
   ELSE IF C99C32 = 2 THEN R99032 = 1;
   ELSE IF C99C32 = 3 THEN R99032 = 2;
   ELSE IF C99C32 = 4 THEN R99032 = 3;
   ELSE IF C99C32 < 0 THEN R99032 = .;

   IF C99C35 = 1      THEN R99035 = 1;
   ELSE IF C99C35 = 2 THEN R99035 = 1;
   ELSE IF C99C35 = 3 THEN R99035 = 2;
```

```
ELSE IF C99C35 = 4 THEN R99035 = 3;
ELSE IF C99C35 < 0 THEN R99035 = .;

IF C99C36 = 1      THEN R99036 = 1;
ELSE IF C99C36 = 2 THEN R99036 = 1;
ELSE IF C99C36 = 3 THEN R99036 = 2;
ELSE IF C99C36 = 4 THEN R99036 = 3;
ELSE IF C99C36 < 0 THEN R99036 = .;

IF C99C37 = 1      THEN R99037 = 1;
ELSE IF C99C37 = 2 THEN R99037 = 1;
ELSE IF C99C37 = 3 THEN R99037 = 2;
ELSE IF C99C37 = 4 THEN R99037 = 3;
ELSE IF C99C37 < 0 THEN R99037 = .;

IF C99C39 = 1      THEN R99039 = 1;
ELSE IF C99C39 = 2 THEN R99039 = 1;
ELSE IF C99C39 = 3 THEN R99039 = 2;
ELSE IF C99C39 = 4 THEN R99039 = 3;
ELSE IF C99C39 < 0 THEN R99039 = .;

IF C99C40 = 1      THEN R99040 = 1;
ELSE IF C99C40 = 2 THEN R99040 = 1;
ELSE IF C99C40 = 3 THEN R99040 = 2;
ELSE IF C99C40 = 4 THEN R99040 = 3;
ELSE IF C99C40 < 0 THEN R99040 = .;

IF C99C33 = 1      THEN R99033 = 1;
ELSE IF C99C33 = 2 THEN R99033 = 1;
ELSE IF C99C33 = 3 THEN R99033 = 2;
ELSE IF C99C33 = 4 THEN R99033 = 3;
ELSE IF C99C33 < 0 THEN R99033 = .;

IF C99C34 = 1      THEN R99034 = 1;
ELSE IF C99C34 = 2 THEN R99034 = 1;
ELSE IF C99C34 = 3 THEN R99034 = 2;
ELSE IF C99C34 = 4 THEN R99034 = 3;
ELSE IF C99C34 < 0 THEN R99034 = .;

IF C99D61 = 1      THEN R99061 = 5;
ELSE IF C99D61 = 2 THEN R99061 = 4;
ELSE IF C99D61 = 3 THEN R99061 = 3;
ELSE IF C99D61 = 4 THEN R99061 = 2;
ELSE IF C99D61 = 5 THEN R99061 = 1;
ELSE IF C99D61 > 5 | C99D61 < 1 THEN R99061 = .;

************************************************************************
* Recode variables to one missing condition "."
* This also renames all the "C99xxx" to 'R99xxx".
************************************************************************;
R99005 = C99C05;  IF R99005 < 0 THEN R99005 = .;
R99013 = C99C13;  IF R99013 < 0 THEN R99013 = .;
R99030 = C99C30;  IF R99030 < 0 THEN R99030 = .;
R99031 = C99C31;  IF R99031 < 0 THEN R99031 = .;
R99049 = C99C49;  IF R99049 < 0 THEN R99049 = .;
R99051 = C99C51;  IF R99051 < 0 THEN R99051 = .;
R99056 = C99C56;  IF R99056 < 0 THEN R99056 = .;
R99041 = C99C41;  IF R99041 < 0 THEN R99041 = .;
R99007 = C99C07;  IF R99007 < 0 THEN R99007 = .;
R99057 = C99C57;  IF R99057 < 0 THEN R99057 = .;

LABEL R99005   = "CC06 - Recoded CAHPS variable"
      R99013   = "CC11 - Recoded CAHPS variable"
      R99030   = "CC23 - Recoded CAHPS variable"
      R99031   = "CC24 - Recoded CAHPS variable"
      R99018   = "CC16 - Recoded CAHPS variable"
      R99020   = "CC18 - Recoded CAHPS variable"
      R99023   = "CC20 - Recoded CAHPS variable"
      R99032   = "CC25 - Recoded CAHPS variable"
      R99035   = "CC28 - Recoded CAHPS variable"
      R99036   = "CC29 - Recoded CAHPS variable"
      R99037   = "CC30 - Recoded CAHPS variable"
      R99039   = "CC32 - Recoded CAHPS variable"
```

```
         R99040   = "CC33 - Recoded CAHPS variable"
         R99033   = "CC26 - Recoded CAHPS variable"
         R99034   = "CC27 - Recoded CAHPS variable"
         R99049   = "CC36 - Recoded CAHPS variable"
         R99051   = "CC38 - Recoded CAHPS variable"
         R99056   = "CC40 - Recoded CAHPS variable"
         R99041   = "CC34 - Recoded CAHPS variable"
         R99007   = "CC09 - Recoded CAHPS variable"
         R99057   = "CC41 - Recoded CAHPS variable"
         R99061   = "CC42 - Recoded CAHPS variable"
         PRODUCT  = "Product ID - CAHPS variable";
       ;
RUN;

TITLE1 "Recode 1999 Child CAHPS Questions (8687-200)";
TITLE2 "Program Name: BENCHC02.SAS By Keith Rathbun";
TITLE3 "Program Input: BENCHC01.SD2";
TITLE4 "Program Output: BENCHC02.SD2";

PROC CONTENTS; RUN;

PROC FREQ;
TABLES  AGEGROUP
        ZAGE
        XSEXA
        SREDHIGH
        C99C05 * R99005
        C99C13 * R99013
        C99C30 * R99030
        C99C31 * R99031
        C99C18 * R99018
        C99C20 * R99020
        C99C23 * R99023
        C99C32 * R99032
        C99C35 * R99035
        C99C36 * R99036
        C99C37 * R99037
        C99C39 * R99039
        C99C40 * R99040
        C99C33 * R99033
        C99C34 * R99034
        C99C49 * R99049
        C99C51 * R99051
        C99C56 * R99056
        C99C41 * R99041
        C99C07 * R99007
        C99C57 * R99057
        C99D61 * R99061
   /MISSING LIST;
RUN;
_
```

```
*************************************************************************
*
* PROGRAM:  BENCHC03.SAS
* TASK:     1999 DOD HEALTH CARE SURVEY ANALYSIS (8687-200)
* PURPOSE:  Recode 1999 Child CAHPS Questions
*
* WRITTEN:  June 2000 BY ERIC SCHONE
*
* MODIFIED: 1) June 2000 BY KEITH RATHBUN - Output permanent datasets with
*           scores and standard errors and process the rest of the
*           composites and ratings.
*           2) July 2000 BY KEITH RATHBUN - Update macro calls for child
*           survey.  Update to accomodate a 5th dependent variable.
*
* INPUT:    1) BENCHC02.SD2 - 1999 Child CAHPS Questions Renamed to be
*              consistent with the 1999 MPR DOD Survey.
*
* OUTPUTS:  1) Benchmark Composite Scores Data Sets
*
* NOTES:
*
* 1) Run this program after BENCHC02.SAS.
* 2) This program will generate the input for BENCHC04.SAS.
*
*************************************************************************
* Assign data libraries and options
*************************************************************************;
libname in  'j:\8687\programs\benchmark';
libname in2 'j:\8687\programs\reportcards\cahps_child99\data';
libname out 'j:\8687\programs\benchmark\datachld';

%let wgt=wrwt99;

OPTIONS MLOGIC MPRINT NOCENTER LS=132 PS=79;


%macro comb(f,t,q,l);

proc summary data=&f;
 var &t;
 where &q~=.;
 weight &wgt;
 output out=temp mean=&t;
run;

data temp;
 set temp;
 array old &t;
 call symput('z',left(dim(old)));
run;

data temp(drop=_type_ &t);
 set temp;
 array old &t;
 array new var1-var&z;
  do i=1 to &z;
   new(i)=old(i);
  end;
run;

data &q._&l;
 merge temp c_&q;
 array coeffs &t;
 array means var1-var&z;
  DO I = 1 TO DIM(COEFFS);
   IF COEFFS(I) = . THEN COEFFS(I) = 0;
   IF MEANS(I) = . THEN MEANS(I) = 0;
   ADJUST + ( COEFFS(I) * MEANS(I) );
  END;

 ADJUST = ADJUST + INTERCEP;
 &q._&l=adjust;
```

```
run;

%mend comb;

%macro adjust(x,y);



proc summary data=setup;
where &x>.;
 class product;
 output out=count;
run;

data count(rename=(_freq_=denom)) count2;
 set count;
 if _type_=0 then output count2;
 else output count;
run;

data count(keep=pweight product);
 if _n_=1 then set count2;
 set count;
 pweight=denom/_freq_;
run;

data temp;
 merge count  setup; by product;

run;

proc summary data=temp;
where &x=.;
weight pweight;
var &y;
output out=temp2 mean=&y;
data temp2;
 set temp2;
 array old &y;
 call symput('z',left(dim(old)));
run;
data temp2(keep=var1-var&z);
 set temp2;
 array old &y;
 array new var1-var&z;
  do i=1 to &z;
   new(i)=old(i);
  end;
run;
data temp;
set temp;
if _n_=1 then set temp2;
 array old &y;
 array new var1-var&z;
  do i=1 to &z;
  if old(i)=. then
   old(i)=new(i);
  end;
  proc reg data=temp outest=c_&x noprint;
 model &x=&y;
 weight pweight;
 output out=r_&x r=r_&x;
run;

proc sort data=r_&x; by product;
run;

PROC DESCRIPT DATA=r_&x DESIGN=STRWR NOPRINT;
 WEIGHT pweight;
 SETENV DECWIDTH=4;
 NEST product / missunit;
 VAR R_&x;
 OUTPUT  SEMEAN / TABLECELL=DEFAULT
```

```
 FILENAME=s_&x;
RUN;

data s_&x(rename=(semean=s_&x));
 set s_&x(keep=semean);
 %do i=5 %to 5;
  %if &i=8 %then %do;

   data group8;
    set in2.group5 in2.group6 in2.group7;
   run;
   %comb(group8,&y,&x,8);
  %end;
  %else %do;
   %comb(in2.group&i,&y,&x,&i);
  %end;
 %end;


%mend adjust;

/* adjust all the variables */

%macro comp(compno,a,b,c,d,e);
 %if &a~= %then %do;
    %let n=r_&a;
    %let m=s_&a;
    %do i=5 %to 5;
        %let p&i=&a._&i;
    %end;
    %let grpnum=1;
    proc sort data=r_&a;
       by mpid;
    run;
 %end;

 %if &b~= %then %do;
    %let n=%str(&n r_&b);
    %let m=%str(&m s_&b);
    %do i=5 %to 5;
        %let p&i=%str(&&p&i &b._&i);
    %end;
    %let grpnum=2;
    proc sort data=r_&b;
       by mpid;
    run;
 %end;

 %if &c~= %then %do;
    %let n=%str(&n r_&c);
    %let m=%str(&m s_&c);
    %do i=5 %to 5;
        %let p&i=%str(&&p&i &c._&i);
    %end;
    %let grpnum=3;
    proc sort data=r_&c;
       by mpid;
    run;
 %end;

 %if &d~= %then %do;
    %let n=%str(&n r_&d);
    %let m=%str(&m s_&d);
    %do i=5 %to 5;
        %let p&i=%str(&&p&i &d._&i);
    %end;
    %let grpnum=4;
    proc sort data=r_&d;
       by mpid;
    run;
 %end;

 %if &e~= %then %do;
```

```
    %let n=%str(&n r_&e);
    %let m=%str(&m s_&e);
    %do i=5 %to 5;
        %let p&i=%str(&&p&i &e._&i);
    %end;
    %let grpnum=5;
    proc sort data=r_&e;
       by mpid;
    run;
 %end;

data infile;
 merge &n;
 by mpid;
run;

proc corr outp=outf noprint;
 var &n;
 weight pweight;
run;

data final;
 if _n_=1 then do;
  %if &a~= %then %do;
   set s_&a;
  %end;
  %if &b~= %then %do;
   set s_&b;
  %end;
  %if &c~= %then %do;
   set s_&c;
  %end;
  %if &d~= %then %do;
   set s_&d;
  %end;
  %if &e~= %then %do;
   set s_&e;
  %end;
 end;
 set  outf;
 call symput('s'||compress(_n_),substr(_name_,3));
 where _type_='CORR';
run;

data final;
 set final;
 array r_val &n;
 array s_val &m;
 sde=0;
 do i=1 to dim(s_val);
  %do i=1 %to &grpnum;
   if _name_="R_&&s&i" then
   sde=sde+r_val(i)*s_&&s&i*s_val(i);
  %end;
 end;
run;

data sefin&compno;
 set final end=last;
 tv+sde;
 if last then do;
 sde=(tv**.5)/&grpnum;
 output;
end;

%do i=5 %to 5;
 data temp(keep=&&p&i);
  merge &&p&i;
 run;

data output;
 set &&p&i;
 totadj+adjust;
```

```
run;

data output(keep=totadj);
 set output end=last;
 if last then do;
  totadj=totadj/&grpnum;
  output;
 end;
run;

data out&compno._&i;
 merge output temp;
run;

data out.comp&compno._&i;
   merge out&compno._&i
         sefin&compno;
run;

%end;

%mend comp;
/* create composites */
proc sort data=in.benchc02 out=setup;
 by product;
run;

data setup;
 set setup; by product;
 mpid=_n_;
 IF (ZAGE NE . AND ZAGE NE 255) THEN DO;
      AGEUND6 = 0;
      AGE0612 = 0;
      AGE1317 = 0;

      IF      (ZAGE < 6)         THEN AGEUND6 = 1;
      ELSE IF (6  <= ZAGE <= 12) THEN AGE0612 = 1;
      ELSE IF (13 <= ZAGE <= 17) THEN AGE1317 = 1;
   END;
if agegroup ne . then do;
 ageund18=0; age1824=0; age2534=0; age3544=0; age4554=0; age5564=0; age6574=0;

      if agegroup=0 then ageund18 = 1;
 else if agegroup=1 then age1824  = 1;
 else if agegroup=2 then age2534  = 1;
 else if agegroup=3 then age3544  = 1;
 else if agegroup=4 then age4554  = 1;
 else if agegroup=5 then age5564  = 1;
 else if agegroup=6 then age6574  = 1;
 end;
 if ageund6=1;
run;

/* GETTING NEEDED CARE */
%adjust(r99005,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99013,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99030,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99031,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(1,r99005,r99013,r99030,r99031);

/* GETTING NEEDED CARE QUICKLY */
%adjust(r99018,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99020,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99023,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99032,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(2,r99018,r99020,r99023,r99032);

/* HOW WELL DOCTORS COMMUNICATE */
%adjust(r99035,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99036,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99037,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99039,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99040,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
```

```
%comp(3,r99035,r99036,r99037,r99039,r99040);

/* COURTEOUS AND HELPFUL OFFICE STAFF */
%adjust(r99033,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99034,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(4,r99033,r99034);

/* CUSTOMER SERVICE */
%adjust(r99049,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99051,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99056,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(5,r99049,r99051,r99056);

/* RATING ALL HEALTH CARE: 0 - 10 */
%adjust(r99041,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(6,r99041);

/* RATING OF HEALTH PLAN: 0 - 10 */
%adjust(r99057,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(7,r99057);

/* RATING OF PERSONAL DR: 0 - 10 */
%adjust(r99007,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(8,r99007);
```

```
*************************************************************************
*
* PROGRAM:  BENCHC03.SAS
* TASK:     1999 DOD HEALTH CARE SURVEY ANALYSIS (8687-200)
* PURPOSE:  Recode 1999 Child CAHPS Questions
*
* WRITTEN:  June 2000 BY ERIC SCHONE
*
* MODIFIED: 1) June 2000 BY KEITH RATHBUN - Output permanent datasets with
*           scores and standard errors and process the rest of the
*           composites and ratings.
*           2) July 2000 BY KEITH RATHBUN - Update macro calls for child
*           survey.  Update to accomodate a 5th dependent variable.
*
* INPUT:    1) BENCHC02.SD2 - 1999 Child CAHPS Questions Renamed to be
*              consistent with the 1999 MPR DOD Survey.
*
* OUTPUTS:  1) Benchmark Composite Scores Data Sets
*
* NOTES:
*
* 1) Run this program after BENCHC02.SAS.
* 2) This program will generate the input for BENCHC04.SAS.
*
*************************************************************************
* Assign data libraries and options
*************************************************************************;
libname in  'j:\8687\programs\benchmark';
libname in2 'j:\8687\programs\reportcards\cahps_child99\data';
libname out 'j:\8687\programs\benchmark\datachld';

%let wgt=wrwt99;

OPTIONS MLOGIC MPRINT NOCENTER LS=132 PS=79;


%macro comb(f,t,q,l);

proc summary data=&f;
 var &t;
 where &q~=.;
 weight &wgt;
 output out=temp mean=&t;
run;

data temp;
 set temp;
 array old &t;
 call symput('z',left(dim(old)));
run;

data temp(drop=_type_ &t);
 set temp;
 array old &t;
 array new var1-var&z;
  do i=1 to &z;
   new(i)=old(i);
  end;
run;

data &q._&l;
 merge temp c_&q;
 array coeffs &t;
 array means var1-var&z;
  DO I = 1 TO DIM(COEFFS);
   IF COEFFS(I) = . THEN COEFFS(I) = 0;
   IF MEANS(I) = . THEN MEANS(I) = 0;
   ADJUST + ( COEFFS(I) * MEANS(I) );
  END;

 ADJUST = ADJUST + INTERCEP;
 &q._&l=adjust;
```

```
run;

%mend comb;

%macro adjust(x,y);



proc summary data=setup;
where &x>.;
 class product;
 output out=count;
run;

data count(rename=(_freq_=denom)) count2;
 set count;
 if _type_=0 then output count2;
 else output count;
run;

data count(keep=pweight product);
 if _n_=1 then set count2;
 set count;
 pweight=denom/_freq_;
run;

data temp;
 merge count  setup; by product;

run;

proc summary data=temp;
where &x=.;
weight pweight;
var &y;
output out=temp2 mean=&y;
data temp2;
 set temp2;
 array old &y;
 call symput('z',left(dim(old)));
run;
data temp2(keep=var1-var&z);
 set temp2;
 array old &y;
 array new var1-var&z;
  do i=1 to &z;
   new(i)=old(i);
  end;
run;
data temp;
set temp;
if _n_=1 then set temp2;
 array old &y;
 array new var1-var&z;
  do i=1 to &z;
  if old(i)=. then
   old(i)=new(i);
  end;
  proc reg data=temp outest=c_&x noprint;
 model &x=&y;
 weight pweight;
 output out=r_&x r=r_&x;
run;

proc sort data=r_&x; by product;
run;

PROC DESCRIPT DATA=r_&x DESIGN=STRWR NOPRINT;
 WEIGHT pweight;
 SETENV DECWIDTH=4;
 NEST product / missunit;
 VAR R_&x;
 OUTPUT  SEMEAN / TABLECELL=DEFAULT
```

```
 FILENAME=s_&x;
RUN;

data s_&x(rename=(semean=s_&x));
 set s_&x(keep=semean);
 %do i=6 %to 6;
  %if &i=8 %then %do;

   data group8;
    set in2.group5 in2.group6 in2.group7;
   run;
   %comb(group8,&y,&x,8);
  %end;
  %else %do;
   %comb(in2.group&i,&y,&x,&i);
  %end;
 %end;


%mend adjust;

/* adjust all the variables */

%macro comp(compno,a,b,c,d,e);
 %if &a~= %then %do;
    %let n=r_&a;
    %let m=s_&a;
    %do i=6 %to 6;
        %let p&i=&a._&i;
    %end;
    %let grpnum=1;
    proc sort data=r_&a;
       by mpid;
    run;
 %end;

 %if &b~= %then %do;
    %let n=%str(&n r_&b);
    %let m=%str(&m s_&b);
    %do i=6 %to 6;
        %let p&i=%str(&&p&i &b._&i);
    %end;
    %let grpnum=2;
    proc sort data=r_&b;
       by mpid;
    run;
 %end;

