UNITED STATES DEPARTMENT OF COMMERCE
Economics and Statistics Administration
U.S. Census Bureau
Washington, DC 20233-0001

December 30, 2002

DSSD A.C.E. REVISION II MEMORANDUM SERIES #PP-11

PRED CENSUS AND SURVEY MEASUREMENT STAFF MEMORANDUM SERIES:
CSM-A.C.E. REVISION II-09R

MEMORANDUM FOR: Donna Kostanich
Chair, A.C.E. Revision II Planning and Management Group
Assistant Division Chief, Sampling and Estimation
Decennial Statistical Studies Division

From: Mary H. Mulry *signed 12/30/02*
Chair, A.C.E. Revision II Quality Indicators Group
Statistical Research Division

Through: David Hubble *signed 12/30/02* RJP for
Assistant Division Chief, Evaluations
Planning, Research, and Evaluation Division

Prepared By: Anne T. Kearney
Mathematical Statistician
Planning, Research, and Evaluation Division

Subject: A.C.E. Revision II Missing Data Evaluation Processing Specification

Attached is the specification for running the SAS programs used in calculating the estimates in the A.C.E. Revision II missing data evaluation.

1. INTRODUCTION

For the production missing data evaluation we developed 128 alternatives to the production missing data system (Keathley, et al., 2001). We calculated coverage correction factors (CCFs) and dual system estimates (DSEs) using the alternative missing data methodologies. There were 128 estimates of CCFs and DSEs by production poststratra. We adjusted the CCFs from the original evaluation to make the original estimates comparable to the A.C.E. Revision II missing data evaluation estimates.

For the A.C.E. Revision II missing data evaluation we used the 128 alternatives that were developed in the production evaluation, and recalculated the CCFs and DSEs using the A.C.E. Revision II poststrata, and the original estimation methodology and adjusted the estimates of variation in the 128 alternatives for differences in the missing data procedures between production and A.C.E. Revision II and to compensate for the too small spread in the production estimates (see Kearney, 2002b).

This specification provides the instructions for running the A.C.E. Revision II missing data evaluation system of SAS programs in order to get the estimates of variation due to the A.C.E. Revision II missing data procedures.

2. PROCEDURES FOR GETTING A.C.E. Revision II MISSING DATA EVALUATION ESTIMATES

The programs listed in this section are located on the Statistical Research Division's UNIX machines in the following directory: **/home/akearney/p1eval2k/reACE** (where case matters).

Program 1. <u>Add A.C.E. Revision II Poststrata to the 128 E-Sample and 128 P-Sample Missing Data Alternative Person Files</u>

The program addnewpost.sas reads the 128 E-sample and 128 P-sample missing data alternative person files and the A.C.E. Revision II E-sample and P-sample person files. From the A.C.E. Revision II E-sample file, we read the revised poststrata (and some other variables) and append them to the end of each record on the 128 E-sample files. A similar procedure is followed for the P-sample. See Attachment 1.

Program 2. <u>Cross Classify the A.C.E. Revision II E-Sample Poststrata with the A.C.E. Revision II P-Sample Poststrata</u>

The program crosspostfin.sas cross classifies 744 E-sample poststrata (93 poststrata groups by 8 age/sex classifications) with 512 P-sample poststrata (64 poststrata groups by 8 age/sex categories). There are 7,584 crossed poststrata. See Attachment 2.

Program 3. Add Records to the Crossed Poststrata Level File of Census Counts to Account for the Poststrata with No People in them.

The program zerocellC.sas (case matters) adds records to the file of crossed poststrata census counts so that the records with a zero census count are represented. See Attachment 3.

Program 4. Calculate the Vectors of CCFs for the New Poststrata (7584)

The program vector.sas calculates 128 vectors of CCFs by A.C.E. Revision II poststrata for the full crossing (7584). There are 128 of the 7584 crossed poststrata that have an estimate of 0 because there are no census people in the poststrata. See Attachment 4.

Programs 5, 6, 7, and 8. Prepare Files for Input into DSE Programs to Calculate the High and Low Estimates

The programs psamphighlow.sas and esamphighlow.sas take the production estimation output files and sets the people with unresolved match or residence status (P-Sample) or unresolved correct enumeration status (E-Sample) to 0 or 1 to get the maximum and minimum DSEs. Similarly, the programs psampreace.sas and esampreace.sas take the A.C.E. Revision II full sample files and A.C.E. Revision II evaluation level files and set the people with unresolved status equal to 0 or 1. See Attachments 5, 6, 7, and 8.

Programs 9, 10 and 11. Calculate the $\mathbf{f}_{\mathbf{impute}}\mathbf{(k)}$ and **Var(Di)**

The program calceta.sas calculates the adjustment $\eta = (\delta'/\delta)$ due to changes in the missing data procedures for A.C.E. Revision II. See Kearney (2002b). The program varDifinnoiml.sas (case matters) calculates the adjustment factors adjusted for the uncertainty in the A.C.E. Revision II dual system estimates (DSEs) due to choice of imputation model. These are the $\mathbf{f}_{\mathbf{impute}}\mathbf{(k)}$ in Kearney (2002b). The program calcvarD.sas calculates the standard deviation and range that are reported as the A.C.E. Revision II Evaluation standard deviation (see Kearney, 2002a). See Attachments 9, 10, and 11.

Program 12. Adjust the Standard Deviation from the Production Missing Data Evaluation

The program calcvarDescap.sas takes the 128 vectors of CCFs from the original missing data evaluation and adjusts them so that the original standard deviation can be compared to the A.C.E. Revision II missing data evaluation standard deviation. See Attachment 12.

3. REFERENCES

Kearney, A.T. (2002a) Final Report on the A.C.E. Revision II Missing Data Evaluation, December 2002.

Kearney, A.T. (2002b) DSSD A.C.E. Revision II MEMORANDUM SERIES # PP10, PRED CENSUS AND SURVEY MEASUREMENT STAFF MEMORANDUM SERIES: CSM-A.C.E. Revision II-R4R, A.C.E. Revision II Missing Data Evaluation Study Plan, December 27, 2002.

Keathley, D., Kearney, A., Bell, W. (2001), paper for the Executive Steering Committee For A.C.E. Policy II, Report 12, ESCAP II: Analysis of Missing Data Alternatives for the Accuracy and Coverage Evaluation, October 11, 2001.

Spencer, B.D., Kearney, A.T., Keathley, D., Petroni, R., Belin, T., Mulry, M.H. (2002) Draft report September 26, 2002, Quantifying Bias from Missing Data Procedures in the 2000 A.C.E.