 %if &c~= %then %do;
    %let n=%str(&n r_&c);
    %let m=%str(&m s_&c);
    %do i=6 %to 6;
        %let p&i=%str(&&p&i &c._&i);
    %end;
    %let grpnum=3;
    proc sort data=r_&c;
       by mpid;
    run;
 %end;

 %if &d~= %then %do;
    %let n=%str(&n r_&d);
    %let m=%str(&m s_&d);
    %do i=6 %to 6;
        %let p&i=%str(&&p&i &d._&i);
    %end;
    %let grpnum=4;
    proc sort data=r_&d;
       by mpid;
    run;
 %end;

 %if &e~= %then %do;
```

```
    %let n=%str(&n r_&e);
    %let m=%str(&m s_&e);
    %do i=6 %to 6;
        %let p&i=%str(&&p&i &e._&i);
    %end;
    %let grpnum=5;
    proc sort data=r_&e;
       by mpid;
    run;
 %end;

data infile;
 merge &n;
 by mpid;
run;

proc corr outp=outf noprint;
 var &n;
 weight pweight;
run;

data final;
 if _n_=1 then do;
  %if &a~= %then %do;
   set s_&a;
  %end;
  %if &b~= %then %do;
   set s_&b;
  %end;
  %if &c~= %then %do;
   set s_&c;
  %end;
  %if &d~= %then %do;
   set s_&d;
  %end;
  %if &e~= %then %do;
   set s_&e;
  %end;
 end;
 set  outf;
 call symput('s'||compress(_n_),substr(_name_,3));
 where _type_='CORR';
run;

data final;
 set final;
 array r_val &n;
 array s_val &m;
 sde=0;
 do i=1 to dim(s_val);
  %do i=1 %to &grpnum;
   if _name_="R_&&s&i" then
   sde=sde+r_val(i)*s_&&s&i*s_val(i);
  %end;
 end;
run;

data sefin&compno;
 set final end=last;
 tv+sde;
 if last then do;
 sde=(tv**.5)/&grpnum;
 output;
end;

%do i=6 %to 6;
 data temp(keep=&&p&i);
  merge &&p&i;
 run;

data output;
 set &&p&i;
 totadj+adjust;
```

```
run;

data output(keep=totadj);
 set output end=last;
 if last then do;
  totadj=totadj/&grpnum;
  output;
 end;
run;

data out&compno._&i;
 merge output temp;
run;

data out.comp&compno._&i;
   merge out&compno._&i
         sefin&compno;
run;

%end;

%mend comp;
/* create composites */
proc sort data=in.benchc02 out=setup;
 by product;
run;

data setup;
 set setup; by product;
 mpid=_n_;
 IF (ZAGE NE . AND ZAGE NE 255) THEN DO;
      AGEUND6 = 0;
      AGE0612 = 0;
      AGE1317 = 0;

      IF      (ZAGE < 6)         THEN AGEUND6 = 1;
      ELSE IF (6  <= ZAGE <= 12) THEN AGE0612 = 1;
      ELSE IF (13 <= ZAGE <= 17) THEN AGE1317 = 1;
   END;
if agegroup ne . then do;
 ageund18=0; age1824=0; age2534=0; age3544=0; age4554=0; age5564=0; age6574=0;

      if agegroup=0 then ageund18 = 1;
 else if agegroup=1 then age1824  = 1;
 else if agegroup=2 then age2534  = 1;
 else if agegroup=3 then age3544  = 1;
 else if agegroup=4 then age4554  = 1;
 else if agegroup=5 then age5564  = 1;
 else if agegroup=6 then age6574  = 1;
 end;
 if age0612=1;
run;

/* GETTING NEEDED CARE */
%adjust(r99005,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99013,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99030,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99031,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(1,r99005,r99013,r99030,r99031);

/* GETTING NEEDED CARE QUICKLY */
%adjust(r99018,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99020,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99023,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99032,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(2,r99018,r99020,r99023,r99032);

/* HOW WELL DOCTORS COMMUNICATE */
%adjust(r99035,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99036,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99037,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99039,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99040,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
```

```
%comp(3,r99035,r99036,r99037,r99039,r99040);

/* COURTEOUS AND HELPFUL OFFICE STAFF */
%adjust(r99033,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99034,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(4,r99033,r99034);

/* CUSTOMER SERVICE */
%adjust(r99049,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99051,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%adjust(r99056,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(5,r99049,r99051,r99056);

/* RATING ALL HEALTH CARE: 0 - 10 */
%adjust(r99041,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(6,r99041);

/* RATING OF HEALTH PLAN: 0 - 10 */
%adjust(r99057,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(7,r99057);

/* RATING OF PERSONAL DR: 0 - 10 */
%adjust(r99007,ageund18 age1824 age2534 age3544 age4554 age5564 age6574 r99061);
%comp(8,r99007);
```

```
*************************************************************************
*
* PROGRAM:  BENCHC04.SAS
* TASK:     1999 DOD HEALTH CARE SURVEY ANALYSIS (8687-200)
* PURPOSE:  Convert the Benchmark Scores Database into the WEB layout
*
* WRITTEN:  07/17/2000 BY KEITH RATHBUN
*
* INPUTS:   1) Benchmark data sets with adjusted scores
*              (COMPn_i.SD2 where n = composite number and i = group number)
*
* OUTPUT:   1) BENCHC04.SD2 - Combined Benchmark Scores Database in WEB layout
*
* INCLUDES: 1) LOADCAHC.INC - Format definitions for CAHPS Individual
*              and composite data sets
*
* NOTES:
*
* 1) The following steps need to be run prior to this program:
*    - BENCHC01.SAS - Extract Benchmark variables
*    - BENCHC02.SAS - Recode Benchmark variables
*    - BENCHC03.SAS - Construct Scores and SEMEAN datasets
*
* 2) The output file (BENCHC04.SD2) will be run through the
*    MAKEHTML.SAS program to generate the WEB pages.
*
* 3) The child-based CAHPS composite measures are based on children 12
*    years old or younger.  Therefore, we will only produce benchmarks
*    for groups 5 (Under Age 6) and 6 (6-12 Years).
*
*************************************************************************
* Assign data libraries and options
*************************************************************************;
LIBNAME IN  "DATACHLD";
LIBNAME OUT "DATACHLD";
LIBNAME LIBRARY "J:\8687\DATA1999\CFINAL\FMTLIB";
OPTIONS PS=79 LS=132 COMPRESS=YES NOCENTER;

*************************************************************************
* Load Format definitions for CAHPS Individual and composite data sets.
*************************************************************************;
%INCLUDE "J:\8687\PROGRAMS\LOADWEB\LOADCAHC.INC";

*************************************************************************
*************************************************************************
*
* Process Macro Input Parameters:
*
* 1) CNUM = Composite or rating variable number (1-8)
* 2) GNUM = Group number (1-8)
* 3) NVAR = Number of variables in the composite
* 4) VARS = List of individual variables for composite
* 5) SE   = List of individual standard error variables
*
* ____________________________________________
*    Adjusted Score           Definitions
*    Group Number
* ____________________________________________
* 1. Prime enrollees          XINS_COV IN (1,2,6)
* 2. Enrollees w/mil PCM      XENR_PCM IN (2,6)
* 3. Enrollees w/civ PCM      XENR_PCM = 3
* 4. Nonenrollees             XINS_COV IN (3,4,5)
* 5. Under Age 6              AGEUND6 = 1
* 6. 6-12 Years               AGE0612 = 1
* 7. 13-17 Years              AGE1317 = 1
* 8. All beneficiaries        All beneficiaries
*
*************************************************************************
*************************************************************************;
%MACRO PROCESS(CNUM=, GNUM=, NVAR=, VARS=, SE=);
*************************************************************************
* Assign value for BENTYPE composite year
*************************************************************************;
```

```
%LET YEAR = "1999";

*************************************************************************
* Convert benchmark scores datasets into WEB layout.
*************************************************************************;
DATA COMP&CNUM._&GNUM;
   SET IN.COMP&CNUM._&GNUM;

   LENGTH MAJGRP  $42;
   LENGTH ROWCAT  $30;
   LENGTH BENTYPE $75;
   LENGTH BENEFIT $50;

   ********************************************************************
   * For now, assign SIG = 0
   ********************************************************************;
   SIG = 0;

   ********************************************************************
   * Assign Row Category
   ********************************************************************;
   IF      &GNUM = 5 THEN ROWCAT = "Under Age 6-Benchmark";
   ELSE IF &GNUM = 6 THEN ROWCAT = "6-12 Years-Benchmark";

   ********************************************************************
   * Assign benefit and benefit type
   ********************************************************************;
   IF      &CNUM = 1  THEN BENEFIT = "Getting Needed Care";
   ELSE IF &CNUM = 2  THEN BENEFIT = "Getting Care Quickly";
   ELSE IF &CNUM = 3  THEN BENEFIT = "How Well Doctors Communicate";
   ELSE IF &CNUM = 4  THEN BENEFIT = "Courteous and Helpful Office Staff";
   ELSE IF &CNUM = 5  THEN BENEFIT = "Customer Service";
   ELSE IF &CNUM = 6  THEN BENEFIT = "Health Care";
   ELSE IF &CNUM = 7  THEN BENEFIT = "Health Plan";
   ELSE IF &CNUM = 8  THEN BENEFIT = "Personal Doctor or Nurse";

   BENTYPE = PUT(&YEAR,$BENTYPF.);

   ********************************************************************
   * Assign composite score and SEMEAN
   ********************************************************************;
   SCORE = TOTADJ;
   SEMEAN = SDE;

   ********************************************************************
   * Assign major groups and output composite score records
   ********************************************************************;
   DO I = 0 TO 3; DROP I;
      MAJGRP  = PUT(I,MAJGRPF.);
      OUTPUT;
   END;

   ********************************************************************
   * Now, output the individual score records
   ********************************************************************;
   IF &NVAR GT 1 THEN DO;
      ARRAY ITEMS &VARS;
      ARRAY SE    &SE;
      LENGTH NAME $8;
      DO I = 1 TO DIM(ITEMS); DROP I;
         CALL VNAME(ITEMS(I),NAME);
         NAME = SUBSTR(NAME,1,6);
         SCORE  = ITEMS(I);
         SEMEAN = SE(I);
         BENTYPE = PUT(NAME,$BENTYPF.);
         DO J = 0 TO 3; DROP J;
            MAJGRP  = PUT(J,MAJGRPF.);
            OUTPUT;
         END;
      END;
   END;

KEEP MAJGRP
```

```
     ROWCAT
     BENTYPE
     BENEFIT
     SEMEAN
     SCORE
     SIG
   ;
RUN;

%MEND;

*************************************************************************
* COMPOSITE # 1.
* GETTING NEEDED CARE VARIABLES.
*************************************************************************;
%PROCESS(CNUM=1, GNUM=5, NVAR=4, VARS=R99005_5 R99013_5 R99030_5 R99031_5,
         SE=S_R99005 S_R99013 S_R99030 S_R99031);
%PROCESS(CNUM=1, GNUM=6, NVAR=4, VARS=R99005_6 R99013_6 R99030_6 R99031_6,
         SE=S_R99005 S_R99013 S_R99030 S_R99031);

*************************************************************************
* COMPOSITE # 2.
* GETTING CARE QUICKLY VARIABLES.
*************************************************************************;
%PROCESS(CNUM=2, GNUM=5, NVAR=4, VARS=R99018_5 R99020_5 R99023_5 R99032_5,
         SE=S_R99018 S_R99020 S_R99023 S_R99032);
%PROCESS(CNUM=2, GNUM=6, NVAR=4, VARS=R99018_6 R99020_6 R99023_6 R99032_6,
         SE=S_R99018 S_R99020 S_R99023 S_R99032);

*************************************************************************
* COMPOSITE # 3.
* HOW WELL DOCTORS COMMUNICATE.
*************************************************************************;
%PROCESS(CNUM=3, GNUM=5, NVAR=5, VARS=R99035_5 R99036_5 R99037_5 R99039_5 R99040_5,
         SE=S_R99035 S_R99036 S_R99037 S_R99039 S_R99040);
%PROCESS(CNUM=3, GNUM=6, NVAR=5, VARS=R99035_6 R99036_6 R99037_6 R99039_6 R99040_6,
         SE=S_R99035 S_R99036 S_R99037 S_R99039 S_R99040);

*************************************************************************
* COMPOSITE # 4.
* COURTEOUS AND HELPFUL OFFICE STAFF.
*************************************************************************;
%PROCESS(CNUM=4, GNUM=5, NVAR=2, VARS=R99033_5 R99034_5, SE=S_R99033 S_R99034);
%PROCESS(CNUM=4, GNUM=6, NVAR=2, VARS=R99033_6 R99034_6, SE=S_R99033 S_R99034);

*************************************************************************
* COMPOSITE # 5.
* CUSTOMER SERVICE.
*************************************************************************;
%PROCESS(CNUM=5, GNUM=5, NVAR=3, VARS=R99049_5 R99051_5 R99056_5,
         SE=S_R99049 S_R99051 S_R99056);
%PROCESS(CNUM=5, GNUM=6, NVAR=3, VARS=R99049_6 R99051_6 R99056_6,
         SE=S_R99049 S_R99051 S_R99056);

*************************************************************************
* INDIVIDUAL # 1.
* RATING OF ALL HEALTH CARE: 0 - 10.
*************************************************************************;
%PROCESS(CNUM=6, GNUM=5, NVAR=1, VARS=R99041_5, SE=S_R99041);
%PROCESS(CNUM=6, GNUM=6, NVAR=1, VARS=R99041_6, SE=S_R99041);

*************************************************************************
* INDIVIDUAL # 2.
* RATING OF HEALTH PLAN: 0 - 10.
*************************************************************************;
%PROCESS(CNUM=7, GNUM=5, NVAR=1, VARS=R99057_5, SE=S_R99057);
%PROCESS(CNUM=7, GNUM=6, NVAR=1, VARS=R99057_6, SE=S_R99057);

*************************************************************************
* INDIVIDUAL # 3.
* RATING OF PERSONAL DOCTOR: 0 - 10.
*************************************************************************;
%PROCESS(CNUM=8, GNUM=5, NVAR=1, VARS=R99007_5, SE=S_R99007);
```

```
%PROCESS(CNUM=8, GNUM=6, NVAR=1, VARS=R99007_6, SE=S_R99007);

*****************************************************************************
*****************************************************************************
* STACK up all of the files into one final output dataset.
*****************************************************************************
*****************************************************************************;
DATA OUT.BENCHC04;
   SET COMP1_5  COMP1_6
       COMP2_5  COMP2_6
       COMP3_5  COMP3_6
       COMP4_5  COMP4_6
       COMP5_5  COMP5_6
       COMP6_5  COMP6_6
       COMP7_5  COMP7_6
       COMP8_5  COMP8_6
    ;
    IF SCORE = . THEN DELETE;
RUN;

TITLE1 "1999 DOD Health Survey Scores/Report Cards (8687-200)";
TITLE2 "Program Name: BENCHC04.SAS By Keith Rathbun";
TITLE3 "Program Inputs: Benchmark Individual and Composite data sets with adjusted scores";
TITLE4 "Program Outputs: BENCHC04.SD2 - Combined Benchmark Scores Database in WEB layout";

PROC CONTENTS; RUN;

PROC FREQ;
TABLES BENEFIT BENTYPE MAJGRP ROWCAT
      /MISSING LIST;
RUN;
_
```

6. CALCULATE HEALTH STATUS COMPOSITE SCORES

```
*****************************************************************
*  Project:   DoD Reporting and Analysis 8687-100
*  Program:   MPRCOMPC.SAS
*  Author:    Chris Rankin
*  Date:      7/17/2000
*  Modified:
*
*  Purpose:   Calculate MPR Preventive Care Composites
*             CHILD VERSION
*  Input:     J:\8687\DATA1999\AFINAL\HCS99A_5.SD2
*  Output:    RCMPSUM.SD2
*             MCMPSUM.SD2
*  Include
*   Files:     LOADCAHP.INC
*              CACRTP.INC
*    Note:    Next program is Loadmprc.sas
*****************************************************************;

OPTIONS NOCENTER LS=124 PS=74 SOURCE SOURCE2 MLOGIC MPRINT COMPRESS=YES;

LIBNAME   IN       "J:\8687\DATA1999\CFINAL";
LIBNAME   OUT      ".";
LIBNAME LIBRARY    "J:\8687\DATA1999\CFINAL\FMTLIB";

%LET WGT=WRWT99;
%LET DEBUG=Y;      /** Set to Y for Debug print of datasets **/
%LET YR=99 ;       /** used to name variabales  for 99 & 98 **/
%LET INDATA=HCS99C_5;
%LET YRDATA=HCS99;

/***** The following parameters are used in the Variance ****/
/***** calcuation macro for region and catchment area    ****/

%LET GRPNUM=8;     /** number of groups          **/
%LET COMPNUM=10;   /** numper of variables       **/
%LET CMPNUM1=3;    /** number of variables in first compostie  **/
%LET CMPNUM2=4;    /** number of variables in second composite **/
%LET CMPNUM3=3;    /** number of variables in third composite  **/

%LET COMPCNT=3;    /** number of composites                    **/

%INCLUDE "J:\8687\PROGRAMS\LOADWEB\LOADCAHC.INC";

/**** Note:  output all data to a single dataset for macro  **/
/**** call                                                  **/
/**** Note:  for child data, all responses are in CONUS     **/

*************************************************************************
*
* Beneficiary Groups:
* ____________________________________________
*    Adjusted Score           Definitions
*    Group Number
* ____________________________________________
* 1. Prime enrollees          XINS_COV = 1
* 2. Enrollees w/mil PCM      XENR_PCM = 1
* 3. Enrollees w/civ PCM      XENR_PCM = 2
* 4. Nonenrollees             XINS_COV IN (2,3)
* 5. Under Age 6              AGEUND6 = 1
* 6. 6-12 Years               AGE0612 = 1
* 7. 13-17 Years              AGE1317 = 1
* 8. All beneficiaries        All beneficiaries
*
*************************************************************************;


 DATA &YRDATA(KEEP=BGROUP SUPREGA XREGION CACSMPL  &WGT MPRVAR1-MPRVAR&COMPNUM
              NUM&YR.V1-NUM&YR.V&COMPNUM. DEN&YR.V1-DEN&YR.V&COMPNUM.);

  SET IN.&INDATA(RENAME=(STRATUM=XSTRATUM ZAGE_D1=ZAGE));

 IF XREGION=7 THEN XREGION=8;
```

```
  IF XREGION IN (.,0)    THEN DELETE;
  IF CACSMPL IN (.,9999) THEN DELETE;

/**** note:  for PCM composite, the numerator and denominator are not *****/
/**** the same question(s)                                              *****/

  MPRVAR1=C99C08;        **  had a PCM                              ;
  MPRVAR2=C99C09;        **  know the name of PCM                   ;
  MPRVAR3=C99C10;        **  how much of a problem -- PCM           ;
  MPRVAR4=C99C60A;       **  hard to get health care svcs.         ;
  MPRVAR5=C99C60B;       **  hard to see doctor I prefer            ;
  MPRVAR6=C99C60C;       **  benefits do not meet needs             ;
  MPRVAR7=C99C60D;       **  high quality health care               ;
  MPRVAR8=C99C21;        **  days btwn. appt. and visit             ;
  MPRVAR9=C99C24;        **  wait btwn trying to get care and visit ;
  MPRVAR10=C99C27;       **  wait for well patient care             ;

/**** set up numerator and denominator for proportions ****/

  ARRAY MPRVAR(*) MPRVAR1-MPRVAR&COMPNUM;
  ARRAY  NUMER(*) NUM&YR.V1-NUM&YR.V&COMPNUM;
  ARRAY  DENOM(*) DEN&YR.V1-DEN&YR.V&COMPNUM;

   DO I=1 TO &COMPNUM;
    IF I=1 THEN DO;
    DENOM(I)=1;
     IF C99C08=1 THEN NUMER(I)=1;
     ELSE NUMER(I)=0;
    END;
    ELSE IF I IN (2,3) THEN DO;
     IF C99C08=1 THEN DENOM(I)=1;
     IF I=2 THEN DO;
      IF MPRVAR(I) = 1 THEN NUMER(I) = 1;
      ELSE NUMER(I)=0;
     END;
     ELSE IF I=3 THEN DO;
      IF MPRVAR(I)=3 THEN NUMER(I)=1;
      ELSE NUMER(I)=0;
     END;
    END;