addnewpost.sas **ATTACHMENT 1**

```
* addnewpost.sas : this program adds the reACE full-prod. level poststrata to
  the 128 P- and 128 E-sample missing data files.  We will calculate DSEs by
  the new poststrata to get V for the missing data variance.;

options ls = 80 missing = '' ;
/*
data p_post;
   infile 'p-sample person level reACE post strata, full ACE';
   input clust msn wmsn per newpost;
*/

data p_post;
   infile '/home/akearney/p1eval2k/dmba01/pfullsam.dat' truncover lrecl=365;
   input @17 clust $char6. @35 msn $char5. @40 wmsn $char4. @54 per $char4.
         @352 newage $char3. @355 pfulllong $char7. @362 pfullshort $char4.;

proc sort data = p_post;
   by clust msn wmsn per;

data e_post;
   infile '/home/akearney/p1eval2k/dmba01/efullsam.dat' truncover lrecl= 400;
   input @17 clust $char6. @35 puid $char12. @47 pseq $char2. @49 pnc $char5.
         @354 srhhmem $char1. @357 spft $char2. @365 srsource $char2.
         @382 srepdate $char4. @390 efulllong $char7. @397 efullshort $char4.;

proc sort data = e_post;
   by clust puid pseq pnc;

%macro newpost(alt= , samp=, num= ,ltr= ,recl= , mer1= , mer2= ,mer3= );

data &samp.&alt.&num.&ltr;
infile "/vol/srdu8.files4/more-p1eval2k/alt&alt.&samp.&num.&ltr..dat"
          truncover lrecl=&recl;

%if &samp = p %then %do;

/*  Input statement for the production Estimation Output File - P Sample */

   input   @1 fipsstat $char2.
                @3 region $char1.
                @4 div $char1.
                @5 aicind 1.
                @6 sampstr 1.
                @7 tea 1.
                @8 ccou $char3.
                @11 tract $char6.
                @17 clust $char6.
                @23 block $char6.
                @29 lco $char4.
                @33 eclusurb 1.
                @34 tesindo 1.
                @35 msn $char5.
                @40 wmsn $char4.
                @44 finoutc 2.
                @46 finouti 2.
                @48 cst $char2.
```

```
    @50 cou $char3.
    @54 per $char4.
    @58 st $char2.
    @60 mid $char12.
    @72 mseq $char2.
    @74 mpnc $char5.
    @79 dtcode 2.
    @81 prepospc 2.
    @83 prenddc 2.
    @85 posspsc 2.
    @87 ndatdfc 2.
    @89 posspsi 2.
    @91 ndatdfi 2.
    @93 popct 2.
    @95 tenure 1.
    @96 toba 1.
    @97 age 3.
    @100 race 2.
    @102 sex 1.
    @103 hisp 1.
    @104 relate 2.
    @106 prxyin 1.
    @107 moverper 1.
    @108 moverhh 1.
    @109 rsc $char1.
    @113 cmpmat $char2.
    @115 clemat $char2.
    @117 bfumat $char2.
    @119 afumat $char2.
    @121 finmat $char2.
    @123 fuflag $char1.
    @124 mateind 1.
    @125 tesper 1.
    @126 wpinit 1.
    @127 wpfin 1.
    @128 addcde 1.
    @129 teswgt 7.4
@136 fu_code1 $char1.
@137 fu_code2 1.
    @146 tenure2 1.
    @147 age2 1.
    @148 race2 1.
    @150 hisp2 1.
    @151 rtype 1.
    @152 relat2 1.
    @153 tenifl 1.
    @154 ageifl 1.
    @155 sexifl 1.
    @156 hispif 1.
    @157 raceif 1.
    @158 amtimp 1.
    @159 bfugp 2.
    @161 frs 5.3
    @166 rprob 10.8
    @176 mprob 11.8
    @187 pwght 14.6
    @201 niwgto 15.6
    @216 niwgti 15.6
```

```
                @231 tesfinwt 15.6
                @246 fipsmsa $char4.
                @250 msasize $char1.
                @251 msaname $char60.
                @311 retrate 5.1
                @316 rrnum 10.
                @326 rrden 10.
                @336 domain 1.
                @337 cutoff 5.1
                @342 rratind 1.
                @343 msatea 1.
                @344 agesex 1.
                @345 poststr $char3.
                @348 cpoststr $char3.       ;

%end;  %* samp = p;

%else %if &samp = e %then %do;

        input   @1 cst $char2.
                @3 region $char1.
                @4 div $char1.
                @5 aicind 1.
                @6 sampstr 1.
                @7 tea 1.
                @8 ccou $char3.
                @11 tract $char6.
                @17 clust $char6.
                @23 block $char6.
                @29 lco $char4.
                @33 eclusurb 1.
                @34 tesind 1.
                @35 puid $char12.
                @47 pseq $char2.
                @49 pnc $char5.
                @54 mpsind $char1.
                @55 mmspot $char5.
                @60 munitid $char4.
                @64 mperid $char4.
                @68 dtcode 2.
                @70 nesamp 2.
                @72 inp 4.
                @76 rtenure 1.
                @77 page 3.
                @80 cpage 3.
                @83 race 2.
                @85 psex 1.
                @86 pspan 1.
                @87 prelsup 2.
                @89 numedup 1.
                @90 numdup 2.
                @92 nru 1.
                @93 cmpmat $char2.
                @95 clemat $char2.
                @97 bfumat $char2.
                @99 afumat $char2.
                @101 finmat $char2.
                @103 fuflag $char1.
                @104 tesper 1.
                @105 mateind 1.
                @106 wpinit 1.
                @107 wpfin 1.
                @108 addcde 1.
                @109 strcde 1.
                @110 ewght 14.6
                @124 fage 1.
                @125 frace 1.
                @126 ftenure 1.
                @127 fspan 1.
                @128 fsex 1.
                @129 frel 1.
                @130 teswgt 7.4
                @137 afnmathu $char2.
                @139 prcode 1.
                @140 mhumspot $char5.
                @145 mhuwmsid $char4.
                @149 st $char2.
                @151 cou $char3.
            @154 fu_code1 $char1.
            @155 fu_code2 1.
                @159 tenure2 1.
                @160 age2 1.
                @161 race2 1.
                @163 relate2 1.
                @164 tenifl 1.
                @165 ageifl 1.
                @166 sexifl 1.
                @167 hispif 1.
                @168 raceif 1.
                @169 amtimp 1.
                @170 bfugp 2.
                @172 fes 5.3
                @177 ceprobi 10.8
                @187 ceprobf 11.8
                @198 tesfinwt 14.6
                @212 fipsmsa $char4.
                @216 msasize $char1.
                @217 msaname $char60.
                @277 retrate 5.1
                @282 rrnum 10.
                @292 rrden 10.
                @302 domain 1.
                @303 cutoff 5.1
                @308 rratind 1.
                @309 msatea 1.
                @310 agesex 1.
                @311 poststr $char3.
                @314 cpoststr $char3.     ;
%end; %* samp = e;

proc sort data = &samp.&alt.&num.&ltr;
   by clust &mer1 &mer2 &mer3 ;

data mergepost;
   merge  &samp._post (in = a) &samp.&alt.&num.&ltr (in = b);
      by clust &mer1 &mer2 &mer3 ;
   if b;

proc datasets library = work memtype = data;
   delete &samp.&alt.&num.&ltr;
run;

*proc freq data = mergepost;
*   tables newpost / missing ;
*   title "mergepost, samp = &samp  how many records on the file? ";

data _null_;
   set mergepost;
  file "/vol/srdu8.files4/more-p1eval2k/reACE/realt&alt.&samp.&num.&ltr..dat"
        lrecl = &recl ;
/*  Input statement for the Missing Data Output File - P Sample */
/*  Estimation variables are included but commented out for now */

%if &samp = p %then %do;

   put   @1 fipsstat $char2.
                @3 region $char1.
                @4 div $char1.
                @5 aicind 1.
                @6 sampstr 1.
                @7 tea 1.
                @8 ccou $char3.
                @11 tract $char6.
                @17 clust $char6.
                @23 block $char6.
                @29 lco $char4.
                @33 eclusurb 1.
                @34 tesindo 1.
                @35 msn $char5.
                @40 wmsn $char4.
                @44 finoutc 2.
                @46 finouti 2.
                @48 cst $char2.
                @50 cou $char3.
                @54 per $char4.
                @58 st $char2.
                @60 mid $char12.
                @72 mseq $char2.
                @74 mpnc $char5.
                @79 dtcode 2.
                @81 prepospc 2.
                @83 prenddc 2.
                @85 posspsc 2.
                @87 ndatdfc 2.
                @89 posspsi 2.
                @91 ndatdfi 2.
                @93 popct 2.
                @95 tenure 1.
                @96 toba 1.
                @97 age 3.
                @100 race 2.
                @102 sex 1.
                @103 hisp 1.
                @104 relate 2.
                @106 prxyin 1.
                @107 moverper 1.
                @108 moverhh 1.
                @109 rsc $char1.
                @113 cmpmat $char2.
                @115 clemat $char2.
                @117 bfumat $char2.
                @119 afumat $char2.
                @121 finmat $char2.
                @123 fuflag $char1.
                @124 mateind 1.
                @125 tesper 1.
                @126 wpinit 1.
                @127 wpfin 1.