    IF 4 LE I LE 10 THEN DO;
     IF MPRVAR(I) > . THEN DENOM(I)=1;
     IF I IN (4,5,6) THEN DO;
      IF MPRVAR(I)IN (1,2) THEN NUMER(I)=1;
      ELSE NUMER(I)=0;
     END;
     ELSE IF I=7 THEN DO;
      IF MPRVAR(I) IN (4,5) THEN NUMER(I)=1;
      ELSE NUMER(I)=0;
     END;
     ELSE IF I IN (8,9,10) THEN DO;
      IF MPRVAR(I) IN (1,2,3,4) THEN NUMER(I)=1;
      ELSE NUMER(I)=0;
     END;
    END;
   END;

 DROP I;

*************************************************************************;

* taken from cahps, step 1 program ;

   IF (ZAGE NE . AND ZAGE NE 255) THEN DO;
      AGEUND6 = 0;
      AGE0612 = 0;
      AGE1317 = 0;

      IF      (ZAGE < 6)         THEN AGEUND6 = 1;
      ELSE IF (6  <= ZAGE <= 12) THEN AGE0612 = 1;
      ELSE IF (13 <= ZAGE <= 17) THEN AGE1317 = 1;
   END;
```

```
* create the beneficiary group/enrollment group subsets;

   IF XINS_COV = 1 THEN DO;
    BGROUP = 1; OUTPUT;
   END;

   IF XENR_PCM = 1   THEN DO;
    BGROUP = 2; OUTPUT;
   END;

   IF XENR_PCM = 2        THEN DO;
    BGROUP = 3; OUTPUT;
   END;

   IF XINS_COV IN (2,3) THEN DO;
    BGROUP = 4; OUTPUT;
   END;

   IF AGEUND6  = 1        THEN DO;
    BGROUP = 5; OUTPUT;
   END;

   IF AGE0612  = 1        THEN DO;
    BGROUP = 6; OUTPUT;
   END;

   IF AGE1317  = 1        THEN DO;
    BGROUP = 7; OUTPUT;
   END;

   BGROUP=8;
   OUTPUT;

RUN;

*************************************************************;

/**** Next, check catchment areas for requisite number of observations */
/**** for the macro calls (exclude cacsmpl w/ <2 obs)                  */
/**** also, keep list of region/catchment area combinations            */


 PROC FREQ NOPRINT DATA=&YRDATA;
  TABLE BGROUP*SUPREGA*CACSMPL/MISSING LIST
  OUT=OBSCNT(DROP=PERCENT);
 RUN;

 PROC SORT DATA=&YRDATA;
  BY BGROUP SUPREGA CACSMPL;
 RUN;

 DATA HCSDB /*FAILED*/; /** Note:  include all catchment areas **/
  MERGE &YRDATA(IN=IN_ALL) OBSCNT(IN=IN_OBS);
  BY BGROUP SUPREGA CACSMPL;

  IF COUNT < 2 THEN PUT "Less than 2 obs: SUPREGA=" SUPREGA "CACSMPL=" CACSMPL;
  ** OUTPUT FAILED;
  **END;
  **ELSE DO;
   OUTPUT HCSDB;
  **END;
 RUN;

 DATA OBSCNT;
  SET OBSCNT;
  RENAME BGROUP=GROUP;
 RUN;

 PROC SORT NODUPKEY DATA=OBSCNT;
  BY GROUP CACSMPL;
 RUN;
```

```
*************************************************************;

/*** Note:  for child MPR Composites, no significance tests are performed  ***/
/*** so standard errors are not calcuated                                  ***/
/*** A_Sudaan, Getcor, and Getsig macros are not needed(see adult program  ***/

/*******************************************************/
/***  Macro to derive composites for each         *******/
/***  beneficiary group, level                    *******/
/***  output one dataset for each group           *******/
/*******************************************************/

%MACRO GETPROP(BYVAR);

 %LET COMP1 = %EVAL(&CMPNUM1+1);
 %LET COMP2 = %EVAL(&CMPNUM1+&CMPNUM2);
 %LET COMP3 = %EVAL(&COMP2+1);

 %IF %UPCASE(&BYVAR)=SUPREGA %THEN %LET PREF=R;
 %ELSE %IF %UPCASE(&BYVAR)=CONUS %THEN %LET PREF=M;


  PROC MEANS NWAY NOPRINT DATA=HCSDB;
   CLASS BGROUP &BYVAR;
   VAR NUM&YR.V1-NUM&YR.V&COMPNUM
       DEN&YR.V1-DEN&YR.V&COMPNUM;
   WEIGHT &WGT;
   OUTPUT OUT= &PREF.CMPSUM(DROP = _TYPE_)
   SUM =  ;
  RUN;

  PROC MEANS NWAY NOPRINT DATA=HCSDB;
   CLASS BGROUP &BYVAR;
   VAR DEN&YR.V1-DEN&YR.V&COMPNUM;
   OUTPUT OUT=&PREF.DGFR(DROP=_TYPE_ _FREQ_)
   SUM= OBS&YR.V1-OBS&YR.V&COMPNUM;
  RUN;

  DATA ALLCAT;
   MERGE &PREF.CMPSUM(RENAME=(_FREQ_=N_OBS&YR.))
         &PREF.DGFR;
   BY BGROUP &BYVAR;
  RUN;


  PROC MEANS NWAY NOPRINT DATA=ALLCAT; /** summarize for all children **/
    CLASS BGROUP;
    VAR OBS&YR.V1-OBS&YR.V&COMPNUM
        NUM&YR.V1-NUM&YR.V&COMPNUM
        DEN&YR.V1-DEN&YR.V&COMPNUM;
    OUTPUT OUT=ALLCHILD(DROP=_TYPE_ _FREQ_)
    SUM= ;
  RUN;


  DATA OUT.MPRCOMPC;
    SET ALLCAT
       ALLCHILD;

    IF SUPREGA = . THEN SUPREGA = 0;  /** all children dataset **/;

    MAJGRP=PUT(SUPREGA, MAJGRPF.);

  /**** set up group variable  **/

   RENAME BGROUP=GROUP;

  /**** set up proportions, and composites **/

   ARRAY PROPORT PRP&YR.V1-PRP&YR.V&COMPNUM;
   ARRAY NUMER   NUM&YR.V1-NUM&YR.V&COMPNUM;
   ARRAY DENOM   DEN&YR.V1-DEN&YR.V&COMPNUM;
```

```
    DO J=1 TO DIM(PROPORT);
     PROPORT(J) = NUMER(J)/DENOM(J);
    END;
   DROP J;

  /**** 3 composites **/

   %DO Q=1 %TO &COMPCNT;
    %IF &Q=1 %THEN %DO;
     CP&YR.NUM&Q.=SUM(OF NUM&YR.V1-NUM&YR.V&CMPNUM1);
     CP&YR.DEN&Q.=SUM(OF DEN&YR.V1-DEN&YR.V&CMPNUM1);
     CP&YR.OBS&Q.=SUM(OF OBS&YR.V1-OBS&YR.V&CMPNUM1);
    %END;
    %ELSE %IF &Q=2 %THEN %DO;
     CP&YR.NUM&Q.=SUM(OF NUM&YR.V&COMP2.-NUM&YR.V&COMP2);
     CP&YR.DEN&Q.=SUM(OF DEN&YR.V&COMP2.-DEN&YR.V&COMP2);
     CP&YR.OBS&Q.=SUM(OF OBS&YR.V&COMP2.-OBS&YR.V&COMP2);
    %END;
    %ELSE %DO;
     CP&YR.NUM&Q.=SUM(OF NUM&YR.V&COMP3.-NUM&YR.V&COMPNUM);
     CP&YR.DEN&Q.=SUM(OF DEN&YR.V&COMP3.-DEN&YR.V&COMPNUM);
     CP&YR.OBS&Q.=SUM(OF OBS&YR.V&COMP3.-OBS&YR.V&COMPNUM);
    %END;
   COMP&YR.&Q.=CP&YR.NUM&Q./CP&YR.DEN&Q.;
  %END;

/** Note:  don't need weights for benchmarks **/

 %IF &DEBUG=Y AND &PREF=R %THEN %DO;

   PROC PRINT DATA=OUT.MPRCOMPC;  /* print out final dataset */
   RUN;                           /* for region to check     */
 %END;

%MEND GETPROP;

%GETPROP(SUPREGA);

ENDSAS;


*******************************************************************;
* Note : since no benchmarks are calculated or significance tests
* are performed, the additional macros from the adult program
* are not needed, and the program is terminated here
*******************************************************************;



/*******************************************************/
/*** First, calculate standard errors and create      ***/
/*** a file for each analytical unit                  ***/
/*******************************************************/

 PROC SORT DATA=HCSDB;
  BY STRATUM;
 RUN;


/*******************************************************/
/***** Sudaan macro to calculate standard errors   *****/
/***** there are three output datasets created     *****/
/***** (REGION, CACSMPL, MHS)                      *****/
/***** Note:  7/10/2000 use CONUS for MHS          *****/
/***** Note:  there are 8 variables and 8 groups   *****/
/*******************************************************/

%MACRO A_SUDAAN(TABLEVAR);

/*** set the number of levels in the proc descript ***/
/*** for region or catchment                       ***/
```

```
 %IF %UPCASE(&TABLEVAR)=XREGION %THEN %DO;
  %LET ENDNUM=&REGNUM;
  %LET PREF=R;         /** dataset prefix for region data    **/
 %END;
 %ELSE %IF %UPCASE(&TABLEVAR)=CACSMPL %THEN %DO;
  %LET ENDNUM=&CATCHNUM;
  %LET PREF=C;         /** dataset prefix for catchement area data **/
 %END;
 %ELSE %IF %UPCASE(&TABLEVAR)=CONUS %THEN %LET PREF=M;

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

 %DO I=1 %TO &GRPNUM;     /** 8 groups    **/

  %DO J=1 %TO &COMPNUM;   /** 8 variables **/

   DATA INDATA&I.&J(KEEP=&WGT MHS CONUS XREGION CACSMPL
                          NUM&YR.V&J DEN&YR.V&J STRATUM);
    SET HCSDB;
     WHERE XREGION > 0 AND BGROUP=&I
     AND  DEN&YR.V&J > 0;
     %IF %UPCASE(&TABLEVAR)=CONUS %THEN %DO;
      IF CONUS NE 1 THEN DELETE;  /* don't include all MHS */
     %END;
   RUN;

/**** Calculate values for regions, catchment areas ****/

   %IF %UPCASE(&TABLEVAR) NE CONUS %THEN %DO;

    PROC DESCRIPT DATA=INDATA&I.&J DESIGN=STRWR NOPRINT;
     WEIGHT &WGT;
     SETENV DECWIDTH=4;
     NEST STRATUM / MISSUNIT;
      VAR NUM&YR.V&J;
     TABLES &TABLEVAR;
     SUBGROUP &TABLEVAR;
     LEVELS &ENDNUM;
     OUTPUT SEMEAN/ TABLECELL=DEFAULT
     FILENAME=&PREF.GRP&I.V&J;
    RUN;

   %END;
   %ELSE %IF %UPCASE(&TABLEVAR)=CONUS %THEN %DO;

/**** No tables, levels, or subgroups needed ****/

    PROC DESCRIPT DATA=INDATA&I.&J DESIGN=STRWR NOPRINT;
     WEIGHT &WGT;
     SETENV DECWIDTH=4;
     NEST STRATUM / MISSUNIT;
      VAR NUM&YR.V&J;
     OUTPUT SEMEAN/ TABLECELL=DEFAULT
     FILENAME=&PREF.GRP&I.V&J;
    RUN;

   %END;

/***** first, put all variables into one dataset for each group *****/

    DATA &PREF.GRP&I.V&J;
     SET &PREF.GRP&I.V&J;
      IF SEMEAN NE .;
      MHS=1;
      %IF %UPCASE(&TABLEVAR)=CONUS %THEN %DO;
        CONUS=1;
      %END;
    RUN;

   %IF &J=1 %THEN %DO;

    DATA &PREF.SEGRP&I;
     SET &PREF.GRP&I.V&J(KEEP=&TABLEVAR SEMEAN);
```

```
      GROUP=&I;
      IF SEMEAN NE .;
      RENAME SEMEAN = SERR&YR.V&J;
    RUN;
   %END;
   %ELSE %DO;

    DATA &PREF.SEGRP&I;
     MERGE &PREF.SEGRP&I &PREF.GRP&I.V&J(KEEP=&TABLEVAR SEMEAN);
      BY &TABLEVAR;
      GROUP=&I;
      RENAME SEMEAN = SERR&YR.V&J;
    RUN;
   %END;
  %END;

/***** Put all data into one dataset  *****/
/***** Note:  changed output dataset  *****/
/***** to include group               *****/

  %IF &I=1 %THEN %DO;

   DATA OUT.&PREF.SERR;
    SET &PREF.SEGRP&I;
     KEEP GROUP &TABLEVAR SERR&YR.V1-SERR&YR.V&COMPNUM;
   RUN;
  %END;
  %ELSE %DO;

   DATA OUT.&PREF.SERR;
    SET OUT.&PREF.SERR
        &PREF.SEGRP&I;
   RUN;
  %END;

/******** DEBUG PRINT   *******/

   %IF &DEBUG=Y %THEN %DO;
    %IF &I=&GRPNUM AND &PREF=R %THEN %DO;
     PROC PRINT DATA=OUT.&PREF.SERR;
      VAR &TABLEVAR GROUP SERR&YR.V1-SERR&YR.V&COMPNUM;
     RUN;
    %END;
   %END;

 %END;

%MEND A_SUDAAN;

%A_SUDAAN (CONUS);      /** 7/10/2000 changed MHS to CONUS **/
%A_SUDAAN (XREGION);
%A_SUDAAN (CACSMPL);

*************************************************************;

/*******************************************************/
/*** Next, calculate correlation coefficients         ***/
/*** and create a file for each analytical unit       ***/
/*******************************************************/


%MACRO GETCORR(BYVAR);

 %IF %UPCASE(&BYVAR)=XREGION %THEN %LET PREF=R;
 %ELSE %IF %UPCASE(&BYVAR)=CACSMPL %THEN %LET PREF=C;
 %ELSE %IF %UPCASE(&BYVAR)=CONUS %THEN %LET PREF=M;

  PROC SORT DATA=HCSDB;
   BY &BYVAR;
  RUN;

 %DO I = 1 %TO &GRPNUM;
```

```
  PROC CORR NOPRINT DATA=HCSDB OUTP=&PREF.CORRC&I;
   %IF %UPCASE(&BYVAR)=CONUS %THEN %DO;
    WHERE BGROUP=&I AND CONUS = 1;
   %END;
   %ELSE %DO;
    WHERE BGROUP=&I;
   %END;
    BY &BYVAR;
      VAR PRVVAR1-PRVVAR&COMPNUM;
     WITH PRVVAR1-PRVVAR&COMPNUM;
     WEIGHT &WGT;
  RUN;

  DATA &PREF.CORRC&I;
   SET &PREF.CORRC&I;
    WHERE _TYPE_="CORR";
    GROUP=&I;
    ARRAY OLD PRVVAR1-PRVVAR&COMPNUM;
    ARRAY NEW COR&YR.V1-COR&YR.V&COMPNUM;
     DO J = 1 TO &COMPNUM;
      NEW(J)=OLD(J);
     END;
    DROP J PRVVAR1-PRVVAR&COMPNUM;
  RUN;

  %IF &I=1 %THEN %DO;

   DATA &PREF.CORRC;
    SET &PREF.CORRC&I;
   RUN;

  %END;
  %ELSE %DO;

   DATA &PREF.CORRC;
    SET &PREF.CORRC
        &PREF.CORRC&I;
   RUN;

  %END;

  %IF &DEBUG=Y %THEN %DO;
   %IF &I=&COMPNUM AND &PREF=R %THEN %DO;
    PROC PRINT DATA=&PREF.CORRC;
     WHERE GROUP=1;
    RUN;
   %END;
  %END;
 %END;

/*** Flatten dataset(for each region, condense matrix to one row) ***/

 %DO K=1 %TO &COMPNUM;

   DATA &PREF.CORR&K;
    SET &PREF.CORRC;
     WHERE _NAME_ = "PRVVAR&K";
     ARRAY CORR   (&COMPNUM) COR&YR.V1-COR&YR.V&COMPNUM;
     ARRAY CORR&K (&COMPNUM) COR&YR.V&K.1-COR&YR.V&K.&COMPNUM;
      DO L=1 TO &COMPNUM;
       CORR&K(L)=CORR(L);
      END;
     KEEP GROUP &BYVAR COR&YR.V&K.1-COR&YR.V&K.&COMPNUM;
   RUN;

  %IF &K=1 %THEN %DO;

   DATA OUT.&PREF.CORR;
    SET &PREF.CORR&K;
   RUN;

  %END;
  %ELSE %DO;
```

```
   DATA OUT.&PREF.CORR;
    MERGE OUT.&PREF.CORR(IN=IN_1) &PREF.CORR&K(IN=IN_2);
     BY GROUP &BYVAR;
   RUN;

  %END;

  %IF &DEBUG=Y %THEN %DO;
   %IF &PREF=R %THEN %DO;
    PROC PRINT DATA=OUT.&PREF.CORR;
          WHERE GROUP=1;
    RUN;
   %END;
  %END;

 %END;
%MEND GETCORR;

%GETCORR(CONUS);      /** changed to CONUS 7/10/2000 **/
%GETCORR(XREGION);
%GETCORR(CACSMPL);

*****************************************************************;



/*******************************************************/
/** since MHS benchmarks will be displayed           ****/
/** set up adjustment factor to apply to             ****/
/** each analytical unit's composite benchmarks      ****/
/*******************************************************/


DATA OUT.ADJUST;
 SET OUT.MCMPSUM(KEEP=GROUP CP&YR.BMK1 CP&YR.BMK2);
 RENAME CP&YR.BMK1=MHS&YR.BM1;
 RENAME CP&YR.BMK2=MHS&YR.BM2;
RUN;


/**** simple macro added to get adjusted composite scores ****/
/**** for region and catchment to be displayed in report  ****/
/**** cards                                                ****/

%MACRO ADJUST(FILE,LEVEL);

  DATA OUT.&FILE;
   MERGE OUT.&FILE (IN=IN_1) OUT.ADJUST;
   BY GROUP;
   %DO Z=1 %TO &COMPCNT;
     %IF &LEVEL=CONUS %THEN %DO;
      ADJ&YR.CP&Z.=1;
      C&YR.SCOR&Z=COMP&YR.&Z.*ADJ&YR.CP&Z.*100;
     %END;
     %ELSE %DO;
      ADJ&YR.CP&Z.=MHS&YR.BM&Z./CP&YR.BMK&Z.;
      C&YR.SCOR&Z=COMP&YR.&Z.*ADJ&YR.CP&Z.*100;
     %END;
   %END;
 RUN;

%MEND ADJUST;

%ADJUST(MCMPSUM,CONUS)
%ADJUST(RCMPSUM,REG);
%ADJUST(CCMPSUM,CAT);


*****************************************************************;

/*******************************************************/
/***  Macro to merge 3 datasets for each          *******/
```

```
/***  called by analytical unit                    *******/
/***  output final dataset for                     *******/
/***  Region, Catchment, MHS                       *******/
/*******************************************************/


%MACRO GETSIG(BYVAR);

 %LET START = %EVAL(&CMPNUM1+1);
 %LET NEXT  = %EVAL(&CMPNUM1+2);

 %IF &BYVAR=XREGION %THEN %LET PREF=R;
 %ELSE %IF &BYVAR=CACSMPL %THEN %LET PREF=C;
 %ELSE %IF &BYVAR=CONUS %THEN %LET PREF=M;

 DATA OUT.&PREF.FINAL(KEEP= MAJGRP REGION REGCAT GOALVAR1-GOALVAR&COMPNUM
                SIG&YR.V1-SIG&YR.V&COMPNUM SCOR&YR.V1-SCOR&YR.V&COMPNUM
                CP&YR.SIG1-CP&YR.SIG&COMPCNT CP&YR.1SE CP&YR.2SE
                C&YR.SCOR1-C&YR.SCOR&COMPCNT CP&YR.BMK1-CP&YR.BMK&COMPCNT
                SERR&YR.V1-SERR&YR.V&COMPNUM CP&YR.1SE CP&YR.2SE
                COMP&YR.1 COMP&YR.2 PROP&YR.V1-PROP&YR.V&COMPNUM
                DF&YR.SCR1-DF&YR.SCR&COMPNUM DF&YR._CP1 DF&YR._CP2
                NOBS&YR.V1-NOBS&YR.V&COMPNUM CP&YR.OBS1-CP&YR.OBS&COMPCNT
                DEN&YR.V1-DEN&YR.V&COMPNUM CP&YR.DEN1-CP&YR.DEN&COMPCNT)

  /** output a dataset to check **/

     OUT.&PREF.CHECK(DROP=DROP=SESQ&YR.V1-SESQ&YR.V&COMPNUM
                           PROP&YR.V1-PROP&YR.V&COMPNUM
                           SEM&YR.V11-SEM&YR.V&COMPNUM.&COMPNUM);

  FORMAT MAJGRP $30. REGION $15. REGCAT $26.;

  %IF &PREF=C %THEN %DO;

   MERGE OBSCNT(IN=IN_OBS) OUT.&PREF.CMPSUM(IN=IN_PROP) OUT.&PREF.CORR
         OUT.&PREF.SERR;
    BY GROUP &BYVAR;
    IF IN_OBS;

  %END;
  %ELSE %DO;

   MERGE OUT.&PREF.CMPSUM(IN=IN_PROP) OUT.&PREF.CORR
         OUT.&PREF.SERR;
    BY GROUP &BYVAR;
    IF IN_PROP;

  %END;

  /**  MAJGRP -- text field for group   **/

       IF GROUP=1 THEN MAJGRP="Prime Enrollees            ";
  ELSE IF GROUP=2 THEN MAJGRP="Enrollees with military PCM";
  ELSE IF GROUP=3 THEN MAJGRP="Enrollees with civilian PCM";
  ELSE IF GROUP=4 THEN MAJGRP="Non-enrolled beneficiaries ";
  ELSE IF GROUP=5 THEN MAJGRP="Active duty                ";
  ELSE IF GROUP=6 THEN MAJGRP="Active duty dependents     ";
  ELSE IF GROUP=7 THEN MAJGRP="Retirees and dependents    ";
  ELSE IF GROUP=8 THEN MAJGRP="All Beneficiaries          ";


  /**** REGION AND REGCAT SETUP          **/

  %IF &PREF=C %THEN %DO;
   REGCAT=PUT(CACSMPL, CAC.);
   REGION=PUT(XREGION, REGIONF.);
  %END;
  %ELSE %IF &PREF=R %THEN %DO;
   REGION=PUT(XREGION, REGIONF.);
   REGCAT=PUT(XREGION, REGIONF.);
  %END;
  %ELSE %DO;
```

```
 REGION="Conus MHS";
 REGCAT="Conus MHS";
%END;

 /**** setup t statistics, degreees of freedom    **/

  ARRAY    TSTAT{&COMPNUM} T_&YR.V1-T_&YR.V&COMPNUM;
  ARRAY    BMARK{&COMPNUM} GOALVAR1-GOALVAR&COMPNUM;
  ARRAY  STNDERR{&COMPNUM} SERR&YR.V1-SERR&YR.V&COMPNUM;
  ARRAY  SERRSQR{&COMPNUM} SESQ&YR.V1-SESQ&YR.V&COMPNUM;
  ARRAY     DEGF{&COMPNUM} DF&YR.SCR1-DF&YR.SCR&COMPNUM;
  ARRAY    DENOM{&COMPNUM} DEN&YR.V1-DEN&YR.V&COMPNUM;
  ARRAY  PROPORT{&COMPNUM} PROP&YR.V1-PROP&YR.V&COMPNUM;
  ARRAY    SCORE{&COMPNUM} SCOR&YR.V1-SCOR&YR.V&COMPNUM;
  ARRAY   PVALUE{&COMPNUM} PVAL&YR.V1-PVAL&YR.V&COMPNUM;
  ARRAY      SIG{&COMPNUM} SIG&YR.V1-SIG&YR.V&COMPNUM;
  ARRAY    N_OBS{&COMPNUM} NOBS&YR.V1-NOBS&YR.V&COMPNUM;

 /** get the item variance, t-statistics, df, p-values **/
 /** and whether significant                           **/

   DO I=1 TO &COMPNUM;
    SERRSQR{I}=STNDERR{I}**2;  /* Item variance */
    SCORE{I}=PROPORT{I}*100;   /* Score (prop. * 100) */
    IF STNDERR{I} > 0 THEN TSTAT{I}=(PROPORT{I}-BMARK{I})/STNDERR{I};
    ELSE TSTAT{I}=.;
    DEGF{I}=N_OBS{I}-1;
    PVALUE{I}=(1-PROBT(ABS(TSTAT{I}),DEGF{I}))*2;
    IF PVALUE{I} GE .05 THEN SIG{I}=0;
    ELSE IF PVALUE{I} < .05 THEN DO;
     IF PROPORT{I} > BMARK{I} THEN SIG{I}=1;
     IF PROPORT{I} < BMARK{I} THEN SIG{I}=-1;
    END;
   END;
  DROP I;

 /** multiply each item pair std. errors and correlation coefficients **/
 /** preventive care composite                                        **/

  ARRAY SERRC1{&CMPNUM1} SERR&YR.V1-SERR&YR.V&CMPNUM1;

   %DO J = 1 %TO &CMPNUM1;
    ARRAY  SMEAN&J{&CMPNUM1} SEM&YR.V&J.1-SEM&YR.V&J.&CMPNUM1;
    ARRAY CORVAR&J{&CMPNUM1} COR&YR.V&J.1-COR&YR.V&J.&CMPNUM1;
     DO K=1 TO &CMPNUM1;
      SMEAN&J{K}=SERR&YR.V&J*SERRC1{K}*CORVAR&J{K};
     END;
    SEM&YR.V&J.&J=0; /** don't count in final standard error calculation **/

   %END;
 DROP K;
 /** multiply each item pair std. errors and correlation coefficients **/
 /** access to care composite                                         **/

  ARRAY SERRC2{&CMPNUM2} SERR&YR.V&START-SERR&YR.V&COMPNUM;

   %DO L = &START %TO &COMPNUM;
    ARRAY  SMEAN&L{&CMPNUM2} SEM&YR.V&L.&START-SEM&YR.V&L.&COMPNUM;
    ARRAY CORVAR&L{&CMPNUM2} COR&YR.V&L.&START-COR&YR.V&L.&COMPNUM;
     DO M=1 TO &CMPNUM2;
      SMEAN&L{M}=SERR&YR.V&L*SERRC2{M}*CORVAR&L{M};
     END;
    SEM&YR.V&L.&L=0;  /** don't coun't in final standard error calculation **/
   %END;
  DROP M;
/** calculate composite t-statistic, pvalue, and whether significant **/
/** for composites                                                   **/

  %DO P=1 %TO &COMPCNT;
   %IF &P=1 %THEN %DO;

  /** composite standard error comprised of two parts **/
    CP&YR.&P.SE1=SUM(OF SESQ&YR.V1-SESQ&YR.V&CMPNUM1);
```

```
      CP&YR.&P.SE2=SUM(OF SEM&YR.V11-SEM&YR.V&CMPNUM1.&CMPNUM1.);

     %END;
     %ELSE %DO;
      CP&YR.&P.SE1=SUM(OF SESQ&YR.V&START-SESQ&YR.V&COMPNUM);
      CP&YR.&P.SE2=SUM(OF SEM&YR.V&START.&START.-SEM&YR.V&COMPNUM.&COMPNUM.);

     %END;

    /** add the two parts of the composite standard error  **/
    /** calculate the composite t statistics and p-values  **/
    /** determine whether differences are sigificant       **/

     CP&YR.&P.SE=SQRT(CP&YR.&P.SE2+CP&YR.&P.SE1)/&&CMPNUM&P.;
     IF CP&YR.&P.SE > 0 THEN CP&YR._T&P.=(COMP&YR.&P.-CP&YR.BMK&P.)/CP&YR.&P.SE;
     ELSE CP&YR._T&P.= .;
     DF&YR._CP&P.=CP&YR.OBS&P. - 1;
     CP&YR._P&P.=(1-PROBT(ABS(CP&YR._T&P.),DF&YR._CP&P.))*2;
     IF CP&YR._P&P GE .05 THEN CP&YR.SIG&P=0;
     ELSE IF CP&YR._P&P < .05 THEN DO;
      IF COMP&YR.&P. > CP&YR.BMK&P THEN CP&YR.SIG&P= 1;
      ELSE IF COMP&YR.&P. < CP&YR.BMK&P THEN CP&YR.SIG&P=-1;
     END;

    %END;

 OUTPUT OUT.&PREF.FINAL;

 %IF &PREF=M %THEN %DO;
  OUTPUT OUT.&PREF.CHECK;
 %END;

RUN;

%MEND GETSIG;

%GETSIG(CONUS);    /** changed to CONUS 7/10/2000 **/
%GETSIG(XREGION);
%GETSIG(CACSMPL);

ENDSAS;

*************************************************************;
```

7. CONVERT HEALTH STATUS COMPOSITE SCORES INTO WEB LAYOUT

```
/*****************************************************************
/*  Project:   DoD Reporting and Analysis 8687-100               *
/*  Program:   LOADMPR.SAS                                       *
/*  Author:    Chris Rankin                                      *
/*  Date:      08/01/2000                                        *
/*  Modified:                                                    *
/*  Purpose:   Load MPR composites for child data                *
/*                                                               *
/*   Input:    RFINAL.SD2                                        *
/*             MFINAL.SD2                                        *
/*   Output:   Loadmprc.sd2                                      *
/*****************************************************************/


OPTIONS COMPRESS=YES NOCENTER LS=124 PS=74 SOURCE SOURCE2;

LIBNAME   INLIB    ".";
LIBNAME   OUT      ".";
LIBNAME   LIBRARY  "J:\8687\DATA1999\CFINAL\FMTLIB";
LIBNAME   IN2      "J:\8687\PROGRAMS\LOADWEB";

%LET COMPNUM=10; /*** number of questions in all composites  ***/
%LET COMPCNT=3; /*** number of composites                   ***/


%LET YR=99;
%LET YEAR=1999;

%INCLUDE "J:\8687\PROGRAMS\LOADWEB\LOADCAHC.INC";

PROC FORMAT;
 VALUE BENF
  1-3, 11 = 'Primary Care Manager              '
  4-7, 12 = 'Attitudes Toward TRICARE Prime    '
  8-10,13 = 'Waiting times                     ';

RUN;


 DATA OUT.LOADMPRC(KEEP=MAJGRP ROWCAT BENEFIT BENTYPE SCORE N_OBS N_WGT);
   LENGTH MAJGRP  $42;
   LENGTH ROWCAT  $30;
   LENGTH BENTYPE $75;
   LENGTH BENEFIT $50;

  SET INLIB.MPRCOMPC;

  ARRAY PROPORT{*} PRP&YR.V1-PRP&YR.V&COMPNUM. COMP&YR.1-COMP&YR.&COMPCNT.;
  ARRAY    NOBS{*} OBS&YR.V1-OBS&YR.V&COMPNUM. CP&YR.OBS1-CP&YR.OBS&COMPCNT;
  ARRAY    NWGT{*} DEN&YR.V1-DEN&YR.V&COMPNUM. CP&YR.DEN1-CP&YR.DEN&COMPCNT;

   DO I=1 TO 13;
    SCORE=PROPORT{I}*100;
   ** SIG=SIGNIF{I};
    N_OBS=NOBS{I};
    N_WGT=NWGT{I};
    ROWCAT=PUT(GROUP,ROWCATF.);
    BENEFIT=PUT(I,BENF.);
    IF I=1 THEN DO;
     BENTYPE="Had a TRICARE PCM";
     OUTPUT;
    END;
    ELSE IF I=2 THEN DO;
     BENTYPE="Parent Knows Child's PCM's Name";
     OUTPUT;
    END;
    ELSE IF I=3 THEN DO;
     BENTYPE= "Had a 'Big Problem' Getting to See PCM";
     OUTPUT;
    END;
    ELSE IF I=4 THEN DO;
     BENTYPE= "Makes it Hard to Get Needed Health Services";
```

```
     OUTPUT;
    END;
    ELSE IF I=5 THEN DO;
     BENTYPE="Makes it Hard to See the Doctor the Parent Prefers";
     OUTPUT;
    END;
    ELSE IF I=6 THEN DO;
     BENTYPE="Health Benefits Do Not Meet Child's Needs";
     OUTPUT;
    END;
    ELSE IF I=7 THEN DO;
     BENTYPE="Provides High-Quality Health Care";
     OUTPUT;
    END;
    ELSE IF I=8 THEN DO;
     BENTYPE= "Regular or Routine Care Within 7 Days";
     OUTPUT;
    END;
    ELSE IF I=9 THEN DO;
     BENTYPE="Illness or Injury (Urgent) Care on Same Day";
     OUTPUT;
    END;
    ELSE IF I=10 THEN DO;
     BENTYPE="Well-Child Care in Less Than 4 Weeks";
     OUTPUT;
    END;
    ELSE IF I=11 THEN DO;
     BENTYPE= "&YEAR";
     OUTPUT;
    END;
    ELSE IF I=12 THEN DO;
     BENTYPE="&YEAR";
     OUTPUT;
    END;
    ELSE IF I=13 THEN DO;
     BENTYPE="&YEAR";
     OUTPUT;
    END;
   END;
 RUN;


PROC FREQ  DATA=OUT.LOADMPRC;
  WHERE BENEFIT IN(/*'Attitudes Toward TRICARE Prime    ',*/
                  'Waiting times                      ',
                  /*'Primary Care Manager             '*/);
 TABLES BENEFIT*BENTYPE*ROWCAT*MAJGRP/MISSING LIST
 OUT=CHECK1;
 RUN;


PROC FREQ NOPRINT DATA=IN2.FAKEC ;
 WHERE BENEFIT IN(/*'Attitudes Toward TRICARE Prime    ',*/
                  'Waiting times                      ',
                  /*'Primary Care Manager             '*/);
 TABLES BENEFIT*BENTYPE*ROWCAT*MAJGRP/MISSING LIST
 OUT=CHECK2;
 RUN;


DATA MISSING;
 MERGE CHECK1(IN=IN_1) CHECK2(IN=IN_2);
 BY BENEFIT BENTYPE ROWCAT MAJGRP;
 IF IN_2 & NOT IN_1 THEN OUTPUT MISSING;
RUN;

PROC PRINT DATA=MISSING;
RUN;
```

8. GENERATE WEB LAYOUT FOR 1999 CHILD TRICARE CONSUMER REPORTS

```
/*****************************************************************/
/* PROJECT:  8687 - 1999 Annual Survey                           */
/* PROGRAM:  FAKEC.SAS                                           */
/* PURPOSE:  Generate Fake Data for Report Cards                 */
/* AUTHOR:   Natalie Justh                                       */
/*****************************************************************/

LIBNAME  OUT  'C:\8687\WEB';
OPTIONS COMPRESS=YES;

PROC FORMAT;
  VALUE ROWMAT
    1  = 'CONUS MHS                 '
    2  = 'Under Age 6               '
    3  = 'Under Age 6-Benchmark      '
    4  = '6-12 Years                '
    5  = '6-12 Years-Benchmark       '
    6  = '13-17 Years               '
    7  = 'Prime Enrollees           '
    8  = 'Enrollees with military PCM'
    9  = 'Enrollees with civilian PCM'
   10  = 'Non-enrolled beneficiaries ';

  VALUE BEN
  1 = 'Getting Needed Care               '
  2 = 'Getting Care Quickly              '
  3 = 'How Well Doctors Communicate      '
  4 = 'Courteous and Helpful Office Staff'
  5 = 'Customer Service                  '
  6 = 'Personal Doctor or Nurse          '
  7 = 'Health Care                       '
  8 = 'Health Plan                       '
  9 = 'Primary Care Manager              '
 10 = 'Attitudes Toward TRICARE Prime    '
 11 = 'Waiting times                     ';

  VALUE MAJOR
  1 = 'All Children'
  2 = 'Children in New Regions (1, 2, & 5)'
  3 = 'Children in Mature Regions (6, 9-12, & 16)'
  4 = 'Children in Other Regions (3, 4, & 7/8)';

RUN;




DATA FAKEC;

  KEEP MAJGRP ROWCAT BENEFIT BENTYPE I K;

  LENGTH MAJGRP $  42
         ROWCAT $  30
         BENTYPE $ 75;



  DO I=1 TO 4;             ** 4 Major groups **;

     MAJGRP=PUT(I,MAJOR.);

     DO J=1 TO 10;         ** Age/Enrollment **;

     ROWCAT=PUT(J,ROWMAT.);


        DO K=1 TO 11;      ** 11 Benefits **;

        BENEFIT=PUT(K,BEN.);

          IF K=1 THEN DO;
              BENTYPE="Problems Getting Personal Doctor/Nurse"; OUTPUT;
```

```
            BENTYPE="Problems Getting Referral to Specialist"; OUTPUT;
            BENTYPE="Problems Getting Necessary Care"; OUTPUT;
            BENTYPE="Delays in Care While Awaiting Approval"; OUTPUT;
            BENTYPE="1999"; OUTPUT;
        END;
        ELSE IF K=2 THEN DO;
            BENTYPE="Advice over Telephone"; OUTPUT;
            BENTYPE="Urgent Care as soon as Parent Wanted"; OUTPUT;
            BENTYPE="Routine Care as soon as Parent Wanted"; OUTPUT;
            BENTYPE="Wait More Than 30 Minutes Past Appointment"; OUTPUT;
            BENTYPE="1999"; OUTPUT;
        END;
        ELSE IF K=3 THEN DO;
            BENTYPE="Listens Carefully"; OUTPUT;
            BENTYPE="Explains so you can Understand"; OUTPUT;
            BENTYPE="Explains so your child can Understand"; OUTPUT;
            BENTYPE="Shows Respect"; OUTPUT;
            BENTYPE="Spends Time with your child"; OUTPUT;
            BENTYPE="1999"; OUTPUT;
        END;
        ELSE IF K=4 THEN DO;
            BENTYPE="Courteous and Respectful"; OUTPUT;
            BENTYPE="Helpful"; OUTPUT;
            BENTYPE="1999"; OUTPUT;
        END;
        ELSE IF K=5 THEN DO;
            BENTYPE="Problem Getting Help from Customer Service"; OUTPUT;
            BENTYPE="Problem Finding/Understanding Written Material"; OUTPUT;
            BENTYPE="Problem with Paperwork"; OUTPUT;
            BENTYPE="1999"; OUTPUT;
        END;
        ELSE IF K=9 THEN DO;
            BENTYPE="Had a TRICARE PCM"; OUTPUT;
            BENTYPE="Parent Knows Child's PCM's Name"; OUTPUT;
            BENTYPE="Had a 'Big Problem' Getting to See PCM"; OUTPUT;
            BENTYPE="1999"; OUTPUT;
        END;
        ELSE IF K=10 THEN DO;
            BENTYPE="Makes it Hard to Get Needed Health Services"; OUTPUT;
            BENTYPE="Makes it Hard to See the Doctor the Parent Prefers"; OUTPUT;
            BENTYPE="Health Benefits Do Not Meet Child's Needs"; OUTPUT;
            BENTYPE="Provides High-Quality Health Care"; OUTPUT;
            BENTYPE="1999"; OUTPUT;
        END;
        ELSE IF K=11 THEN DO;
            BENTYPE="Regular or Routine Care Within 7 Days"; OUTPUT;
            BENTYPE="Illness or Injury (Urgent) Care on Same Day"; OUTPUT;
            BENTYPE="Well-Child Care in Less Than 4 Weeks"; OUTPUT;
            BENTYPE="1999"; OUTPUT;
        END;
        ELSE IF K IN (6,7,8) THEN DO;
            BENTYPE="1999"; OUTPUT;
        END;
      END;
    END;
  END;
RUN;


/*** Assign score and sig values ***/
DATA FAKEC;
  SET FAKEC;

  /*** significance is either -1, 0, or 1 ***/
  SIG = INT(RANUNI(1)*3)-1;



  /*** These BENEFITs have range 0-100 ***/
  IF BENEFIT='Getting Needed Care' THEN SCORE = INT(RANUNI(1)*100);
  IF BENEFIT='Getting Care Quickly' THEN SCORE = INT(RANUNI(1)*100);
  IF BENEFIT='How Well Doctors Communicate' THEN SCORE = INT(RANUNI(1)*100);
  IF BENEFIT='Courteous and Helpful Office Staff' THEN SCORE = INT(RANUNI(1)*100);
```

```
  IF BENEFIT='Customer Service' THEN SCORE = INT(RANUNI(1)*100);
  IF BENEFIT='Primary Care Manager' THEN SCORE = INT(RANUNI(1)*100);
  IF BENEFIT='Attitudes Toward TRICARE Prime' THEN SCORE = INT(RANUNI(1)*100);
  IF BENEFIT='Waiting times' THEN SCORE = INT(RANUNI(1)*100);
  IF BENEFIT='Personal Doctor or Nurse' THEN SCORE = INT(RANUNI(1)*100);
  IF BENEFIT='Health Care' THEN SCORE = INT(RANUNI(1)*100);
  IF BENEFIT='Health Plan' THEN SCORE = INT(RANUNI(1)*100);