                @128 addcde 1.
                @129 teswgt 7.4
            @136 fu_code1 $char1.
            @137 fu_code2 1.
                @146 tenure2 1.
                @147 age2 1.
                @148 race2 1.
                @150 hisp2 1.
                @151 rtype 1.
                @152 relat2 1.
                @153 tenifl 1.
                @154 ageifl 1.
                @155 sexifl 1.
                @156 hispif 1.
                @157 raceif 1.
                @158 amtimp 1.
                @159 bfugp 2.
                @161 frs 5.3
                @166 rprob 10.8
                @176 mprob 11.8
                @187 pwght 14.6
                @201 niwgto 15.6
                @216 niwgti 15.6
                @231 tesfinwt 15.6
                @246 fipsmsa $char4.
                @250 msasize $char1.
                @251 msaname $char60.
                @311 retrate 5.1
                @316 rrnum 10.
                @326 rrden 10.
                @336 domain 1.
                @337 cutoff 5.1
                @342 rratind 1.
                @343 msatea 1.
                @344 agesex 1.
                @345 poststr $char3.
                @348 cpoststr $char3.
                @351 newage $char3.
                @354 pfulllong $char7.
                @361 pfullshort $char4.  ;

%end;  %* end if &samp = p;

%else %if &samp = e %then %do;
```

```
put   @1 cst $char2.
        @3 region $char1.
        @4 div $char1.
        @5 aicind 1.
        @6 sampstr 1.
        @7 tea 1.
        @8 ccou $char3.
        @11 tract $char6.
        @17 clust $char6.
        @23 block $char6.
        @29 lco $char4.
        @33 eclusurb 1.
        @34 tesind 1.
        @35 puid $char12.
        @47 pseq $char2.
        @49 pnc $char5.
        @54 mpsind $char1.
        @55 mmspot $char5.
        @60 munitid $char4.
        @64 mperid $char4.
        @68 dtcode 2.
        @70 nesamp 2.
        @72 inp 4.
        @76 rtenure 1.
        @77 page 3.
        @80 cpage 3.
        @83 race 2.
        @85 psex 1.
        @86 pspan 1.
        @87 prelsup 2.
        @89 numedup 1.
        @90 numdup 2.
        @92 nru 1.
        @93 cmpmat $char2.
        @95 clemat $char2.
        @97 bfumat $char2.
        @99 afumat $char2.
        @101 finmat $char2.
        @103 fuflag $char1.
        @104 tesper 1.
        @105 mateind 1.
        @106 wpinit 1.
        @107 wpfin 1.
        @108 addcde 1.
        @109 strcde 1.
        @110 ewght 14.6
        @124 fage 1.
        @125 frace 1.
        @126 ftenure 1.
        @127 fspan 1.
        @128 fsex 1.
        @129 frel 1.
        @130 teswgt 7.4
        @137 afnmathu $char2.
        @139 prcode 1.
        @140 mhumspot $char5.
        @145 mhuwmsid $char4.
        @149 st $char2.
```

```
                @151 cou $char3.
            @154 fu_code1 $char1.
            @155 fu_code2 1.
                @159 tenure2 1.
                @160 age2 1.
                @161 race2 1.
                @163 relate2 1.
                @164 tenifl 1.
                @165 ageifl 1.
                @166 sexifl 1.
                @167 hispif 1.
                @168 raceif 1.
                @169 amtimp 1.
                @170 bfugp 2.
                @172 fes 5.3
                @177 ceprobi 10.8
                @187 ceprobf 11.8
                @198 tesfinwt 14.6
                @212 fipsmsa $char4.
                @216 msasize $char1.
                @217 msaname $char60.
                @277 retrate 5.1
                @282 rrnum 10.
                @292 rrden 10.
                @302 domain 1.
                @303 cutoff 5.1
                @308 rratind 1.
                @309 msatea 1.
                @310 agesex 1.
                @311 poststr $char3.
                @314 cpoststr $char3.
                @317 srhhmem $char1.
                @318 spft $char2.
                @320 srsource $char2.
                @322 srepdate $char4.
                @326 efulllong $char7.
                @333 efullshort $char4.    ;

%end; %* samp = e;

/* The code below is to verify that the files output above have the correct
   number of records and that the records have the correct length.  If
   everything is correct then delete copy of the original input files. */

data _null_;
infile "/vol/srdu8.files4/more-p1eval2k/reACE/realt&alt.&samp.&num.&ltr..dat"
        lrecl = &recl truncover end = lastrec;

%if &samp = p %then %do;

   input @1 wholerec $char364.;

%end;
%else %if &samp = e %then %do;

   input @1 wholerec $char336. ;

%end;
```

```
   len = length(wholerec);

   if len in (22,336,364) then
      countrec + 1;

   if lastrec then do;
      put "countrec = " countrec " totrecs = " _n_;
      if _n_ = countrec then do;
put "_n_ = " _n_ ;
    X "/bin/rm -f
/vol/srdu8.files4/more-p1eval2k/alt&alt.&samp.&num.&ltr..dat" ;
put 'sysrc = ' "&sysrc";
   end;
   end;

%mend newpost;

* ******************************start P-Sample*************************;

%macro mac1;
%do n = 1 %to 16;
%newpost(alt=1,samp=p,num=&n,ltr= ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
%end;
%mend mac1;

%mac1;

%macro mac2;
%do n = 1 %to 16;
%newpost(alt=3,samp=p,num=&n,ltr= ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
%end;
%mend mac2;

%mac2;

%macro mac3;
%do n = 1 %to 6;
   %if &n = 1 %then
      %let l = a;
   %else %if &n = 2 %then
      %let l = b;
   %else %if &n = 3 %then
      %let l = c;
   %else %if &n = 4 %then
      %let l = d;
   %else %if &n = 5 %then
      %let l = e;
   %else %if &n = 6 %then
      %let l = f;

   %newpost(alt=4,samp=p,num=1,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=4,samp=p,num=2,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=4,samp=p,num=4,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=4,samp=p,num=5,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=4,samp=p,num=7,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=4,samp=p,num=9,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=4,samp=p,num=12,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=4,samp=p,num=16,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=5,samp=p,num=3,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=5,samp=p,num=6,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=5,samp=p,num=8,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=5,samp=p,num=10,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=5,samp=p,num=11,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=5,samp=p,num=13,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=5,samp=p,num=14,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);
   %newpost(alt=5,samp=p,num=15,ltr=&l ,recl=364,mer1=msn,mer2=wmsn,mer3=per);

%end;
%mend mac3;

%mac3;

* *******************************end P-Sample***********************;
* *****************************start E-Sample************************;

%macro mac4;
%do n = 1 %to 16;
%newpost(alt=1,samp=e,num=&n,ltr= ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
%end;
%mend mac4;

%mac4;

%macro mac5;
%do n = 1 %to 16;
%newpost(alt=3,samp=e,num=&n,ltr= ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
%end;
%mend mac5;

%mac5;

%macro mac6;
%do n = 1 %to 6;
   %if &n = 1 %then
      %let l = a;
   %else %if &n = 2 %then
      %let l = b;
   %else %if &n = 3 %then
      %let l = c;
   %else %if &n = 4 %then
      %let l = d;
   %else %if &n = 5 %then
      %let l = e;
   %else %if &n = 6 %then
      %let l = f;

   %newpost(alt=4,samp=e,num=1,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
   %newpost(alt=4,samp=e,num=2,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
   %newpost(alt=4,samp=e,num=4,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
   %newpost(alt=4,samp=e,num=5,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
   %newpost(alt=4,samp=e,num=7,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
   %newpost(alt=4,samp=e,num=9,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
  %newpost(alt=4,samp=e,num=12,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
  %newpost(alt=4,samp=e,num=16,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
```

```
   %newpost(alt=5,samp=e,num=3,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
   %newpost(alt=5,samp=e,num=6,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
   %newpost(alt=5,samp=e,num=8,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
  %newpost(alt=5,samp=e,num=10,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
  %newpost(alt=5,samp=e,num=11,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
  %newpost(alt=5,samp=e,num=13,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
  %newpost(alt=5,samp=e,num=14,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);
  %newpost(alt=5,samp=e,num=15,ltr=&l ,recl=336,mer1=puid,mer2=pseq,mer3=pnc);

%end;
%mend mac6;

%mac6;

* ***************************end E-Sample******************************;

run;
```