  /*** Even though fake data, need to have benchmarks near high end of range ***/
  IF ROWCAT="Under Age 6-Benchmark" THEN DO;
      /*** These BENEFITs have range 0-100 ***/
      IF BENEFIT='Getting Needed Care' THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='Getting Care Quickly' THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='How Well Doctors Communicate' THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='Courteous and Helpful Office Staff'  THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='Customer Service' THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='Primary Care Manager' THEN DO;
         IF BENTYPE="Had a TRICARE PCM" THEN SCORE=.;
         ELSE IF BENTYPE="Parent Knows Child's PCM's Name" THEN SCORE=.;
         ELSE IF BENTYPE="1999" THEN SCORE=.;
         ELSE IF BENTYPE="Had a 'Big Problem' Getting to See PCM" THEN SCORE = (RANUNI(1)+90);
      END;
      IF BENEFIT='Attitudes Toward TRICARE Prime' THEN SCORE=.;
      IF BENEFIT='Waiting times' THEN SCORE=.;
      IF BENEFIT='Personal Doctor or Nurse' THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='Health Care' THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='Health Plan' THEN SCORE = (RANUNI(1)+90);

  END;

  IF ROWCAT="6-12 Years-Benchmark" THEN DO;
      /*** These BENEFITs have range 0-100 ***/
       IF BENEFIT='Getting Needed Care' THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='Getting Care Quickly' THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='How Well Doctors Communicate' THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='Courteous and Helpful Office Staff'  THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='Customer Service' THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='Primary Care Manager' THEN DO;
         IF BENTYPE="Had a TRICARE PCM" THEN SCORE=.;
         ELSE IF BENTYPE="Parent Knows Child's PCM's Name" THEN SCORE=.;
         ELSE IF BENTYPE="1999" THEN SCORE=.;
         ELSE IF BENTYPE="Had a 'Big Problem' Getting to See PCM" THEN SCORE = (RANUNI(1)+90);
      END;
      IF BENEFIT='Attitudes Toward TRICARE Prime' THEN SCORE=.;
      IF BENEFIT='Waiting times' THEN SCORE=.;
      IF BENEFIT='Personal Doctor or Nurse' THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='Health Care' THEN SCORE = (RANUNI(1)+90);
      IF BENEFIT='Health Plan' THEN SCORE = (RANUNI(1)+90);

  END;
RUN;


/*** Need to clean up data ***/
DATA OUT.FAKEC;
  SET FAKEC;

  /*** Need to set oddball records to missing ***/
  IF ROWCAT="Under Age 6-Benchmark" THEN SIG=.;
  ELSE IF ROWCAT="6-12 Years-Benchmark" THEN SIG=.;
  ELSE IF ROWCAT="CONUS MHS" THEN SIG=.;
  ELSE IF ROWCAT="13-17 Years" THEN SIG=.;
  ELSE IF ROWCAT="Prime Enrollees" THEN SIG=.;
  ELSE IF ROWCAT="Enrollees with military PCM" THEN SIG=.;
  ELSE IF ROWCAT="Enrollees with civilian PCM" THEN SIG=.;
  ELSE IF ROWCAT="Non-enrolled beneficiaries" THEN SIG=.;
  ELSE IF BENEFIT="Primary Care Manager" THEN DO;
     IF BENTYPE IN ("Had a TRICARE PCM","Parent Knows Child's PCM's Name","1999")
     THEN SIG=.;
  END;
  ELSE IF BENEFIT="Attitudes Toward TRICARE Prime" THEN SIG=.;
  ELSE IF BENEFIT="Waiting times" THEN SIG=.;
```

```
  DROP I K;

RUN;


PROC FREQ;
  TABLES MAJGRP ROWCAT BENTYPE BENEFIT SIG;
RUN;

PROC MEANS N MIN MAX MEAN;
  VAR SCORE;
RUN;
```

9. MERGE FINAL CAHPS AND HEALTH STATUS VARIABLES FOR CHILD TRICARE CONSUMER REPORTS

```
*****************************************************************************
*
* PROGRAM:  MERGFINC.SAS
* TASK:     1999 DOD HEALTH CARE SURVEY ANALYSIS (8687-200)
* PURPOSE:  Merge the final CAHPS and MPR Scores Databases
*           into the WEB layout preserving the order of the FAKEC.SD2.
*
* WRITTEN:  06/07/2000 BY KEITH RATHBUN
*
* INPUTS:   1) MPR and CAHPS Individual and Composite data sets with adjusted
*              scores, and benchmark data for 1998 and 1999 DoD HCS.
*              - LOADMPRC.SD2 - MPR Scores Database
*              - LOADCAHC.SD2 - CAHPS Scores Database
*              - BENCHC04.SD2 - 1999 CAHPS Benchmark Database
*              - FAKEC.SD2    - WEB Layout in Column order
*
* OUTPUT:   1) MERGFINC.SD2 - Combined Scores Database in WEB layout
*
* MODIFIED: 1) 07/24/2000 By Keith Rathbun - Adapted from MERGFINL.SAS to
*              reflect the requirements of the Child Report Card.
*
* NOTES:
*
* 1) The following steps need to be run prior to this program:
*  - STEP1C.SAS      - Recode questions and generate CAHPS group files
*  - STEP2C.SAS      - Calculate CAHPS individual adjusted scores for groups 1-7
*  - ALLSCORE.SAS    - Calculate CAHPS individual adjusted scores overall (group 8)
*  - COMPOSIT.SAS    - Calculate composite adjusted scores for group 1-8
*  - MPRCOMPC.SAS    - Calculate MPR individual and composite scores
*  - LOADMPRC.SAS    - Load MPR individual and composite scores into WEB layout
*  - BENCHC01-04.SAS - Convert 1999 Benchmark Scores into WEB layout
*  - LOADCAHC.SAS    - Convert 1999 CAHPS Scores Database into WEB layout
*
* 2) The output file (MERGFINC.SD2) will be run through the
*    MAKEHTMC.SAS program to generate the WEB pages.
*
****************************************************************************
* Assign data libraries and options
****************************************************************************;
LIBNAME IN1  ".";
LIBNAME IN2  "J:\8687\PROGRAMS\LOADWEB\CAHPS_CHILD99\DATA";
LIBNAME IN3  "J:\8687\PROGRAMS\REPORTCARDS\MPR_CHILD99";
LIBNAME IN4  "J:\8687\PROGRAMS\BENCHMARK\DATACHLD";
LIBNAME OUT  ".";
OPTIONS PS=79 LS=132 COMPRESS=YES NOCENTER;

****************************************************************************
* Construct ORDERing variable from WEB layout
****************************************************************************;
DATA ORDER;
   SET IN1.FAKEC;
   ORDER = _N_;
   LENGTH KEY $200;
   KEY = UPCASE(TRIM(BENEFIT)) ||
         UPCASE(TRIM(BENTYPE)) ||
         UPCASE(TRIM(MAJGRP))  ||
         UPCASE(TRIM(ROWCAT));
   KEEP KEY ORDER;
RUN;

PROC SORT DATA=ORDER; BY KEY; RUN;

****************************************************************************
* Merge the Scores Databases
****************************************************************************;
DATA MERGFINC;
   SET IN2.LOADCAHC(IN=INCAHP99)
       IN3.LOADMPRC(IN=INMPR99)
       IN4.BENCHC04(IN=INBEN99);
   SVCAHP99 = INCAHP99;
```

```
   SVMPR99  = INMPR99 ;
   SVBEN99  = INBEN99 ;
   LENGTH KEY     $200;
   KEY = UPCASE(TRIM(BENEFIT)) ||
         UPCASE(TRIM(BENTYPE)) ||
         UPCASE(TRIM(MAJGRP))  ||
         UPCASE(TRIM(ROWCAT));
   IF SCORE = . THEN DELETE;
RUN;

PROC SORT DATA=MERGFINC; BY KEY; RUN;

***************************************************************************
* Append ORDERing variable to the merged Scores database file
***************************************************************************;
DATA MERGFINC OUT.MISSING;
   MERGE MERGFINC(IN=IN1) ORDER(IN=IN2);
   BY KEY;

   LENGTH FLAG $30;
   IF IN1 AND IN2 THEN FLAG = "IN SCORES DB AND LAYOUT";
   ELSE IF    IN1 THEN FLAG = "IN SCORES DB ONLY";
   ELSE IF    IN2 THEN FLAG = "IN LAYOUT ONLY";

   LENGTH SOURCE $30;
   IF SVCAHP99 = 1 THEN SOURCE = "CAHPS 99    ";
   IF SVMPR99  = 1 THEN SOURCE = "MPR 99      ";
   IF SVBEN99  = 1 THEN SOURCE = "BENCHMARK 99";

   * IF FLAG = "IN SCORES DB ONLY" THEN DELETE;

   IF IN1 AND NOT IN2 THEN OUTPUT OUT.MISSING; *Missing from layout;
   IF IN1 THEN OUTPUT MERGFINC;
RUN;

***************************************************************************
* Reorder file according to WEB layout
***************************************************************************;
PROC SORT DATA=MERGFINC OUT=OUT.MERGFINC; BY ORDER; RUN;

DATA FAKEC;
   SET IN1.FAKEC;
   ORDER = _N_;
RUN;

DATA LAYONLY;
   MERGE FAKEC(IN=IN1) OUT.MERGFINC(IN=IN2 KEEP=ORDER);
   BY ORDER;
   IF IN1 AND NOT IN2;
RUN;

TITLE1 "1999 DOD Health Survey Scores/Report Cards (8687-200)";
TITLE2 "Program Name: MERGFINC.SAS By Keith Rathbun";
TITLE3 "Program Inputs: MPR and CAHPS Combined Scores data sets and WEB Layout";
TITLE4 "Program Outputs: MERGFINC.SD2 - Merged Final Scores Database for input to MAKEHTMC.SAS";

TITLE5 "MERGFINC.SD2 Data source counts";
PROC FREQ DATA=OUT.MERGFINC;
TABLES SOURCE FLAG SVCAHP99 SVMPR99 SVBEN99
                   SVCAHP99*SVMPR99*SVBEN99
      /MISSING LIST;
RUN;

TITLE5 "MERGFINC.SD2 Data attribute counts";
PROC FREQ DATA=OUT.MERGFINC;
TABLES BENEFIT BENTYPE MAJGRP ROWCAT
      /MISSING LIST;
RUN;

TITLE5 "LAYONLY.SD2 Data attribute counts";
PROC FREQ DATA=LAYONLY;
TABLES BENEFIT BENTYPE MAJGRP ROWCAT
      /MISSING LIST;
```

```
RUN;

TITLE5 "No matching record found in LAYOUT file (FAKEC.SD2)";
PROC FREQ DATA=OUT.MISSING;
TABLES MAJGRP ROWCAT BENTYPE BENEFIT
       MAJGRP*ROWCAT*BENTYPE*BENEFIT
      /MISSING LIST;
RUN;

TITLE5 "No matching record found in LAYOUT file (FAKEC.SD2)";
PROC PRINT DATA=OUT.MISSING;
VAR MAJGRP ROWCAT BENTYPE BENEFIT;
RUN;
_
```

10. CALCULATE SIGNIFICANCE TEST FOR CAHPS SCORES

```
***************************************************************************
*
* PROGRAM:  SIGNIF_C.SAS
* TASK:     1999 DOD HEALTH CARE SURVEY ANALYSIS (8687-200)
* PURPOSE:  Perform significance tests for CAHPS scores
*
* WRITTEN:  07/27/2000 BY KEITH RATHBUN
*
* INPUTS:   1) MERGFINC.SD2 - MPR and CAHPS Scores Database in WEB layout
*           2) FAKEC.SD2 - Scores Database WEB Layout
*
* OUTPUT:   1) SIGNIF_C.SD2 - Combined Scores Database in WEB layout
*
* NOTES:
*
* 1) This is the 2nd to last program to be run in the scores DB series.  All
*    report card records must be merged (MERGFINC.SAS) prior to the
*    significance tests.
*
* 2) The output file (SIGNIF_C.SD2) will be run through the
*    CONUS_C.SAS program to generate the CONUS scores.
*
***************************************************************************
* Assign data libraries and options
***************************************************************************;
LIBNAME IN1  ".";
LIBNAME OUT  ".";
OPTIONS PS=79 LS=132 COMPRESS=YES NOCENTER ERRORS=1;

***************************************************************************
* Append 1999 BENCHMARK records to perform significance tests
***************************************************************************;
DATA BENCH1(KEEP=T MAJGRP ROWCAT BENEFIT BENTYPE BSCORE BSEMEAN);
   SET IN1.MERGFINC;
   WHERE ROWCAT IN("6-12 Years-Benchmark","Under Age 6-Benchmark")
                AND SVMPR99 = 0;
   LENGTH T $10;
   IF ROWCAT = "6-12 Years-Benchmark" THEN T = "6-12";
   ELSE T = "UNDER 6";
   RENAME SCORE  = BSCORE;
   RENAME SEMEAN = BSEMEAN;
RUN;
PROC SORT DATA=BENCH1; BY T MAJGRP BENEFIT BENTYPE; RUN;

DATA BENCH2(KEEP=N_WGT N_OBS T MAJGRP BENEFIT BENTYPE SCORE SEMEAN ORDER);
   SET IN1.MERGFINC;
   WHERE ROWCAT IN("6-12 Years","Under Age 6")
                AND SVMPR99 = 0;
   LENGTH T $10;
   IF ROWCAT = "6-12 Years" THEN T = "6-12";
   ELSE T = "UNDER 6";
RUN;
PROC SORT DATA=BENCH2; BY T MAJGRP BENEFIT BENTYPE; RUN;

***************************************************************************
* Extract scores to perform significance tests
***************************************************************************;
DATA BENCHMRK(KEEP=ORDER SIG);
   MERGE BENCH1(IN=IN1) BENCH2(IN=IN2);
   BY T MAJGRP BENEFIT BENTYPE;
   TEMP = (SCORE-BSCORE)/SQRT(BSEMEAN**2+SEMEAN**2);
   TEST = 2*(1-PROBT(ABS(TEMP),N_OBS-1));
   SIG = 0;
   IF N_OBS >= 30 AND TEST < 0.05 THEN SIG = 1;
   IF SCORE < BSCORE THEN SIG = -SIG;
   IF IN1 AND IN2;
RUN;
PROC SORT DATA=BENCHMRK; BY ORDER; RUN;

***************************************************************************
* Extract MPR records
***************************************************************************;
```

```
DATA MPR;
   SET IN1.MERGFINC;
   BY ORDER;
   WHERE SVMPR99 = 1;
RUN;
PROC SORT DATA=MPR; BY ORDER; RUN;

DATA BENCHMRK;
   MERGE IN1.MERGFINC(IN=IN1) BENCHMRK(IN=IN2);
   BY ORDER;
   IF IN2;
RUN;

DATA MERGFINC;
   SET IN1.MERGFINC;
   BY ORDER;
   WHERE ROWCAT NOT IN("6-12 Years","Under Age 6")
                AND SVMPR99 = 0;
RUN;

***************************************************************************
* Combine newly created CAHPS SIG values with MPR records
***************************************************************************;
DATA ALLSCORE;
   SET MPR BENCHMRK MERGFINC;
   BY ORDER;
RUN;

***************************************************************************
* Store missing values into SIG and SCORE for FAKE.SD2 records
***************************************************************************;
DATA FAKEC;
   SET IN1.FAKEC;
   SIG = .;
   SCORE = .;
   ORDER = _N_;
RUN;

DATA FAKEONLY;
   MERGE ALLSCORE(IN=IN1) FAKEC(IN=IN2);
   BY ORDER;
   SOURCE = "FAKE ONLY";
   FLAG   = "FAKE ONLY";
   IF IN2 AND NOT IN1;
RUN;

DATA OUT.SIGNIF_C;
   SET FAKEONLY ALLSCORE;
   BY ORDER;
RUN;

TITLE1 "1999 DOD Health Survey Scores/Report Cards (8687-200)";
TITLE2 "Program Name: SIGNIF_C.SAS By Keith Rathbun";
TITLE3 "Program Inputs: MPR and CAHPS data records in WEB Layout";
TITLE4 "Program Outputs: SIGNIF_C.SD2 - Merged Final Scores Database for input to MAKEHTMC.SAS";

TITLE5 "FREQs of SIGNIF_C.SD2";
PROC FREQ;
   TABLES SIG SOURCE FLAG MAJGRP ROWCAT BENEFIT BENTYPE
  /MISSING LIST;
RUN;

TITLE5 "FREQs of FAKEC.SD2";
PROC FREQ DATA=IN1.FAKEC;
   TABLES MAJGRP ROWCAT BENEFIT BENTYPE
  /MISSING LIST;
RUN;
_
```

11. CALCULATE SCORES AND SIGNIFICANCE TEST FOR CAHPS CONUS SCORES

```
***************************************************************************
*
* PROGRAM:  CONUS_C.SAS
* TASK:     1999 DOD HEALTH CARE SURVEY ANALYSIS (8687-200)
* PURPOSE:  Generate CAHPS CONUS scores and significance tests
*
* WRITTEN:  07/27/2000 BY KEITH RATHBUN
*
* INPUTS:   1) SIGNIF_C.SD2 - Scores Database in WEB Layout
*           2) MERGFINC.SD2 - Merged Scores Databases in WEB Layout
*           3) FAKEC.SD2 - Scores Database WEB Layout
*
* OUTPUT:   1) CONUS_C.SD2 - Combined CAHPS/MPR Scores Database in WEB layout
*           2) LT30C.SD2 - Records with <= 30 observations
*
* NOTES:
*
* 1) The following steps need to be run prior to this program:
*    - STEP1C.SAS    - Recode questions and generate group files
*    - STEP2C.SAS    - Calculate individual adjusted scores for group 1-7
*    - ALLSCORE.SAS - Calculate individual adjusted scores overall (group 8)
*    - COMPOSIT.SAS - Calculate composite adjusted scores for group 1-8
*    - MERGFINC.SAS - Merge the final CAHPS and MPR Scores Databases
*    - SIGNIF_C.SAS - Perform significance tests for CAHPS scores
*
***************************************************************************
* Assign data libraries and options
***************************************************************************;
LIBNAME IN1  "J:\8687\PROGRAMS\LOADWEB";
LIBNAME OUT  ".";
OPTIONS PS=79 LS=132 COMPRESS=YES NOCENTER;

***************************************************************************
***************************************************************************
*
* Process Macro Input Parameters:
*
* 1) BENTYPE = Benefit Type
* 2) BENEFIT = COMPOSITE Benefit Type
* 3) ROWCAT  = Row Category
*
***************************************************************************
***************************************************************************;
* Set up empty template file for data merge purposes and set first time flag
***************************************************************************;
%LET DSN = SIGNIF_C;
DATA INIT;
   SET IN1.&DSN;
   DELETE;
RUN;
%LET FLAG = 0;

%MACRO PROCESS(BENTYPE=,BENEFIT=,ROWCAT=);
DATA TEMP;
   SET IN1.&DSN END=FINISHED;
   WHERE BENTYPE = "&BENTYPE" AND
         ROWCAT  = "&ROWCAT"  AND
         BENEFIT = "&BENEFIT" AND
         MAJGRP NE "All Children";
RUN;

DATA TEMP;
   SET TEMP END=FINISHED;
   IF _N_ = 1 THEN DO;
      SUMSCOR1 = 0;    RETAIN SUMSCOR1;
      SUMWGT1 = 0;     RETAIN SUMWGT1;
      SUMSE2 = 0;      RETAIN SUMSE2;
      SUMWGT2 = 0;     RETAIN SUMWGT2;
      N_OBS1 = 0;      RETAIN N_OBS1;
   END;
   ************************************************************************
   * Note: For the Child Survey only CONUS were sent surveys
```

```
   ************************************************************************;
   IF SCORE  NE . AND N_WGT NE . THEN SUMSCOR1 = SUMSCOR1 + (SCORE*N_WGT);
   IF N_WGT  NE . THEN SUMWGT1  = SUMWGT1  + N_WGT;
   IF SEMEAN NE . AND N_WGT NE . THEN SUMSE2 = SUMSE2 + (SEMEAN*N_WGT)**2;
   IF N_OBS  NE . THEN N_OBS1 + N_OBS;
   IF FINISHED THEN GOTO FINISHED;
   RETURN;

KEEP MAJGRP ROWCAT BENTYPE BENEFIT SIG SCORE SEMEAN N_OBS N_WGT
     FLAG SOURCE SUMSCOR1 SUMWGT1 SUMSE2 KEY;

FINISHED:
   SCORE   = SUMSCOR1/SUMWGT1;
   SEMEAN  = SQRT(SUMSE2)/SUMWGT1;
   N_OBS   = N_OBS1;
   N_WGT   = SUMWGT1;
   SOURCE  = "CONUS";
   FLAG    = "CONUS";
   MAJGRP  = "All Children";
   KEY = UPCASE(TRIM(BENEFIT)) ||
         UPCASE(TRIM(BENTYPE)) ||
         UPCASE(TRIM(MAJGRP))  ||
         UPCASE(TRIM(ROWCAT));
   OUTPUT;
RUN;

%IF &FLAG = 0 %THEN %DO;
   DATA FINAL;
      SET TEMP;
   RUN;
%END;
%ELSE %DO;
   DATA FINAL;
      SET FINAL TEMP;
   RUN;
%END;
%LET FLAG = 1;

%MEND;

*************************************************************************
* Process 1999 CONUS Composites
*************************************************************************
*************************************************************************
* 1) Create CONUS for Getting Needed Care
*************************************************************************;
%PROCESS(BENTYPE=1999,
         ROWCAT=Under Age 6                , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=6-12 Years                 , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=13-17 Years                , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with military PCM, BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=Prime Enrollees            , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=CONUS MHS                  , BENEFIT=Getting Needed Care);

%PROCESS(BENTYPE=Problems Getting Personal Doctor/Nurse,
         ROWCAT=Under Age 6                , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Personal Doctor/Nurse,
         ROWCAT=6-12 Years                 , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Personal Doctor/Nurse,
         ROWCAT=13-17 Years                , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Personal Doctor/Nurse,
         ROWCAT=Enrollees with military PCM, BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Personal Doctor/Nurse,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Getting Needed Care);
```

```
%PROCESS(BENTYPE=Problems Getting Personal Doctor/Nurse,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Personal Doctor/Nurse,
         ROWCAT=Prime Enrollees            , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Personal Doctor/Nurse,
         ROWCAT=CONUS MHS                  , BENEFIT=Getting Needed Care);

%PROCESS(BENTYPE=Problems Getting Referral to Specialist,
         ROWCAT=Under Age 6                , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Referral to Specialist,
         ROWCAT=6-12 Years                 , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Referral to Specialist,
         ROWCAT=13-17 Years                , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Referral to Specialist,
         ROWCAT=Enrollees with military PCM, BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Referral to Specialist,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Referral to Specialist,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Referral to Specialist,
         ROWCAT=Prime Enrollees            , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Referral to Specialist,
         ROWCAT=CONUS MHS                  , BENEFIT=Getting Needed Care);

%PROCESS(BENTYPE=Problems Getting Necessary Care,
         ROWCAT=Under Age 6                , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Necessary Care,
         ROWCAT=6-12 Years                 , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Necessary Care,
         ROWCAT=13-17 Years                , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Necessary Care,
         ROWCAT=Enrollees with military PCM, BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Necessary Care,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Necessary Care,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Necessary Care,
         ROWCAT=Prime Enrollees            , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Problems Getting Necessary Care,
         ROWCAT=CONUS MHS                  , BENEFIT=Getting Needed Care);

%PROCESS(BENTYPE=Delays in Care While Awaiting Approval,
         ROWCAT=Under Age 6                , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Delays in Care While Awaiting Approval,
         ROWCAT=6-12 Years                 , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Delays in Care While Awaiting Approval,
         ROWCAT=13-17 Years                , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Delays in Care While Awaiting Approval,
         ROWCAT=Enrollees with military PCM, BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Delays in Care While Awaiting Approval,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Delays in Care While Awaiting Approval,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Delays in Care While Awaiting Approval,
         ROWCAT=Prime Enrollees            , BENEFIT=Getting Needed Care);
%PROCESS(BENTYPE=Delays in Care While Awaiting Approval,
         ROWCAT=CONUS MHS                  , BENEFIT=Getting Needed Care);

*************************************************************************
* 2) Create CONUS for Getting Care Quickly
*************************************************************************;
%PROCESS(BENTYPE=1999,
         ROWCAT=Under Age 6                , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=1999,
         ROWCAT=6-12 Years                 , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=1999,
         ROWCAT=13-17 Years                , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with military PCM, BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=1999,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Getting Care Quickly);
```

```
%PROCESS(BENTYPE=1999,
         ROWCAT=Prime Enrollees            , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=1999,
         ROWCAT=CONUS MHS                  , BENEFIT=Getting Care Quickly);

%PROCESS(BENTYPE=Advice over Telephone,
         ROWCAT=Under Age 6                , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Advice over Telephone,
         ROWCAT=6-12 Years                 , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Advice over Telephone,
         ROWCAT=13-17 Years                , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Advice over Telephone,
         ROWCAT=Enrollees with military PCM, BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Advice over Telephone,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Advice over Telephone,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Advice over Telephone,
         ROWCAT=Prime Enrollees            , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Advice over Telephone,
         ROWCAT=CONUS MHS                  , BENEFIT=Getting Care Quickly);

%PROCESS(BENTYPE=Urgent Care as soon as Parent Wanted,
         ROWCAT=Under Age 6                , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Urgent Care as soon as Parent Wanted,
         ROWCAT=6-12 Years                 , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Urgent Care as soon as Parent Wanted,
         ROWCAT=13-17 Years                , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Urgent Care as soon as Parent Wanted,
         ROWCAT=Enrollees with military PCM, BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Urgent Care as soon as Parent Wanted,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Urgent Care as soon as Parent Wanted,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Urgent Care as soon as Parent Wanted,
         ROWCAT=Prime Enrollees            , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Urgent Care as soon as Parent Wanted,
         ROWCAT=CONUS MHS                  , BENEFIT=Getting Care Quickly);

%PROCESS(BENTYPE=Routine Care as soon as Parent Wanted,
         ROWCAT=Under Age 6                , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Routine Care as soon as Parent Wanted,
         ROWCAT=6-12 Years                 , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Routine Care as soon as Parent Wanted,
         ROWCAT=13-17 Years                , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Routine Care as soon as Parent Wanted,
         ROWCAT=Enrollees with military PCM, BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Routine Care as soon as Parent Wanted,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Routine Care as soon as Parent Wanted,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Routine Care as soon as Parent Wanted,
         ROWCAT=Prime Enrollees            , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Routine Care as soon as Parent Wanted,
         ROWCAT=CONUS MHS                  , BENEFIT=Getting Care Quickly);

%PROCESS(BENTYPE=Wait More Than 30 Minutes Past Appointment,
         ROWCAT=Under Age 6                , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Wait More Than 30 Minutes Past Appointment,
         ROWCAT=6-12 Years                 , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Wait More Than 30 Minutes Past Appointment,
         ROWCAT=13-17 Years                , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Wait More Than 30 Minutes Past Appointment,
         ROWCAT=Enrollees with military PCM, BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Wait More Than 30 Minutes Past Appointment,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Wait More Than 30 Minutes Past Appointment,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Wait More Than 30 Minutes Past Appointment,
         ROWCAT=Prime Enrollees            , BENEFIT=Getting Care Quickly);
%PROCESS(BENTYPE=Wait More Than 30 Minutes Past Appointment,
         ROWCAT=CONUS MHS                  , BENEFIT=Getting Care Quickly);
```

```
*************************************************************************
* 3) Create CONUS for How Well Doctors communicate
*************************************************************************;
%PROCESS(BENTYPE=1999,
         ROWCAT=Under Age 6                , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=1999,
         ROWCAT=6-12 Years                 , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=1999,
         ROWCAT=13-17 Years                , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with military PCM, BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=1999,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=1999,
         ROWCAT=Prime Enrollees            , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=1999,
         ROWCAT=CONUS MHS                  , BENEFIT=How Well Doctors Communicate);

%PROCESS(BENTYPE=Listens Carefully,
         ROWCAT=Under Age 6                , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Listens Carefully,
         ROWCAT=6-12 Years                 , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Listens Carefully,
         ROWCAT=13-17 Years                , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Listens Carefully,
         ROWCAT=Enrollees with military PCM, BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Listens Carefully,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Listens Carefully,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Listens Carefully,
         ROWCAT=Prime Enrollees            , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Listens Carefully,
         ROWCAT=CONUS MHS                  , BENEFIT=How Well Doctors Communicate);

%PROCESS(BENTYPE=Explains so you can Understand,
         ROWCAT=Under Age 6                , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so you can Understand,
         ROWCAT=6-12 Years                 , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so you can Understand,
         ROWCAT=13-17 Years                , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so you can Understand,
         ROWCAT=Enrollees with military PCM, BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so you can Understand,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so you can Understand,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so you can Understand,
         ROWCAT=Prime Enrollees            , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so you can Understand,
         ROWCAT=CONUS MHS                  , BENEFIT=How Well Doctors Communicate);

%PROCESS(BENTYPE=Explains so your child can Understand,
         ROWCAT=Under Age 6                , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so your child can Understand,
         ROWCAT=6-12 Years                 , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so your child can Understand,
         ROWCAT=13-17 Years                , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so your child can Understand,
         ROWCAT=Enrollees with military PCM, BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so your child can Understand,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so your child can Understand,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so your child can Understand,
         ROWCAT=Prime Enrollees            , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Explains so your child can Understand,
         ROWCAT=CONUS MHS                  , BENEFIT=How Well Doctors Communicate);

%PROCESS(BENTYPE=Shows Respect,
         ROWCAT=Under Age 6                , BENEFIT=How Well Doctors Communicate);
```

```
%PROCESS(BENTYPE=Shows Respect,
         ROWCAT=6-12 Years                 , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Shows Respect,
         ROWCAT=13-17 Years                , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Shows Respect,
         ROWCAT=Enrollees with military PCM, BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Shows Respect,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Shows Respect,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Shows Respect,
         ROWCAT=Prime Enrollees            , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Shows Respect,
         ROWCAT=CONUS MHS                  , BENEFIT=How Well Doctors Communicate);

%PROCESS(BENTYPE=Spends Time with your child,
         ROWCAT=Under Age 6                , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Spends Time with your child,
         ROWCAT=6-12 Years                 , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Spends Time with your child,
         ROWCAT=13-17 Years                , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Spends Time with your child,
         ROWCAT=Enrollees with military PCM, BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Spends Time with your child,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Spends Time with your child,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Spends Time with your child,
         ROWCAT=Prime Enrollees            , BENEFIT=How Well Doctors Communicate);
%PROCESS(BENTYPE=Spends Time with your child,
         ROWCAT=CONUS MHS                  , BENEFIT=How Well Doctors Communicate);

*****************************************************************************
* 4) Create CONUS for Courteous and Helpful Office Staff
*****************************************************************************;
%PROCESS(BENTYPE=1999,
         ROWCAT=Under Age 6                , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=1999,
         ROWCAT=6-12 Years                 , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=1999,
         ROWCAT=13-17 Years                , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with military PCM, BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=1999,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=1999,
         ROWCAT=Prime Enrollees            , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=1999,
         ROWCAT=CONUS MHS                  , BENEFIT=Courteous and Helpful Office Staff);

%PROCESS(BENTYPE=Courteous and Respectful,
         ROWCAT=Under Age 6                , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=Courteous and Respectful,
         ROWCAT=6-12 Years                 , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=Courteous and Respectful,
         ROWCAT=13-17 Years                , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=Courteous and Respectful,
         ROWCAT=Enrollees with military PCM, BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=Courteous and Respectful,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=Courteous and Respectful,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=Courteous and Respectful,
         ROWCAT=Prime Enrollees            , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=Courteous and Respectful,
         ROWCAT=CONUS MHS                  , BENEFIT=Courteous and Helpful Office Staff);

%PROCESS(BENTYPE=Helpful,
         ROWCAT=Under Age 6                , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=Helpful,
         ROWCAT=6-12 Years                 , BENEFIT=Courteous and Helpful Office Staff);
```

```
%PROCESS(BENTYPE=Helpful,
         ROWCAT=13-17 Years                , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=Helpful,
         ROWCAT=Enrollees with military PCM, BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=Helpful,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=Helpful,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=Helpful,
         ROWCAT=Prime Enrollees            , BENEFIT=Courteous and Helpful Office Staff);
%PROCESS(BENTYPE=Helpful,
         ROWCAT=CONUS MHS                  , BENEFIT=Courteous and Helpful Office Staff);

***************************************************************************
* 5) Create CONUS for Customer Service
***************************************************************************;
%PROCESS(BENTYPE=1999,
         ROWCAT=Under Age 6                , BENEFIT=Customer Service);
%PROCESS(BENTYPE=1999,
         ROWCAT=6-12 Years                 , BENEFIT=Customer Service);
%PROCESS(BENTYPE=1999,
         ROWCAT=13-17 Years                , BENEFIT=Customer Service);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with military PCM, BENEFIT=Customer Service);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Customer Service);
%PROCESS(BENTYPE=1999,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Customer Service);
%PROCESS(BENTYPE=1999,
         ROWCAT=Prime Enrollees            , BENEFIT=Customer Service);
%PROCESS(BENTYPE=1999,
         ROWCAT=CONUS MHS                  , BENEFIT=Customer Service);

%PROCESS(BENTYPE=Problem Getting Help from Customer Service,
         ROWCAT=Under Age 6                , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Getting Help from Customer Service,
         ROWCAT=6-12 Years                 , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Getting Help from Customer Service,
         ROWCAT=13-17 Years                , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Getting Help from Customer Service,
         ROWCAT=Enrollees with military PCM, BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Getting Help from Customer Service,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Getting Help from Customer Service,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Getting Help from Customer Service,
         ROWCAT=Prime Enrollees            , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Getting Help from Customer Service,
         ROWCAT=CONUS MHS                  , BENEFIT=Customer Service);

%PROCESS(BENTYPE=Problem Finding/Understanding Written Material,
         ROWCAT=Under Age 6                , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Finding/Understanding Written Material,
         ROWCAT=6-12 Years                 , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Finding/Understanding Written Material,
         ROWCAT=13-17 Years                , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Finding/Understanding Written Material,
         ROWCAT=Enrollees with military PCM, BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Finding/Understanding Written Material,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Finding/Understanding Written Material,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Finding/Understanding Written Material,
         ROWCAT=Prime Enrollees            , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem Finding/Understanding Written Material,
         ROWCAT=CONUS MHS                  , BENEFIT=Customer Service);

%PROCESS(BENTYPE=Problem with Paperwork,
         ROWCAT=Under Age 6                , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem with Paperwork,
         ROWCAT=6-12 Years                 , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem with Paperwork,
         ROWCAT=13-17 Years                , BENEFIT=Customer Service);
```

```
%PROCESS(BENTYPE=Problem with Paperwork,
         ROWCAT=Enrollees with military PCM, BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem with Paperwork,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem with Paperwork,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem with Paperwork,
         ROWCAT=Prime Enrollees            , BENEFIT=Customer Service);
%PROCESS(BENTYPE=Problem with Paperwork,
         ROWCAT=CONUS MHS                  , BENEFIT=Customer Service);

***************************************************************************
* 6) Create CONUS for Health Care
***************************************************************************;
%PROCESS(BENTYPE=1999,
         ROWCAT=Under Age 6                , BENEFIT=Health Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=6-12 Years                 , BENEFIT=Health Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=13-17 Years                , BENEFIT=Health Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with military PCM, BENEFIT=Health Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Health Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Health Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=Prime Enrollees            , BENEFIT=Health Care);
%PROCESS(BENTYPE=1999,
         ROWCAT=CONUS MHS                  , BENEFIT=Health Care);

***************************************************************************
* 7) Create CONUS for Health Plan
***************************************************************************;
%PROCESS(BENTYPE=1999,
         ROWCAT=Under Age 6                , BENEFIT=Health Plan);
%PROCESS(BENTYPE=1999,
         ROWCAT=6-12 Years                 , BENEFIT=Health Plan);
%PROCESS(BENTYPE=1999,
         ROWCAT=13-17 Years                , BENEFIT=Health Plan);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with military PCM, BENEFIT=Health Plan);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Health Plan);
%PROCESS(BENTYPE=1999,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Health Plan);
%PROCESS(BENTYPE=1999,
         ROWCAT=Prime Enrollees            , BENEFIT=Health Plan);
%PROCESS(BENTYPE=1999,
         ROWCAT=CONUS MHS                  , BENEFIT=Health Plan);

***************************************************************************
* 8) Create CONUS for Personal Doctor or Nurse
***************************************************************************;
%PROCESS(BENTYPE=1999,
         ROWCAT=Under Age 6                , BENEFIT=Personal Doctor or Nurse);
%PROCESS(BENTYPE=1999,
         ROWCAT=6-12 Years                 , BENEFIT=Personal Doctor or Nurse);
%PROCESS(BENTYPE=1999,
         ROWCAT=13-17 Years                , BENEFIT=Personal Doctor or Nurse);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with military PCM, BENEFIT=Personal Doctor or Nurse);
%PROCESS(BENTYPE=1999,
         ROWCAT=Enrollees with civilian PCM, BENEFIT=Personal Doctor or Nurse);
%PROCESS(BENTYPE=1999,
         ROWCAT=Non-enrolled beneficiaries , BENEFIT=Personal Doctor or Nurse);
%PROCESS(BENTYPE=1999,
         ROWCAT=Prime Enrollees            , BENEFIT=Personal Doctor or Nurse);
%PROCESS(BENTYPE=1999,
         ROWCAT=CONUS MHS                  , BENEFIT=Personal Doctor or Nurse);

***************************************************************************
* Store missing values into SIG and SCORE for FAKEC.SD2 records
```

```
*************************************************************************;
DATA FAKEC;
   SET IN1.FAKEC;
   SIG = .;
   SCORE = .;
   ORDER = _N_;
   KEY = UPCASE(TRIM(BENEFIT)) ||
         UPCASE(TRIM(BENTYPE)) ||
         UPCASE(TRIM(MAJGRP))  ||
         UPCASE(TRIM(ROWCAT));
RUN;

PROC SORT DATA=FAKEC; BY KEY; RUN;
PROC SORT DATA=FAKEC  OUT=TEMP(KEEP=ORDER KEY); BY KEY; RUN;
PROC SORT DATA=FINAL; BY KEY; RUN;

DATA FINAL;
   MERGE FINAL(IN=IN1) TEMP(IN=IN2);
   BY KEY;
   IF IN1 AND IN2;
RUN;

*************************************************************************
* Append 1999 BENCHMARK records to perform significance tests
*************************************************************************;
DATA BENCH1(KEEP=T MAJGRP ROWCAT BENEFIT BENTYPE BSCORE BSEMEAN);
   SET IN1.MERGFINC;
   WHERE ROWCAT IN("6-12 Years-Benchmark","Under Age 6-Benchmark")
                AND SVMPR99 = 0;
   LENGTH T $10;
   IF ROWCAT = "6-12 Years-Benchmark" THEN T = "6-12";
   ELSE T = "UNDER 6";
   RENAME SCORE  = BSCORE;
   RENAME SEMEAN = BSEMEAN;
RUN;
PROC SORT DATA=BENCH1; BY T MAJGRP BENEFIT BENTYPE; RUN;

DATA BENCH2(KEEP=ORDER KEY N_OBS N_WGT T MAJGRP BENEFIT BENTYPE SCORE SEMEAN SOURCE FLAG);
   SET FINAL;
   LENGTH T $10;
   IF ROWCAT = "6-12 Years" THEN T = "6-12";
   ELSE T = "UNDER 6";
   IF ROWCAT IN("6-12 Years","Under Age 6") THEN OUTPUT BENCH2;
RUN;
PROC SORT DATA=BENCH2; BY T MAJGRP BENEFIT BENTYPE; RUN;

*************************************************************************
* Extract scores to perform significance tests
*************************************************************************;
DATA SIGTESTS(KEEP=ORDER SIG);
   MERGE BENCH1(IN=IN1) BENCH2(IN=IN2);
   BY T MAJGRP BENEFIT BENTYPE;
   TEMP = (SCORE-BSCORE)/SQRT(BSEMEAN**2+SEMEAN**2);
   TEST = 2*(1-PROBT(ABS(TEMP),N_OBS-1));
   SIG = 0;
   IF N_OBS >= 30 AND TEST < 0.05 THEN SIG = 1;
   IF SCORE < BSCORE THEN SIG = -SIG;
   IF IN1 AND IN2;
RUN;
PROC SORT DATA=SIGTESTS; BY ORDER; RUN;
PROC SORT DATA=FINAL;    BY ORDER; RUN;