crosspostfin.sas **ATTACHMENT 2**

```
* crosspostfin.sas : this program reads the production level E-sample file to
  form the new E-sample poststrata. This program crosses the ReACE E-sample
 poststrata with the ReACE P-sample poststrata.;

options ls = 80;
libname dmba01 '/home/akearney/p1eval2k/dmba01';
libname rev '/home/akearney/p1eval2k/reACE';
libname rev2 '/vol/srdu8.files4/more-p1eval2k/reACE';

* this part of the data step creates ReACE E poststrata before it was
available;

/*
data dmba01.efullsam;
   infile '/home/akearney/p1eval2k/dmba01/efullsam.dat' truncover lrecl = 400;
   input   @1 cst $char2.
                @3 region $char1.
                @4 div $char1.
                @5 aicind 1.
                @6 sampstr 1.
                @7 tea 1.
                @8 ccou $char3.
                @11 tract $char6.
                @17 cluster $char6.
                @23 block $char6.
                @29 lco $char4.
                @33 eclusurb 1.
                @34 tesind 1.
                @35 puid $char12.
                @47 pseq $char2.
                @49 pnc $char5.
                @54 mpsind $char1.
                @55 mmspot $char5.
                @60 munitid $char4.
                @64 mperid $char4.
                @68 dtcode 2.
                @70 nesamp 2.
                @72 inp 4.
                @76 rtenure 1.
                @77 page 3.
                @80 cpage 3.
                @83 race 2.
                @85 psex 1.
                @86 pspan 1.
                @87 prelsup 2.
                @89 numedup 1.
                @90 numdup 2.
                @92 nru 1.
                @93 cmpmat $char2.
                @95 clemat $char2.
                @97 bfumat $char2.
                @99 afumat $char2.
                @101 finmat $char2.
                @103 fuflag $char1.
                @104 tesper 1.
                @105 mateind 1.
```

```
             @106 wpinit 1.
             @107 wpfin 1.
             @108 addcde 1.
             @109 strcde 1.
             @110 ewght 14.6
             @124 fage 1.
             @125 frace 1.
             @126 ftenure 1.
             @127 fspan 1.
             @128 fsex 1.
             @129 frel 1.
             @130 teswgt 7.4
             @137 afnmathu $char2.
             @139 prcode 1.
             @140 mhumspot $char5.
             @145 mhuwmsid $char4.
             @149 st $char2.
             @151 cou $char3.
         @154 fu_code1 $char1.
         @155 fu_code2 1.
             @159 tenure2 1.
             @160 age2 1.
             @161 race2 1.
             @163 relate2 1.
             @164 tenifl 1.
             @165 ageifl 1.
             @166 sexifl 1.
             @167 hispif 1.
             @168 raceif 1.
             @169 amtimp 1.
             @170 bfugp 2.
             @172 fes 5.3
             @177 ceprobi 10.8
             @187 ceprobf 11.8
             @198 tesfinwt 14.6
             @212 fipsmsa $char4.
             @216 msasize $char1.
             @217 msaname $char60.
             @277 retrate 5.1
             @282 rrnum 10.
             @292 rrden 10.
             @302 domain 1.
             @303 cutoff 5.1
             @308 rratind 1.
             @309 msatea 1.
             @310 agesex 1.
             @311 poststr $char3.
             @314 cpoststr $char3.
             @354 srhhmem $char1.
             @357 spft $char2.
             @365 srsource $char2.
             @382 srepdate $char4.
             @390 efulllong $char7.
             @397 efullshort $char4. ;

length e744back $7;
retain e744back '       ';
```

```
length e5var $5;
length e93 $2;

e5var = substr(efulllong,1,5);
e93 = substr(efullshort,1,2);

if ((spft in ('05','06','17','18','19','20') or
   srsource in ('31','32','33')) and srhhmem in ('2','3')) then
   proxy = 1;
else
   proxy = 2;

edomain = domain;

etenure = tenure2;

if (srsource in ('01','  ') or '03' <= srsource <= '12' or
   '30' <= srsource <= '36') and srepdate <= '0324' then
   rdate = 1;
else if (srsource in ('01','  ') or '03' <= srsource <= '12' or
   '30' <= srsource <= '36') and srepdate >= '0325' then
   rdate = 2;
else if (srsource = '37' or '13' <= srsource <= '29') and
        srepdate <= '0601' then
   rdate = 3;
else if (srsource = '37' or '13' <= srsource <= '29') and
        srepdate >= '0602' then
   rdate = 4;

if (prelsup in (1,2) and inp in (2,3)) or (prelsup in (3,4,5) and
   page <= 17 and inp in (2,3)) then
   relsize = 1;
else if (prelsup in (1,2) and inp >= 4) or (prelsup in (3,4,5) and
   page <= 17 and inp >= 4) then
   relsize = 2;
else if prelsup = 1 and inp = 1 then
   relsize = 3;
else if (prelsup in (3,4,5) and page >= 18 and inp in (2,3)) or
        (prelsup in (6,7,8,9,10,11,12,13,14,15,16,17,18,19) and
        inp in (2,3)) then
   relsize = 4;
else if (prelsup in (3,4,5) and page >= 18 and inp >= 4) or
        (prelsup in (6,7,8,9,10,11,12,13,14,15,16,17,18,19) and
        inp >= 4) then
   relsize = 5;

if age2 = 1 then do;
   if 000 <= page <= 009 then
      agecat = 1;
   else if 010 <= page <= 017 then
      agecat = 2;
end;
else if age2 in (2,3,4) then
   agecat = age2 + 1;

esex = psex;

length epsgroup $2;
```

```
if proxy = 1 then do;
   if edomain = 7 then
      epsgroup = '87';
   else if edomain = 4 then
      epsgroup = '88';
   else if edomain = 3 then
      epsgroup = '89';
   else if edomain = 5 then
      epsgroup = '90';
   else if edomain = 6 then
      epsgroup = '91';
   else if edomain = 1 then
      epsgroup = '92';
   else if edomain = 2 then
      epsgroup = '93';
end;
else if proxy = 2 then do;
   if edomain = 7 then do;
      if etenure = 1 then do;
         if relsize = 1 then do;
            if rdate = 1 then
               epsgroup = '01';
            else if rdate = 2 then
               epsgroup = '02';
            else if rdate = 3 then
               epsgroup = '03';
            else if rdate = 4 then
               epsgroup = '04';
         end;
         else if relsize = 2 then do;
            if rdate = 1 then
               epsgroup = '05';
            else if rdate = 2 then
               epsgroup = '06';
            else if rdate = 3 then
               epsgroup = '07';
            else if rdate = 4 then
               epsgroup = '08';
         end;
         else if relsize = 3 then do;
            if rdate = 1 then
               epsgroup = '09';
            else if rdate = 2 then
               epsgroup = '10';
            else if rdate = 3 then
               epsgroup = '11';
            else if rdate = 4 then
               epsgroup = '12';
         end;
         else if relsize = 4 then do;
            if rdate = 1 then
               epsgroup = '13';
            else if rdate = 2 then
               epsgroup = '14';
            else if rdate = 3 then
               epsgroup = '15';
            else if rdate = 4 then
               epsgroup = '16';
         end;
         else if relsize = 5 then do;
            if rdate = 1 then
               epsgroup = '17';
            else if rdate = 2 then
               epsgroup = '18';
            else if rdate = 3 then
               epsgroup = '19';
            else if rdate = 4 then
               epsgroup = '20';
         end;
      end;
      else if etenure = 2 then do;
         if relsize in (1,2) then do;
            if rdate = 1 then
               epsgroup = '21';
            else if rdate = 2 then
               epsgroup = '22';
            else if rdate = 3 then
               epsgroup = '23';
            else if rdate = 4 then
               epsgroup = '24';
         end;
         else if relsize in (3,4,5) then do;
            if rdate = 1 then
               epsgroup = '25';
            else if rdate = 2 then
               epsgroup = '26';
            else if rdate = 3 then
               epsgroup = '27';
            else if rdate = 4 then
               epsgroup = '28';
         end;
      end;
   end;
   else if domain = 4 then do;
      if etenure = 1 then do;
         if relsize in (1,2) then do;
            if rdate = 1 then
               epsgroup = '29';
            else if rdate = 2 then
               epsgroup = '30';
            else if rdate = 3 then
               epsgroup = '31';
            else if rdate = 4 then
               epsgroup = '32';
         end;
         else if relsize in (3,4,5) then do;
            if rdate = 1 then
               epsgroup = '33';
            else if rdate = 2 then
               epsgroup = '34';
            else if rdate = 3 then
               epsgroup = '35';
            else if rdate = 4 then
               epsgroup = '36';
         end;
      end;
      else if etenure = 2 then do;
         if relsize in (1,2) then do;
            if rdate = 1 then
               epsgroup = '37';
            else if rdate = 2 then
               epsgroup = '38';
            else if rdate = 3 then