DATA SIGTESTS;
   MERGE FINAL(IN=IN1) SIGTESTS(IN=IN2);
   BY ORDER;
   SOURCE = "CONUS_C";
   FLAG   = "CONUS_C";
   IF IN1;
   KEEP ORDER KEY MAJGRP ROWCAT BENEFIT BENTYPE N_OBS N_WGT
        FLAG SOURCE SCORE SEMEAN SIG;
RUN;
%LET DSN = SIGNIF_C;
*************************************************************************
```

```
* Get records from SIGNIF_C.SD2 that are NOT FAKE ONLY
*************************************************************************;
DATA &DSN;
   SET IN1.&DSN;
   IF FLAG NE "FAKE ONLY";
RUN;
PROC SORT DATA=&DSN; BY ORDER; RUN;

***************************************************************************
* Combine previously created records with the new file
***************************************************************************;
DATA COMBINE OUT.LT30C;
   SET &DSN SIGTESTS;
   BY ORDER;
   ************************************************************************
   * Remove N_OBS < 30 OR N_WGT < 200
   ************************************************************************;
   IF (N_OBS < 30 OR N_WGT < 200) AND
      (ROWCAT NOT IN("Under Age 6-Benchmark","6-12 Years-Benchmark"))
      THEN OUTPUT OUT.LT30C;
   ELSE OUTPUT COMBINE;
RUN;

PROC SORT DATA=FAKEC; BY ORDER; RUN;
DATA FAKEONLY;
   MERGE COMBINE(IN=IN1) FAKEC(IN=IN2);
   BY ORDER;
   SOURCE = "FAKE ONLY";
   FLAG   = "FAKE ONLY";
   IF IN2 AND NOT IN1;
RUN;

DATA OUT.CONUS_C;
   SET FAKEONLY COMBINE;
   BY ORDER;
   ************************************************************************
   * Convert CAHPS rating scores to 1-100 scale
   ************************************************************************;
   IF BENEFIT IN ("Health Care","Health Plan","Personal Doctor or Nurse") THEN
      SCORE = SCORE*10;
   ************************************************************************
   * Convert CAHPS Composites and Individual (not ratings) to 1-100 scale
   ************************************************************************;
   IF NOT (BENEFIT IN ("Health Care","Health Plan","Personal Doctor or Nurse",
           "Attitudes Toward TRICARE Prime","Primary Care Manager","Waiting times")) THEN
      SCORE = 50*(SCORE-1);
   ************************************************************************
   * One of the Getting Care Quickly Questions was different.  So, we need to
   * reset the significance test and score.
   ************************************************************************;
   if benefit = "Primary Care Manager" & rowcat = "Non-enrolled beneficiaries" then
  score = .P;
   IF BENEFIT = "Getting Care Quickly" & bentype in ('1999','Wait More Than 30 Minutes Past
Appointment')
    THEN DO;
      SIG = 0;
      IF ROWCAT IN("Under Age 6-Benchmark","6-12 Years-Benchmark") THEN SCORE = .A;
   END;
RUN;

TITLE1 "1999 DOD Health Survey Scores/Report Cards (8687-200)";
TITLE2 "Program Name: CONUS_C.SAS By Keith Rathbun";
TITLE3 "Program Inputs: SIGNIF_C.SD2 - Scores Database in WEB Layout";
TITLE4 "Program Outputs: CONUS_C.SD2 - CONUS Scores Database in WEB layout";

PROC FREQ;
TABLES FLAG SOURCE SIG BENEFIT BENTYPE MAJGRP ROWCAT
      /MISSING LIST;
RUN;
_
```

12. GENERATE HTML FILE FOR THE 1999 CHILD TRICARE CONSUMER REPORT

```
*=====================================================================;
*   Programmer:  Mark A. Brinkley (modified by Natalie Justh)         ;
*        Title:  MAKEHTMC.SAS                                          ;
*       Client:  8687-200                                              ;
*         Date:  05-03-2000                                            ;
*                                                                      ;
*      Purpose:  This program is to create 36 HTML                     ;
*                 report cards for the 1999 DOD project (CHILD)        ;
*                                                                      ;
*  Input files:  FAKEC.SD2 (fake data just for testing)                ;
*                                                                      ;
* Output files:  DATA\                                                 ;
*                                                                      ;
*=====================================================================;


%LET HTMLSP=%STR( );
%LET QUOTE=%STR("");

%LET fontface=%STR(Arial,Helvetica,Swiss,Geneva);
%LET hdcolr=%STR('white');
%LET BLUE=%STR('#663300');
%LET GREEN=%STR('#009933');
%LET RED=%STR('#cc0000');
%LET GRAY=%STR('white');

%LET LOGO=%STR('child_side.gif');
%LET HELP_BUTTON=%STR('help75.gif');
%LET HOME_BUTTON=%STR('home75.gif');
%LET BACK_BUTTON=%STR('back75.gif');


LIBNAME SRC1 'j:\8687\programs\loadweb'  ACCESS=READONLY;
OPTIONS LS=132;


*********************************************************************;
****   Macro to create html pages                                 ****;
****          var1=major group                                    ****;
****          var2=0                                              ****;
****          var3=benefit                                        ****;
*********************************************************************;
%MACRO MKHTML(var1,var2,var3);

/** Load in data **/
DATA SUBSET;
  SET SRC1.CONUS_C;

  LENGTH MAJGRPA $ 47;

  IF MAJGRP="Children in Mature Regions (6, 9-12, & 16)" THEN
     MAJGRPA="Children in Mature Regions (6, 9-12, & Alaska)";
     ELSE IF MAJGRP="All Children" THEN MAJGRPA="Children in CONUS MHS";
     ELSE MAJGRPA=MAJGRP;
  IF ROWCAT="CONUS MHS" THEN ROWCAT="All Children";

  /*** VAR1 indicated major group ***/
  %if &var1.=1 %then %let major=%STR(Children in CONUS MHS);
  %if &var1.=2 %then %let major=%STR(Children in New Regions (1, 2, & 5));
  %if &var1.=3 %then %let major=%STR(Children in Mature Regions (6, 9-12, & Alaska));
  %if &var1.=4 %then %let major=%STR(Children in Other Regions (3, 4, & 7/8));


  IF MAJGRPA="&major.";    /*** Subset data by major group ***/


  FILEOUT1=COMPRESS("htmc\c&var1.-&var2.-&var3..htm");
  call symput('fileout1',FILEOUT1);


  /*** VAR3 dictates type of benefit heading ***/
  %if &var3=0 %then %do;
```

```
      %let headvar=BENEFIT;
  %end;
  %else %let headvar=BENTYPE;
RUN;



/*** Subset data by Benefit ***/
DATA SUBSET;
  SET SUBSET;

  %if &var3.=0 %then %do;
     IF BENTYPE="1999";
     %let columns=11;
     %let colmns=12;
  %end;
  %else %if &var3.=1 %then %do;
     IF BENEFIT="Getting Needed Care";
     %let columns=4;
     %let colmns=5;
  %end;
  %else %if &var3.=2 %then %do;
     IF BENEFIT="Getting Care Quickly";
     %let columns=4;
     %let colmns=5;
  %end;
  %else %if &var3.=3 %then %do;
     IF BENEFIT="How Well Doctors Communicate";
     %let columns=5;
     %let colmns=6;
  %end;
  %else %if &var3.=4 %then %do;
     IF BENEFIT="Courteous and Helpful Office Staff";
     %let columns=2;
     %let colmns=3;
  %end;
  %else %if &var3.=5 %then %do;
     IF BENEFIT="Customer Service";
     %let columns=3;
     %let colmns=4;
  %end;
  %else %if &var3.=9 %then %do;
     IF BENEFIT="Personal Doctor or Nurse";
     %let columns=0;
     %let colmns=0;
  %end;
  %else %if &var3.=10 %then %do;
     IF BENEFIT="Health Care";
     %let columns=0;
     %let colmns=0;
  %end;
  %else %if &var3.=11 %then %do;
     IF BENEFIT="Health Plan";
     %let columns=0;
     %let colmns=0;
  %end;
  %else %if &var3.=6 %then %do;
     IF BENEFIT="Primary Care Manager";
     %let columns=3;
     %let colmns=4;
  %end;
  %else %if &var3.=7 %then %do;
     IF BENEFIT="Attitudes Toward TRICARE Prime";
     %let columns=4;
     %let colmns=5;
  %end;
  %else %if &var3.=8 %then %do;
     IF BENEFIT="Waiting times";
     %let columns=3;
     %let colmns=4;
  %end;
```

```
  *** Set macro variable ***;
  %if &var3.=0 %then %do;
      %let sub_ben=%STR(1999 Composite Scores);
  %end;
  %else %do;
    call symput('sub_ben',BENEFIT);
  %end;

RUN;



*************************************;
****  Put out 1st rows of table   ****;
*************************************;
DATA HTML;
  SET SUBSET;

  IF ROWCAT IN ("All Children");
  HREFTOP=COMPRESS("c&var1.-0-0.htm");

  /*** Create macro variable date with today's date ***/
  DATETIME=DATETIME();
  CALL SYMPUT ('DATETIME',left(put(datetime,datetime20.)));
  DROP DATETIME;
RUN;



/*** Initialize HTML page ***/
DATA _NULL_;
  FILE "&FILEOUT1.";

  PUT "<! Created &datetime.>";
  PUT "<html><head><title>";
  PUT "&major., &sub_ben.";
  PUT "</title></head>";
  PUT "<body bgcolor='#999966' text='#000099' link='#660066' alink='#660066' vlink='#996699'>";

RUN;



/*** If ALL benefits then do special column headers ***/
%if &var3.=0 %then %do;
DATA _NULL_;
  SET HTML END=EOF;

  FILE "&FILEOUT1." MOD;

  IF _N_=1 THEN DO;

         /** MF Changes ROW 1 **/
         PUT "<center><table border='1' cellpadding='2' cellspacing='0' bgcolor='#cccc99' cols=12
width='90%'>";
         PUT "<tr bgcolor='white'>";
         PUT  "      <td  colspan='6' valign='top'  bgcolor='#999966'><img  border='0'  height='25'
width='242' src=&logo.></td>";
         PUT "    <td colspan='6' align='right' valign='bottom' bgcolor='#999966'>";
         PUT "          <div align='right'>";
         PUT "          <a href='index.htm'><img src=&home_button. border='0' alt='Return to Main
Page'></a>&htmlsp. &htmlsp.";
         PUT  "               <a  href="""  HREFTOP   +(-1)  """><img  src=&back_button.  border='0'
alt='Return to Top Level'></a>&htmlsp. &htmlsp.";
         PUT   "                        <a   href='help.htm'><img   src=&help_button.   border='0'
alt='Help'></a></div>";
         PUT "    </td>";
         PUT "</tr>";

         /** MF Changes ROW 2 **/
         PUT "<tr>";
         PUT "        <td colspan='12' bgcolor='#cccc99'>";
```

```
         PUT "            <center>";
         PUT "                <h2><font face='&fontface.' color='#3333cc'>&major.<br>";
         PUT "                 &sub_ben.</font></h2>";
         PUT "            </center>";
         PUT "        </td>";
         PUT "</tr>";

         PUT "<tr bgcolor= &hdcolr.>";            /*** Print out 1st row ***/
         PUT "<td>&htmlsp.</td>";
         PUT    "<td    ALIGN=CENTER    colspan=2><font    face='Arial'    size=-1><b>Ease    of
Access</b></font></td>";
         PUT "<td ALIGN=CENTER colspan=3><font face='Arial' size=-1><b>Communication and Customer
Service</b></font></td>";
         PUT    "<td    ALIGN=CENTER    colspan=3><font    face='Arial'    size=-1><b>Parents'
Ratings</b></font></td>";
         PUT    "<td    ALIGN=CENTER    colspan=3><font    face='Arial'    size=-1><b>TRICARE
Prime</b></font></td>";
         PUT "</tr>";
         PUT "<tr bgcolor= &hdcolr.>";

         /*** Print out 1st column of 2nd row ***/
         %if &var2.=0 %then %do;
           PUT "<td width='7%'><font face='&fontface.'>&htmlsp.</font></td>";
         %end;

         bennum=1;
  END;


  /*** Put Benefits across columns ***/
  IF BENEFIT IN ("Getting Needed Care",
                 "Getting Care Quickly",
                 "How Well Doctors Communicate",
                 "Courteous and Helpful Office Staff",
                 "Customer Service",
                 "Primary Care Manager",
                 "Attitudes Toward TRICARE Prime",
                 "Waiting times") THEN DO;
     HREF=COMPRESS("c&var1.-&var2.-"||bennum||".htm");
     PUT  "<td  width='7%'  align='center'  valign='bottom'><font  face='&fontface.'size='1'><a
href="" HREF +(-1) ">" &HEADVAR. "</a></font></td>";
     bennum+1;
  END;
  ELSE DO;
     HREF=COMPRESS("help.htm#q&var3.");
     PUT  "<td  width='7%'  align='center'  valign='bottom'><font  face='&fontface.'size='1'><a
href="" HREF +(-1) ">" &HEADVAR. "</a></font></td>";
  END;


  IF EOF THEN DO;
    PUT "</tr>";

    /*** Add scale in 3rd row ***/
    PUT "<tr bgcolor= &hdcolr.>";
    PUT "<td align='center' valign='bottom'><font face='&fontface.'><b>Scale</b></font></td>";
    PUT   "<td   align='center'   valign='bottom'   colspan=11><font   face='&fontface.'><b>(0-
100)</b></font></td>";
    PUT="</tr>";

  END;

RUN;
%end;



/*** If Sub-benefit then do differently ***/
%else %do;
DATA _NULL_;
  SET HTML END=EOF;

  FILE "&FILEOUT1." MOD;
```

```
  COLUMNS=&colmns.;
  SPAN1=ROUND(COLUMNS/2,1);
  SPAN2=COLUMNS-SPAN1;

  IF _N_=1 THEN DO;
         *** put table title ***;
         **PUT     "<h2><center><font     face='&fontface.'>     &major.     <br>     &sub_ben.
</font></center></h2>";


         /** MF Changes ROW 1 **/
         PUT   "<center><table   border='1'   cellpadding='2'   cellspacing='0'   bgcolor='#cccc99'
width='85%'>";
         PUT "<tr bgcolor='white'>";
         PUT "    <td colspan=""" SPAN1 +(-1) """ valign='top' bgcolor='#999966'><img border='0'
height='25' width='242' src=&logo.></td>";
         PUT   "           <td   colspan="""   SPAN2   +(-1)   """   align='right'   valign='bottom'
bgcolor='#999966'>";
         PUT "          <div align='right'>";
         PUT "          <a href='index.htm'><img src=&home_button. border='0' alt='Return to Main
Page'></a>&htmlsp. &htmlsp.";
         PUT  "               <a  href="""  HREFTOP   +(-1)  """><img  src=&back_button.  border='0'
alt='Return to Top Level'></a>&htmlsp. &htmlsp.";
         PUT   "                        <a   href='help.htm'><img   src=&help_button.   border='0'
alt='Help'></a></div>";
         PUT "    </td>";
         PUT "</tr>";

         /** MF Changes ROW 2 **/
         PUT "<tr>";
         PUT "        <td colspan="""  COLUMNS +(-1) """ bgcolor='#cccc99'>";
         PUT "            <center>";
         PUT "                <h2><font face='&fontface.' color='#3333cc'>&major. <br>";
         PUT "                 &sub_ben.</font></h2>";
         PUT "            </center>";
         PUT "        </td>";
         PUT "</tr>";


         PUT "<tr bgcolor= &hdcolr.><font face='&fontface.'>";
         PUT "<td width='10%'>&htmlsp.</td>";
  END;


  /*** Print out column headings ***/
  HREF=COMPRESS("help.htm#q&var3");
  IF &HEADVAR^="1999" THEN DO;
     PUT  "<td  width='10%'  align='center'  valign='bottom'><font  face='&fontface.'  size='2'><a
href="""  HREF +(-1) """><b>" &HEADVAR. "</b></a></font></td>";
  END;

  IF EOF THEN DO;

    PUT "</font></tr>";

    /*** Add scale in 3rd row ***/
    PUT      "<tr      bgcolor=      &hdcolr.><td      align='center'      valign='bottom'><font
face='&fontface.'><b>Scale</b></font></td>";
    IF BENEFIT='Getting Needed Care' THEN PUT "<td align='center' valign='bottom' colspan=4><font
face='&fontface.'><b>(0-100)</b></font></td></tr>";
    ELSE  IF  BENEFIT='Getting  Care  Quickly'    THEN  PUT  "<td  align='center'  valign='bottom'
colspan=4><font face='&fontface.'><b>(0-100)</b></font></td></tr>";
    ELSE IF BENEFIT='How Well Doctors Communicate' THEN PUT "<td align='center' valign='bottom'
colspan=5><font face='&fontface.'><b>(0-100)</b></font></td></tr>";
    ELSE  IF  BENEFIT='Courteous  and  Helpful  Office  Staff'  THEN  PUT  "<td  align='center'
valign='bottom' colspan=2><font face='&fontface.'><b>(0-100)</b></font></td></tr>";
    ELSE   IF   BENEFIT='Customer   Service'   THEN   PUT   "<td   align='center'   valign='bottom'
colspan=3><font face='&fontface.'><b>(0-100)</b></font></td></tr>";
    ELSE  IF  BENEFIT='Primary  Care  Manager'  THEN  PUT  "<td  align='center'  valign='bottom'
colspan=3><font face='&fontface.'><b>(0-100)</b></font></td></tr>";
    ELSE IF BENEFIT='Attitudes Toward TRICARE Prime' THEN PUT "<td align='center' valign='bottom'
colspan=4><font face='&fontface.'><b>(0-100)</b></font></td></tr>";
```

```
    ELSE IF BENEFIT='Waiting times' THEN PUT "<td align='center' valign='bottom' colspan=3><font
face='&fontface.'><b>(0-100)</b></font></td></tr>";

  END;

RUN;
%end;



*************************************;
****  Put out rest of table      ****;
****  Colored scores and Stub    ****;
*************************************;
DATA HTML;
  SET SUBSET;
RUN;


***** All Regions ****;
%if &var2.=0 %then %do;
DATA HTML;
  SET HTML END=EOF;

  FILE "&FILEOUT1." MOD;

  LENGTH LROWCAT $40;
  RETAIN LROWCAT;

  IF _N_=1 THEN DO;
     LROWCAT=" ";
     ROWNUM=1;
     ROW=0;
  END;

  IF LROWCAT^=ROWCAT THEN DO;              *** Start new row ***;
  IF BENEFIT IN ("Attitudes Toward TRICARE Prime",
                 "Waiting times") THEN DO;
        ROW+1;
        IF LROWCAT^=" " THEN PUT "</tr>";  *** terminate previous row ***;


        IF ROWCAT="All Children" THEN PUT "<tr><td><b><font face='&fontface.' size='2'>" ROWCAT
"</font></b></td>";

        ELSE IF ROWCAT="Under Age 6" THEN DO;
           PUT   "<tr   bgcolor=   &gray.><td   width='90%'   ALIGN=LEFT   colspan=&COLMNS><b><font
face='Arial' size=-1>Age Group</font></b></td>";
           PUT "<tr><td><b><font face='Arial' size=-1>" ROWCAT "</font></b></td>";
        END;
        ELSE  IF  ROWCAT="6-12  Years"  THEN  PUT  "<tr><td><b><font  face='Arial'  size=-1>"  ROWCAT
"</font></b></td>";
        ELSE  IF  ROWCAT="13-17  Years"  THEN  PUT  "<tr><td><b><font  face='Arial'  size=-1>"  ROWCAT
"<sup>*</sup></font></b></td>";
        ELSE IF ROWCAT="Prime Enrollees" THEN DO;
           PUT   "<tr   bgcolor=   &gray.><td   width='90%'   ALIGN=LEFT   colspan=&COLMNS><b><font
face='Arial' size=-1>Enrollment Group</font></b></td>";
           PUT "<tr><td><b><font face='Arial' size=-1>" ROWCAT "</font></b></td>";
        END;
        ELSE IF ROWCAT IN ("Enrollees with military PCM",
                           "Enrollees with civilian PCM") THEN PUT "<tr><td><b><font face='Arial'
size=-1>&HTMLSP.&HTMLSP." ROWCAT "</font></b></td>";
        ELSE  IF  ROWCAT="Non-enrolled  beneficiaries"  THEN  PUT  "<tr><td><b><font  face='Arial'
size=-1>" ROWCAT "</font></b></td>";