               epsgroup = '39';
            else if rdate = 4 then
               epsgroup = '40';
         end;
         else if relsize in (3,4,5) then do;
            if rdate = 1 then
               epsgroup = '41';
            else if rdate = 2 then
               epsgroup = '42';
            else if rdate = 3 then
               epsgroup = '43';
            else if rdate = 4 then
               epsgroup = '44';
         end;
      end;
   end;
   else if domain = 3 then do;
      if etenure = 1 then do;
         if relsize in (1,2) then do;
            if rdate = 1 then
               epsgroup = '45';
            else if rdate = 2 then
               epsgroup = '46';
            else if rdate = 3 then
               epsgroup = '47';
            else if rdate = 4 then
               epsgroup = '48';
         end;
         else if relsize in (3,4,5) then do;
            if rdate = 1 then
               epsgroup = '49';
            else if rdate = 2 then
               epsgroup = '50';
            else if rdate = 3 then
               epsgroup = '51';
            else if rdate = 4 then
               epsgroup = '52';
         end;
      end;
      else if etenure = 2 then do;
         if relsize in (1,2) then do;
            if rdate = 1 then
               epsgroup = '53';
            else if rdate = 2 then
               epsgroup = '54';
            else if rdate = 3 then
               epsgroup = '55';
            else if rdate = 4 then
               epsgroup = '56';
         end;
         else if relsize in (3,4,5) then do;
```

```
         if rdate = 1 then
            epsgroup = '57';
         else if rdate = 2 then
            epsgroup = '58';
         else if rdate = 3 then
            epsgroup = '59';
         else if rdate = 4 then
            epsgroup = '60';
      end;
   end;
end;
else if edomain = 5 then do;
   if relsize in (1,2) then do;
      if rdate = 1 then
         epsgroup = '61';
      else if rdate = 2 then
         epsgroup = '62';
      else if rdate = 3 then
         epsgroup = '63';
      else if rdate = 4 then
         epsgroup = '64';
   end;
   else if relsize in (3,4,5) then do;
      if rdate = 1 then
         epsgroup = '65';
      else if rdate = 2 then
         epsgroup = '66';
      else if rdate = 3 then
         epsgroup = '67';
      else if rdate = 4 then
         epsgroup = '68';
   end;
end;
else if edomain = 6 then do;
   if relsize in (1,2) then do;
      if rdate = 1 then
         epsgroup = '69';
      else if rdate = 2 then
         epsgroup = '70';
      else if rdate = 3 then
         epsgroup = '71';
      else if rdate = 4 then
         epsgroup = '72';
   end;
   else if relsize in (3,4,5) then do;
      if rdate = 1 then
         epsgroup = '73';
      else if rdate = 2 then
         epsgroup = '74';
      else if rdate = 3 then
         epsgroup = '75';
      else if rdate = 4 then
         epsgroup = '76';
   end;
end;
else if edomain = 1 then do;
   if relsize in (1,2) then
      epsgroup = '77';
```

```
         else if relsize in (3,4,5) then
            epsgroup = '78';
      end;
      else if edomain = 2 then do;
         if relsize in (1,2) then do;
            if rdate = 1 then
               epsgroup = '79';
            else if rdate = 2 then
               epsgroup = '80';
            else if rdate = 3 then
               epsgroup = '81';
            else if rdate = 4 then
               epsgroup = '82';
         end;
         else if relsize in (3,4,5) then do;
            if rdate = 1 then
               epsgroup = '83';
            else if rdate = 2 then
               epsgroup = '84';
            else if rdate = 3 then
               epsgroup = '85';
            else if rdate = 4 then
               epsgroup = '86';
         end;
      end;
   end;

   length newepost525 $4;

* one age/sex group ;
   if epsgroup in ('90','93','11','67','85') then
      newepost525 = epsgroup || '0' || '0';
* two age/sex groups ;
   else if epsgroup in ('09','10','12','15','31','61','62','63','64',
                        '65','66','68','75','79','80','81','82','83',
                        '84','86') then do;
      if agecat in (1,2,3) then
         newepost525 = epsgroup || '8' || '0';
      else if agecat in (4,5) then
         newepost525 = epsgroup || '9' || '0';
   end;
* four age/sex groups ;
   else if epsgroup in ('91','92','29','73','30','74','07','19','35',
                        '47','51','71','16','32','72','76') then do;
      if agecat in (1,2) then
         newepost525 = epsgroup || '6' || '0';
      else if agecat in (3,4,5) then
         newepost525 = epsgroup || put(agecat,1.) || '0';
   end;
* remainder eight age/sex groups ;
   else do;
      if agecat in (1,2) then
         newepost525 = epsgroup || put(agecat,1.) || '0';
      else if agecat in (3,4,5) then
         newepost525 = epsgroup || put(agecat,1.) || put(esex,1.);
   end;

* calculate e744 (uncollapsed age/sex).;
* the code below still has the 5 e ps that have no people in them.  redo in
e744_check;
   length e744 $7;
   length e744short $4;

   if agecat = 1 then do;
      e744 = e5var || '1' || '0';
      e744short = e93 || '1' || '0';
   end;
   else if agecat = 2 then do;
      e744 = e5var || '2' || '0';
      e744short = e93 || '2' || '0';
   end;
   else if agecat = 3 then do;
      if esex = 1 then do;
         e744 = e5var || '3' || '1';
         e744short = e93 || '3' || '1';
      end;
      else if esex = 2 then do;
         e744 = e5var || '3' || '2';
         e744short = e93 || '3' || '2';
      end;
   end;
   else if agecat = 4 then do;
      if esex = 1 then do;
         e744 = e5var || '4' || '1';
         e744short = e93 || '4' || '1';
      end;
      else if esex = 2 then do;
         e744 = e5var || '4' || '2';
         e744short = e93 || '4' || '2';
      end;
   end;
   else if agecat = 5 then do;
      if esex = 1 then do;
         e744 = e5var || '5' || '1';
         e744short = e93 || '5' || '1';
      end;
      else if esex = 2 then do;
         e744 = e5var || '5' || '2';
         e744short = e93 || '5' || '2';
      end;
   end;
*/
proc sort data = dmba01.efullsam;
   by e5var;
/*
data rev.e744_check (keep = e744 e5var ecnt1 ecnt2 e744short);* e525short
e525long);
   set dmba01.efullsam;
      by e5var;

* calculate e744 (uncollapsed age/sex).;

   length e744 $7;
   length e744short $4;

   if last.e5var then do;
   do eage = 1 to 5;
      do esex = 1 to 2;
         if eage in (1,2) then do;
            e744 = e5var || put(eage,1.) || '0';
            e744short = e93 || put(eage,1.) || '0';
*            e525short = e93 || '80';
*            e525long = e5var || '80';
            ecnt1 + 1;
            output;
            leave; * process the esex loop only once for ages 1 and 2;
         end;
         else if eage in (3,4,5) then do;
            e744 = e5var || put(eage,1.) || put(esex,1.);
            e744short = e93 || put(eage,1.) || put(esex,1.);
            ecnt2 + 1;
            output;
         end;
      end;
   end;
   end;
*/
proc sort data = rev.e744_check nodupkey;
   by e744;

proc sort data = dmba01.efullsam;
   by e744;

data dmba01.efullsam;* not_e;
   merge dmba01.efullsam (in = a) rev.e744_check (in = b);
      by e744;

*proc print data = not_e;
*   var e744 efulllong efullshort e93 e5var ecnt1 ecnt2;
*   title 'Not in efullsam';

*proc freq data = dmba01.efullsam;
*   tables e744 e744short efullshort2 / missing ;
*   title 'how many efulllong2 not missing? should be 32';
run;

************************ begin P-Sample ***************************;
/*
data dmba01.pfullsam;
   infile '/home/akearney/p1eval2k/dmba01/pfullsam.dat' lrecl = 365 truncover;
   input   @1 fipsstat $char2.
                @3 region $char1.
                @4 div $char1.
                @5 aicind 1.
                @6 sampstr 1.
                @7 tea 1.
                @8 ccou $char3.
                @11 tract $char6.
                @17 cluster $char6.
                @23 block $char6.
                @29 lco $char4.
                @33 eclusurb 1.
                @34 tesindo 1.
                @35 ilmapspo $char5.
```

```
    @40 unitid $char4.
    @44 finoutc 2.
    @46 finouti 2.
    @48 cst $char2.
    @50 cou $char3.
    @54 perid $char4.
    @58 st $char2.
    @60 mid $char12.
    @72 mseq $char2.
    @74 mpnc $char5.
    @79 dtcode 2.
    @81 prepospc 2.
    @83 prenddc 2.
    @85 posspsc 2.
    @87 ndatdfc 2.
    @89 posspsi 2.
    @91 ndatdfi 2.
    @93 popct 2.
    @95 tenure 1.
    @96 toba 1.
    @97 age 3.
    @100 race 2.
    @102 sex 1.
    @103 hisp 1.
    @104 relate 2.
    @106 prxyin 1.
    @107 moverper 1.
    @108 moverhh 1.
    @109 rsc $char1.
    @113 cmpmat $char2.
    @115 clemat $char2.
    @117 bfumat $char2.
    @119 afumat $char2.
    @121 finmat $char2.
    @123 fuflag $char1.
    @124 mateind 1.