      LROWCAT=ROWCAT;
  END;
  ELSE DO;
        ROW+1;
        IF LROWCAT^=" " THEN PUT "</tr>";  *** terminate previous row ***;
```

```
        IF ROWCAT="All Children" THEN PUT "<tr><td><b><font face='&fontface.' size='2'>" ROWCAT
"</font></b></td>";

        ELSE IF ROWCAT="Under Age 6" THEN DO;
           PUT   "<tr   bgcolor=   &gray.><td   width='90%'   ALIGN=LEFT   colspan=&COLMNS><b><font
face='&fontface.' size='2'>Age Group</font></b></td>";
           PUT "<tr><td><b><font face='&fontface.' size='2'>" ROWCAT "</font></b></td>";
        END;
        ELSE   IF   ROWCAT="Under   Age   6-Benchmark"   THEN   PUT   "<tr><td><b><font   color=&blue.
face='&fontface.' size='2'>&HTMLSP.&HTMLSP.Benchmark</font></b></td>";
        ELSE  IF  ROWCAT="6-12  Years"  THEN  PUT  "<tr><td><b><font  face='&fontface.'  size='2'>"
ROWCAT "</font></b></td>";
        ELSE   IF   ROWCAT="6-12   Years-Benchmark"   THEN   PUT   "<tr><td><b><font   color=&blue.
face='Arial' size=-1>&HTMLSP.&HTMLSP.Benchmark</font></b></td>";
        ELSE  IF  ROWCAT="13-17  Years"  THEN  PUT  "<tr><td><b><font  face='&fontface.'  size='2'>"
ROWCAT "<sup>*</sup></font></b></td>";
        ELSE IF ROWCAT="Prime Enrollees" THEN DO;
           PUT   "<tr   bgcolor=   &gray.><td   width='90%'   ALIGN=LEFT  colspan=&COLMNS><b><font
face='&fontface.' size='2'>Enrollment Group</font></b></td>";
           PUT "<tr><td><b><font face='&fontface.' size='2'>" ROWCAT "</font></b></td>";
        END;
        ELSE IF ROWCAT IN ("Enrollees with military PCM",
                           "Enrollees   with   civilian   PCM")   THEN   PUT   "<tr><td><b><font
face='&fontface.' size='2'>&HTMLSP.&HTMLSP." ROWCAT "</font></b></td>";
         ELSE IF ROWCAT="Non-enrolled beneficiaries" THEN PUT "<tr><td><b><font face='&fontface.'
size='2'>" ROWCAT "</font></b></td>";
        LROWCAT=ROWCAT;
  END;
  END;

%end;

  ********************************************;
  **** Need to output different formats    ****;
  ********************************************;
%if &var3.=0 %then %do;

        IF ROWCAT IN("Under Age 6-Benchmark",
                     "6-12 Years-Benchmark") THEN DO;  *** no significance ***;
         IF  SCORE=.  THEN  PUT  "<td  align='center'  valign='bottom'><b><font  face='&fontface.'
color=&blue. size='2'>NA<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
         ELSE IF SCORE=.A THEN PUT "<td align='center' valign='bottom'><b><font face='&fontface.'
color=&blue. size='2'>NA<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
         ELSE  PUT  "<td  align='center'  valign='bottom'><b><font  face='&fontface.'  color=&blue.
size='2'>" SCORE 3.0 "<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
        END;
        ELSE DO;
          IF SCORE=. THEN DO;
            PUT      "<td      align='center'      valign='bottom'><b><font      face='&fontface.'
size='2'>***<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
          END;
          ELSE IF SCORE=.A THEN DO;
            PUT "<td align='center' valign='bottom'><b><font face='&fontface.' size='2'>NA<!CODE=
" +(-1) ORDER Z5.  "></font></b></td>";
          END;
          ELSE IF SCORE=.P THEN DO;
            PUT "<td align='center' valign='bottom'><font face='&fontface.' size='2'>NP<!CODE= "
+(-1) ORDER Z5.  "></font></td>";
          END;
          ELSE DO;
            IF  SIG=1  THEN  PUT  "<td  align='center'  valign='bottom'><b><font  face='&fontface.'
size='2' color=&green.>" SCORE 3.0  "<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            **ELSE   IF   SIG=.   THEN   PUT   "<td   align='center'   valign='bottom'><b><font
face='&fontface.' size='2'>***<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            ELSE   IF   SIG=.A   THEN   PUT   "<td   align='center'   valign='bottom'><b><font
face='&fontface.' size='2'>NA<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            ELSE   IF   SIG=-1   THEN   PUT   "<td   align='center'   valign='bottom'><i><font
face='&fontface.'   size='2'   color=&red.>"   SCORE   3.0      "<!CODE=   "   +(-1)   ORDER   Z5.
"></font></i></td>";
            ELSE PUT "<td align='center' valign='bottom'><font face='&fontface.' size='2'>" SCORE
3.0 "<!CODE= " +(-1) ORDER Z5.  "></font></td>";
```

```
          END;

      END;

%end;

/* SUBSETS */
%else %do;
IF BENTYPE^="1999" THEN DO;
IF BENEFIT IN ("Attitudes Toward TRICARE Prime",
                "Waiting times")  THEN DO;
   IF ROWCAT^IN("Under Age 6-Benchmark",
                "6-12 Years-Benchmark") THEN DO;
          IF SCORE=. THEN DO;
            PUT      "<td      align='center'      valign='bottom'><b><font      face='&fontface.'
size='2'>***<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
          END;
          ELSE IF SCORE=.A THEN DO;
             PUT "<td align='center' valign='bottom'><b><font face='&fontface.' size='2'>NA<!CODE=
" +(-1) ORDER Z5.  "></font></b></td>";
          END;
          ELSE DO;
            IF  SIG=1  THEN  PUT  "<td  align='center'  valign='bottom'><b><font  face='&fontface.'
size='2' color=&green.>" SCORE 3.0  "<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            **ELSE   IF   SIG=.   THEN   PUT   "<td   align='center'   valign='bottom'><b><font
face='&fontface.' size='2'>***<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            ELSE   IF   SIG=.A   THEN   PUT   "<td   align='center'   valign='bottom'><b><font
face='&fontface.' size='2'>NA<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            ELSE   IF   SIG=-1   THEN   PUT   "<td   align='center'   valign='bottom'><i><font
face='&fontface.'   size='2'   color=&red.>"   SCORE   3.0      "<!CODE=   "   +(-1)   ORDER   Z5.
"></font></i></td>";
             ELSE PUT "<td align='center' valign='bottom'><font face='&fontface.' size='2'>" SCORE
3.0 "<!CODE= " +(-1) ORDER Z5.  "></font></td>";
          END;
   END;
END;
ELSE IF BENEFIT IN("Getting Needed Care",
                "How Well Doctors Communicate",
                "Courteous and Helpful Office Staff",
                "Customer Service",
                "Personal Doctor or Nurse") THEN DO; *** format=3.2 ***;

        IF ROWCAT IN("Under Age 6-Benchmark",
                     "6-12 Years-Benchmark") THEN DO;  *** no significance ***;
         IF  SCORE=.  THEN  PUT  "<td  align='center'  valign='bottom'><b><font  face='&fontface.'
color=&blue. size='2'>***<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
         ELSE IF SCORE=.A THEN PUT "<td align='center' valign='bottom'><b><font face='&fontface.'
color=&blue. size='2'>NA<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
         ELSE  PUT  "<td  align='center'  valign='bottom'><b><font  face='&fontface.'  color=&blue.
size='2'>" SCORE 3.0 "<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
        END;
        ELSE DO;
          IF SCORE=. THEN DO;
            PUT      "<td      align='center'      valign='bottom'><b><font      face='&fontface.'
size='2'>***<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
          END;
          ELSE IF SCORE=.A THEN DO;
            PUT "<td align='center' valign='bottom'><b><font face='&fontface.' size='2'>NA<!CODE=
" +(-1) ORDER Z5.  "></font></b></td>";
          END;
          ELSE DO;
            IF  SIG=1  THEN  PUT  "<td  align='center'  valign='bottom'><b><font  face='&fontface.'
size='2' color=&green.>" SCORE 3.0  "<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            **ELSE   IF   SIG=.   THEN   PUT   "<td   align='center'   valign='bottom'><b><font
face='&fontface.' size='2'>***<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            ELSE   IF   SIG=.A   THEN   PUT   "<td   align='center'   valign='bottom'><b><font
face='&fontface.' size='2'>NA<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            ELSE   IF   SIG=-1   THEN   PUT   "<td   align='center'   valign='bottom'><i><font
face='&fontface.'   size='2'   color=&red.>"   SCORE   3.0      "<!CODE=   "   +(-1)   ORDER   Z5.
"></font></i></td>";
             ELSE PUT "<td align='center' valign='bottom'><font face='&fontface.' size='2'>" SCORE
3.0 "<!CODE= " +(-1) ORDER Z5.  "></font></td>";
          END;
```

```
        END;

END;
ELSE IF BENEFIT = "Getting Care Quickly" THEN DO; *** format=3.2 ***;

        IF ROWCAT IN("Under Age 6-Benchmark",
                     "6-12 Years-Benchmark") THEN DO;  *** no significance ***;
             IF BENTYPE="Wait More Than 30 Minutes Past Appointment" THEN PUT
             "<td   align='center'   valign='bottom'><b><font   face='&fontface.'   color=&blue.
size='2'>NA<!CODE= " +(-1) ORDER Z5.  "><sup>**</sup></font></b></td>";
             ELSE   IF   SCORE=.   THEN   PUT   "<td   align='center'   valign='bottom'><b><font
face='&fontface.' color=&blue. size='2'>***<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
             ELSE   IF   SCORE=.A   THEN   PUT   "<td   align='center'   valign='bottom'><b><font
face='&fontface.' color=&blue. size='2'>NA<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
             ELSE PUT "<td align='center' valign='bottom'><b><font face='&fontface.' color=&blue.
size='2'>" SCORE 3.0 "<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
        END;
        ELSE DO;
          IF SCORE=. THEN DO;
            PUT      "<td      align='center'      valign='bottom'><b><font      face='&fontface.'
size='2'>***<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
          END;
          ELSE IF SCORE=.A THEN DO;
            PUT "<td align='center' valign='bottom'><b><font face='&fontface.' size='2'>NA<!CODE=
" +(-1) ORDER Z5.  "></font></b></td>";
          END;
          ELSE DO;
            IF  SIG=1  THEN  PUT  "<td  align='center'  valign='bottom'><b><font  face='&fontface.'
size='2' color=&green.>" SCORE 3.0  "<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            **ELSE   IF   SIG=.   THEN   PUT   "<td   align='center'   valign='bottom'><b><font
face='&fontface.' size='2'>***<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            ELSE   IF   SIG=.A   THEN   PUT   "<td   align='center'   valign='bottom'><b><font
face='&fontface.' size='2'>NA<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            ELSE   IF   SIG=-1   THEN   PUT   "<td   align='center'   valign='bottom'><i><font
face='&fontface.'   size='2'   color=&red.>"   SCORE   3.0      "<!CODE=   "   +(-1)   ORDER   Z5.
"></font></i></td>";
            ELSE PUT "<td align='center' valign='bottom'><font face='&fontface.' size='2'>" SCORE
3.0 "<!CODE= " +(-1) ORDER Z5.  "></font></td>";
          END;
        END;

END;
  ELSE DO; *** format=3.0 ***;

        IF ROWCAT IN("Under Age 6-Benchmark",
                     "6-12 Years-Benchmark") THEN DO;  *** no significance ***;
         IF  SCORE=.  THEN  PUT  "<td  align='center'  valign='bottom'><b><font  face='&fontface.'
color=&blue. size='2'>NA<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
         ELSE IF SCORE=.A THEN PUT "<td align='center' valign='bottom'><b><font face='&fontface.'
color=&blue. size='2'>NA<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
         ELSE  PUT  "<td  align='center'  valign='bottom'><b><font  face='&fontface.'  color=&blue.
size='2'>" SCORE 3.0 "<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
        END;
        ELSE DO;
          IF SCORE=. THEN DO;
            PUT      "<td      align='center'      valign='bottom'><b><font      face='&fontface.'
size='2'>***<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
          END;
          ELSE IF SCORE=.A THEN DO;
            PUT "<td align='center' valign='bottom'><b><font face='&fontface.' size='2'>NA<!CODE=
" +(-1) ORDER Z5.  "></font></b></td>";
          END;
          ELSE IF SCORE=.P THEN DO;
            PUT "<td align='center' valign='bottom'><font face='&fontface.' size='2'>NP<!CODE= "
+(-1) ORDER Z5.  "></font></td>";
          END;
          ELSE DO;
            IF  SIG=1  THEN  PUT  "<td  align='center'  valign='bottom'><b><font  face='&fontface.'
size='2' color=&green.>" SCORE 3.0  "<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            **ELSE   IF   SIG=.   THEN   PUT   "<td   align='center'   valign='bottom'><b><font
face='&fontface.' size='2'>***<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
            ELSE   IF   SIG=.A   THEN   PUT   "<td   align='center'   valign='bottom'><b><font
face='&fontface.' size='2'>NA<!CODE= " +(-1) ORDER Z5.  "></font></b></td>";
```

```
            ELSE   IF   SIG=-1   THEN   PUT   "<td   align='center'   valign='bottom'><i><font
face='&fontface.'   size='2'   color=&red.>"   SCORE   3.0      "<!CODE=   "   +(-1)   ORDER   Z5.
"></font></i></td>";
            ELSE PUT "<td align='center' valign='bottom'><font face='&fontface.' size='2'>" SCORE
3.0 "<!CODE= " +(-1) ORDER Z5.  "></font></td>";
          END;
        END;

  END;

END;

%end;




  IF EOF THEN DO;
  IF BENEFIT = "Getting Care Quickly" THEN DO;

     /** MF Changes **/
     PUT "<tr>";
     PUT "     <td colspan='&colmns.'><font face='Arial,Helvetica,Swiss,Geneva' size='2'>Source:
1999 Health Care Survey of DOD Beneficiaries</font><font face='Arial,Helvetica,Swiss,Geneva'
size='2' color='#009933'><br>";
     PUT "    <b>Indicates score significantly exceeds benchmark</b></font><b>&htmlsp.<br>";
     PUT "    </b><font face='Arial,Helvetica,Swiss,Geneva' size='2' color='#cc0000'><i>Indicates
score significantly falls short of benchmark</i></font><br>";
     PUT "      <font face='Arial,Helvetica,Swiss,Geneva' size='2'>NA Indicates benchmark not
available</font><br>";
     PUT "    <font face='Arial,Helvetica,Swiss,Geneva' size='2'><sup>*</sup>Benchmarks are only
available for children under age 6 and 6 to 12-year-olds.</font><br>";
     PUT "     <font face='Arial,Helvetica,Swiss,Geneva' size='2'><sup>**</sup>The benchmark is
based on a waiting time of 15 minutes or less past appointment time.</font><br>";
     PUT "     <font face='Arial,Helvetica,Swiss,Geneva' size='2'>*** Indicates significance not
available</font></td>";
     PUT "</tr>";

  END;
  ELSE IF (BENEFIT = "Primary Care Manager" OR &var3=0) THEN DO;
     /** MF Changes **/
     PUT "<tr>";
     PUT "     <td colspan='&colmns.'><font face='Arial,Helvetica,Swiss,Geneva' size='2'>Source:
1999 Health Care Survey of DOD Beneficiaries</font><font face='Arial,Helvetica,Swiss,Geneva'
size='2' color='#009933'><br>";
     PUT "    <b>Indicates score significantly exceeds benchmark</b></font><b>&htmlsp.<br>";
     PUT "    </b><font face='Arial,Helvetica,Swiss,Geneva' size='2' color='#cc0000'><i>Indicates
score significantly falls short of benchmark</i></font><br>";
     PUT "      <font face='Arial,Helvetica,Swiss,Geneva' size='2'>NA Indicates benchmark not
available</font><br>";
     PUT "     <font face='Arial,Helvetica,Swiss,Geneva' size='2'>NP Indicates not enrolled in
TRICARE Prime</font><br>";
     PUT "    <font face='Arial,Helvetica,Swiss,Geneva' size='2'><sup>*</sup>Benchmarks are only
available for children under age 6 and 6 to 12-year-olds.</font><br>";
     PUT "     <font face='Arial,Helvetica,Swiss,Geneva' size='2'>*** Indicates significance not
available</font></td>";
     PUT "</tr>";
  END;
  ELSE DO;
     /** MF Changes **/
     PUT "<tr>";
     PUT "     <td colspan='&colmns.'><font face='Arial,Helvetica,Swiss,Geneva' size='2'>Source:
1999 Health Care Survey of DOD Beneficiaries</font><font face='Arial,Helvetica,Swiss,Geneva'
size='2' color='#009933'><br>";
     PUT "    <b>Indicates score significantly exceeds benchmark</b></font><b>&htmlsp.<br>";
     PUT "    </b><font face='Arial,Helvetica,Swiss,Geneva' size='2' color='#cc0000'><i>Indicates
score significantly falls short of benchmark</i></font><br>";
     PUT "      <font face='Arial,Helvetica,Swiss,Geneva' size='2'>NA Indicates benchmark not
available</font><br>";
     PUT "    <font face='Arial,Helvetica,Swiss,Geneva' size='2'><sup>*</sup>Benchmarks are only
available for children under age 6 and 6 to 12-year-olds.</font><br>";
     PUT "     <font face='Arial,Helvetica,Swiss,Geneva' size='2'>*** Indicates significance not
available</font></td>";
```

```
     PUT "</tr>";
  END;
  END;

RUN;


*************************************;
****  Print out footer info      ****;
*************************************;
DATA _NULL_;
    FILE "&FILEOUT1." MOD;


    HREFTOP=COMPRESS("c&var1.-0-0.htm");


    /** MF Changes **/
    PUT "<tr>";
    PUT "   <td colspan='&colmns.'>";
    PUT "      <center>";
    PUT "             <a href='index.htm'><img src=&home_button. border='0' alt='Return to Main
Page'></a>&htmlsp.&htmlsp.";
    PUT "         <a href=""" HREFTOP  +(-1) """><img src=&back_button. border='0' alt='Return to
Top Level'></a>&htmlsp.&htmlsp.";
    PUT "         <a href='help.htm'><img src=&help_button. border='0' alt='Help'></a><br>";
    PUT "          <font face='Arial,Helvetica,Swiss,Geneva' size='2'><b>Click below to view this
table by other groups<br>";
    PUT "         </b></font>";

    majgrp1=COMPRESS("c1-&var2.-&var3..htm");
    majgrp2=COMPRESS("c2-&var2.-&var3..htm");
    majgrp3=COMPRESS("c3-&var2.-&var3..htm");
    majgrp4=COMPRESS("c4-&var2.-&var3..htm");


    PUT  "<a  href="""  MAJGRP1  +(-1)  """><font  face='&fontface.'  size='2'>All  CONUS  MHS
Children</font></a>&htmlsp.&htmlsp.";
    PUT "<a href="""  MAJGRP2 +(-1) """><font face='&fontface.' size='2'>Children in Regions 1, 2
& 5</font></a>&htmlsp.&htmlsp.";
    PUT "<a href="""  MAJGRP3 +(-1) """><font face='&fontface.' size='2'>Children in Regions 6, 9-
12 & Alaska</font></a>&htmlsp.&htmlsp.";
    PUT "<a href="""  MAJGRP4 +(-1) """><font face='&fontface.' size='2'>Children in Regions 3, 4
& 7/8</font></a>";


RUN;


/*** Close HTML page ***/
DATA _NULL_;
  FILE "&FILEOUT1." MOD;

  PUT "</center></td></tr></table>";
  PUT "</body></html>";

RUN;




%MEND MKHTML;
*************************************************************************;
*************************************************************************;


*********************************************;
**** Create macros to call MKHTML macros ****;
*********************************************;

%MACRO DOALL1();
      %DO J=1 %TO 4;
         %DO K=0 %TO 8;
```

```
            %MKHTML(&J.,0,&K.);
            %MKHTML(&J.,0,&K.);
            %MKHTML(&J.,0,&K.);
            %MKHTML(&J.,0,&K.);
         %END;
      %END;
%MEND DOALL1;


%DOALL1;
```