    @125 tesper 1.
    @126 wpinit 1.
    @127 wpfin 1.
    @128 addcde 1.
    @129 teswgt 7.4
@136 fu_code1 $char1.
@137 fu_code2 1.
    @146 tenure2 1.
    @147 age2 1.
    @148 race2 1.
    @150 hisp2 1.
    @151 rtype 1.
    @152 relat2 1.
    @153 tenifl 1.
    @154 ageifl 1.
    @155 sexifl 1.
    @156 hispif 1.
    @157 raceif 1.
    @158 amtimp 1.
    @159 bfugp 2.
    @161 frs 5.3
    @166 rprob 10.8
```

```
             @176 mprob 11.8
             @187 pwght 14.6
             @201 niwgto 15.6
             @216 niwgti 15.6
             @231 tesfinwt 15.6
             @246 fipsmsa $char4.
             @250 msasize $char1.
             @251 msaname $char60.
             @311 retrate 5.1
             @316 rrnum 10.
             @326 rrden 10.
             @336 domain 1.
             @337 cutoff 5.1
             @342 rratind 1.
             @343 msatea 1.
             @344 agesex 1.
             @345 poststr $char3.
             @348 cpoststr $char3.
             @352 newage $char3.
             @355 pfulllong $char7.
             @362 pfullshort $char4.       ;

length p5var $5;
length p64 $2;

p5var = substr(pfulllong,1,5);

p64 = trim(left(substr(pfullshort,1,2)));

length p512 $7;

if p64 in ('06','08','13','21','48','56','57','58') then do;
   if 0 <= newage <= 9 then do;
      p512 = p5var || '1' || '0';
      p512short = p64 || '1' || '0';
   end;
   else if 10 <= newage <= 17 then do;
      p512 = p5var || '2' || '0';
      p512short = p64 || '2' || '0';
   end;
   else if 18 <= newage <= 29 then do;
      if sex = 1 then do;
         p512 = p5var || '3' || '1';
         p512short = p64 || '3' || '1';
      end;
      else if sex = 2 then do;
         p512 = p5var || '3' || '2';
         p512short = p64 || '3' || '2';
      end;
   end;
   else if 30 <= newage <= 49 then do;
      if sex = 1 then do;
         p512 = p5var || '4' || '1';
         p512short = p64 || '4' || '1';
      end;
      else if sex = 2 then do;
         p512 = p5var || '4' || '2';
         p512short = p64 || '4' || '2';
         end;
      end;
      else if newage >= 50 then do;
         if sex = 1 then do;
            p512 = p5var || '5' || '1';
            p512short = p64 || '5' || '1';
         end;
         else if sex = 2 then do;
            p512 = p5var || '5' || '2';
            p512short = p64 || '5' || '2';
         end;
      end;
   end;
   else do;
      p512 = pfulllong;
      p512short = pfullshort;
   end;

run;
*/

*proc freq data = dmba01.pfullsam;
*   tables newage / out = freqage noprint;

*proc print data = freqage;
*   title 'what is age range?';

run;

*proc print data = dmba01.pfullsam (where=(p5var ne '     ') obs = 20);
*   var p5var pfulllong pfullshort ;
*   title 'P-Sample:  any missing values?';

****************************** get uncollapsed (age/sex) poststrata for E and
P **********;

proc sort data = dmba01.efullsam out = sortdate;
   by e744;

data rev.e744 (keep = e744 e744short e5var efulllong efullshort p64 e93
efulllong2
                      efullshort2 );
   set sortdate;
      by e744;

*  ecnt max value is less than 744 so can't assign it here.;

   length efulllong $ 7 efullshort $ 4;

   if last.e744 then do;
*      ecnt + 1;
      if e744short in ('0910','0920','1010','1210') then do;
         if efulllong = '       ' then
            efulllong = substr(e744,1,5) || '80';
         if efullshort = '    ' then
            efullshort = substr(e744short,1,2) || '80';
      end;
      else if e744short in ('1110') then do;
```

```
         if efulllong = '       ' then
            efulllong = substr(e744,1,5) || '00';
         if efullshort = '    ' then
             efullshort = substr(e744short,1,2) || '00';
      end;
      output;
   end;

proc print data = rev.e744;
   title 'E- revised poststrata uncollapsed age/sex';

proc sort data = dmba01.pfullsam out = sortdatp;
   by p512;

data rev.p512 (keep = p64 p512 p512short p5var pcnt pfulllong pfullshort);
   set sortdatp;
      by p512;

   if last.p512;

      pcnt + 1;

proc print data = rev.p512;
   title 'P- revised poststrata uncollapsed age/sex';
run;

***************************** end get e744 and p512 **********************;

***************************** cross E744 and P512 ***********************;

data rev.e744p ;* (keep = p512 e744 twopost edom eten eage esex reg msa );
   set rev.e744;

   length edomc $1;
   length etenc $1;
   length eagec $1;
   length esexc $1;

   edomc = substr(e744,2,1);
   etenc = substr(e744,3,1);
   eagec = substr(e744,6,1);
   esexc = substr(e744,7,1);

   edom = input(edomc,1.);
   eten = input(etenc,1.);
   eage = input(eagec,1.);
   esex = input(esexc,1.);

   do region = 1 to 4;
      do pdom = 1 to 7;
         do pten = 1 to 2;
            do msatea = 1 to 4;
               do rratind = 0 to 1;
                  do page = 1 to 5;
                     do psex = 0 to 2;

      if edom = pdom and (eten = pten or eten = 0)
         and eage = page and esex = psex then do;
         length reg $1;
         length msa $1;
         length rrat $1;

         if pdom in (1,2,3,4,5,6) or (pdom = 7 and pten = 2) then
            reg = '5';
         else
            reg = put(region,1.);

         if pdom in (4,3) then do;
            if msatea in (1,2) then
               msa = '5';
            else if msatea in (3,4) then
               msa = '6';
         end;
         else if pdom in (5,6,1,2) then
            msa = '7';
         else if pdom = 7 then
            msa = put(msatea,1.);

         if pdom in (5,6,1,2) then
            rrat = '2';
         else
            rrat = put(rratind,1.);

          length pdomc $1;
          length ptenc $1;
          length pagec $1;
          length psexc $1;
          length p5var $5;
          length p512 $7;
          length twopost $15;

          pdomc = put(pdom,1.);
          ptenc = put(pten,1.);
          pagec = put(page,1.);
          psexc = put(psex,1.);

          p5var = trim(left(reg)) || trim(left(pdomc)) || trim(left(ptenc)) ||
trim(left(msa))
             || trim(left(rrat));

          p512 = trim(left(reg)) || trim(left(pdomc)) || trim(left(ptenc)) ||
trim(left(msa))
             || trim(left(rrat)) || trim(left(pagec)) || trim(left(psexc));

         twopost = e744 || ' ' || p512;
         output;
      end;

   end;
      end;
         end;
            end;
               end;
                  end;
                     end;
```

```
*proc sort data = rev.e744p nodupkey out = testep512;
*   by p512;

*proc sort data = rev.e744p nodupkey out = teste744p;
*   by e744;

*proc print data = testep512;
*   var p512;
*   title 'Are there 512 unique values of p512?';

*proc print data = teste744p;
*   var e744;
*   title 'Are there 744 unique values of e744?';


proc sort data = rev.e744p nodupkey;
   by twopost;


*proc sort data = rev.e744p nodupkey out = testep512;
*   by p512;

*proc sort data = rev.e744p nodupkey out = teste744p;
*   by e744;

*proc print data = testep512;
*   var p512;
*   title 'after nodupkey twopost: Are there 512 unique values of p512?';

*proc print data = teste744p;
*   var e744;
*   title 'after nodupkey twopost: Are there 744 unique values of e744?';

*proc freq data = rev.e744p;
*   tables p512 / list missing out = freqdat noprint;
*   title 'Are there 512 p512s? ';

*proc print data = freqdat;
*   sum count;
*   title 'e744p:  how many combinations of p5var';

proc sort data = rev.e744p;
   by p512;

*proc print data = rev.e744p (obs = 10);
*   var twopost p5var e5var p512;
*   title 'E-sample';

proc sort data = rev.p512;
   by p512;

*proc print data = rev.p512 (obs = 10);
*   var p512 p64 p5var;
*   title 'P-sample';

data e744pf1;
   merge rev.e744p rev.p512 ;
      by p512;

*   length twocolp $15;

*   twocolp = trim(left(substr(twopost,1,7))) || ' ' || pfulllong;
```

```
proc sort data = e744pf1;
   by e744;
*   by efullshort;

proc sort data = rev.e744;
   by e744;
*   by efullshort;

*proc print data = e744pf1 (obs = 500);
*   title 'is pfullshort present?';

data rev.e744pfin; * (drop = e525short e525long);
   merge e744pf1 (in = a) rev.e744 (in = b);
      by e744;
*      by efullshort;

   length efullshort2 $ 4 efulllong2 $ 7;

   if a or b then do;
      if e744short in ('0910', '0920','1010','1110','1210') then do;
         efullshort2 = substr(e744short,1,2) || '80';
         efulllong2 = substr(e744,1,5) || '80';
      end;
      else do;
         efullshort2 = '    ';
         efulllong2 = '       ';
      end;

      output rev.e744pfin;
   end;

run;


proc freq data = rev.e744pfin;
*   tables e744 e744short efullshort2 / missing ;
   tables efullshort efullshort2 / missing;
*   tables e744 / out = freqe744 noprint;
*   title 'how many efulllong2 not missing? should be 32';

*proc print data = freqe744;
*   sum count;
*   title 'how many p within each e?';

run;
```

zerocellC.sas ATTACHMENT 3

```
* zerocellC.sas :  this program used a file output from crosspostfin.sas e744pfin to
  assign a zero cell count to the Eexpand crossed Pexpand cells with no census
  people in them. i.e., for the Eexpand 0910, 1010, 1110, and 1210 there are no
  census people.  Instead of using fewer than 7584 cells, I'll impute 0's for these
  poststrata combinations.;

options ls = 80;
libname rev '/home/akearney/p1eval2k/reACE';

data zero (keep = e744short p512short Cstar IIstar DDstar ddla nddla C);
   set rev.e744pfin;

   if e744short in ('0910','1010','1110','1210');

      Cstar = 0;
      IIstar = 0;
      DDstar = 0;
      ddla = 0;
      nddla = 0;
      C = 0;

proc sort data = zero;
   by e744short p512short;

proc print data = zero;
   title 'zero';

data Cdat;
   infile '/home/akearney/p1eval2k/dmba01/reacecensum.dat' truncover;
   input @1 e744short $char4.
         @6 p512short $char4.
         @11 Cstar 10.
         @22 IIstar 10.
         @33 DDstar 10.
         @44 ddla 10.
         @55 nddla 10.
         @66 C 10.
         @77 ccfreace 17.15
         @95 dsereace 25.15  ;

proc freq data = zero;
   tables e744short p512short / missing list;
   title 'zero';

proc freq data = Cdat;
   tables e744short p512short / missing list;
   title 'Cdat';

proc contents data = zero;
proc contents data = Cdat;

proc sort data = Cdat;
   by e744short p512short;

data rev.Cdatfull;
   set zero Cdat;

   if p512short ne '0000' and e744short ne '0000';

proc sort data = rev.Cdatfull;
```

```
   by e744short p512short;

proc contents data = rev.Cdatfull;

proc sort data = rev.e744pfin out = e744pfin;
   by e744short p512short;

data mergeC inepfin inC;
   merge e744pfin (in = a) rev.Cdatfull (in = b);
      by e744short p512short;

   if a and b then
      output mergeC;
   else if a then
      output inepfin;
   else
      output inC;

proc print data = inepfin;
   title 'inepfin';

proc print data = inC;
   title 'inC';

run;
```

vector.sas **ATTACHMENT 4**

```
* vector.sas :  this program creates the (for e.g.) 128 vectors with P/M by ps listed
at the top of the vector and CE/E by ps at the bottom of the vector.  ;

options ls = 80 spool;
*options ls = 132;
libname rev2 '/vol/srdu8.files4/more-p1eval2k/reACE';
libname rev '/home/akearney/p1eval2k/reACE';
libname dmba01 '/home/akearney/p1eval2k/dmba01';

*  read Production output, need for C*, DD*, ddla, and nddla, see memo Q-38;
*  also need C (Cen).  ;

* this file has 7456 records.  will make the additional 128 have 0 entries.;

*   EEXPAND = e744  AND  PEXPAND = p512 ;

* cdatfull has 7584 records.  program zerocellC.sas added 128 records to file
  below and set the counts @11-@75 = 0.  No census people in these cells.;

* Create a vector of CCFs for each missing data alternative (128) at
  the new poststratum level.;

%macro vector(alt= ,num= ,ltr=, rep= );

%******************************** begin P-sample ***************************;

%* only need a few variables, but read entire file incase of needing to do
diagnostics;
/*
data p&alt.dat&num.&ltr (drop = wholerec);
   infile "/vol/srdu8.files4/more-p1eval2k/reACE/realt&alt.p&num.&ltr..dat"
   truncover lrecl=361;

   input   @1 fipsstat $char2.
                @3 region $char1.
                @4 div $char1.
                @5 aicind 1.
                @6 sampstr 1.
                @7 tea 1.
                @8 ccou $char3.
                @11 tract $char6.
                @17 cluster $char6.
                @23 block $char6.
                @29 lco $char4.
                @33 eclusurb 1.
                @34 tesindo 1.
                @35 ilmapspo $char5.
                @40 unitid $char4.
                @44 finoutc 2.
                @46 finouti 2.
                @48 cst $char2.
                @50 cou $char3.
                @54 perid $char4.
                @58 st $char2.
                @60 mid $char12.
                @72 mseq $char2.
                @74 mpnc $char5.
                @79 dtcode 2.
                @81 prepospc 2.
```

```
    @83 prenddc 2.
    @85 posspsc 2.
    @87 ndatdfc 2.
    @89 posspsi 2.
    @91 ndatdfi 2.
    @93 popct 2.
    @95 tenure 1.
    @96 toba 1.
    @97 age 3.
    @100 race 2.
    @102 sex 1.
    @103 hisp 1.
    @104 relate 2.
    @106 prxyin 1.
    @107 moverper 1.
    @108 moverhh 1.
    @109 rsc $char1.
    @113 cmpmat $char2.
    @115 clemat $char2.
    @117 bfumat $char2.
    @119 afumat $char2.
    @121 finmat $char2.
    @123 fuflag $char1.
    @124 mateind 1.
    @125 tesper 1.
    @126 wpinit 1.
    @127 wpfin 1.
    @128 addcde 1.
    @129 teswgt 7.4
@136 fu_code1 $char1.
@137 fu_code2 1.
    @146 tenure2 1.
    @147 age2 1.
    @148 race2 1.
    @150 hisp2 1.
    @151 rtype 1.
    @152 relat2 1.
    @153 tenifl 1.
    @154 ageifl 1.
    @155 sexifl 1.
    @156 hispif 1.
    @157 raceif 1.
    @158 amtimp 1.
    @159 bfugp 2.
    @161 frs 5.3
    @166 rprob 10.8
    @176 mprob 11.8
    @187 pwght 14.6
    @201 niwgto 15.6
    @216 niwgti 15.6
    @231 tesfinwt 15.6
    @246 fipsmsa $char4.
    @250 msasize $char1.
    @251 msaname $char60.
    @311 retrate 5.1
    @316 rrnum 10.
    @326 rrden 10.
    @336 domain 1.
    @337 cutoff 5.1
    @342 rratind 1.
    @343 msatea 1.
    @344 agesex 1.
    @345 poststr $char3.
```

```
                @348 cpoststr $char3.
                @351 newage $char3.
                @354 pfulllong $char7.
                @361 pfullshort $char4.
                @1 wholerec $char364.  ;

%* length wholerec = 22 for 0 person block clusters added to the file for
  variance estimation;

   if length(wholerec) ne 22;

%* assign weight to be used in Pnm, Pom, and Pim (wgtr) and Mnm and Mom (wgtm);
   if moverper in (1,3) then do;
      wgtr = tesfinwt * rprob;
      wgtfull = tesfinwt;
      wgtm = tesfinwt * rprob * mprob;
   end;
   else if moverper = 2 and rsc ne 'R' then do;
      wgtr = niwgti;
      wgtfull = niwgti;
      wgtm = 0;
   end;
   else do;
      wgtr = 0;
      wgtm = 0;
   end;

%* need to assign a 1/0 variable (mat) to sum weights for matches in proc freq;
%* mmat was used for verification purposes.;

   if finmat in ('M ',' M','MR','MU') then do;
      mmat = 'M ';
      mat = 1;
   end;
   else if finmat in ('NC','NP','NR','NU') then do;
      mmat = 'NM';
      mat = 0;
   end;
   else if finmat in ('KI','KP','P ',' P') then do;
      mmat = 'UR';
      mat = 1;
   end;
   else if finmat in ('DP','FP','GP','NL','NN','MN') then do;
      mmat = 'RP';
      mat = 0;
   end;
   else
      mmat = '**';

%* need to assign a 1/0 variable (res) to sum weights for residences in proc freq;
%* rmat, nonres, and unres was used for verification purposes.;

   if finmat in ('M ',' M','MR','NC','NP','NR') then do;
      rmat = 'R';
      res = 1;
      nonres = 0;
      unres = 0;
   end;
   else if finmat in ('MU','NU','KI','KP','P ',' P') then do;
      rmat = 'U';
%* make res = 1 for unresolved so they are included in weighted totals.;
      res = 1;
      nonres = 0;
      unres = 1;
   end;
   else if finmat in ('DP','FP','GP','NL','NN','MN') and moverper ne 2 then do;
      rmat = 'N';
      res = 0;     %*one person with moverper=2 and nonblank finmat ;
      nonres = 1;
      unres = 0;
   end;
   else if moverper = 2 then do;
      rmat = '*';
%* make res = 1 for inmovers so I can get the weighted number of inmovers;
      res = 1;
      nonres = 0;
      unres = 0;
   end;
   else do;
      rmat = '*';
      res = 0;
      nonres = 0;
      unres = 0;
   end;

proc sort data = p&alt.dat&num.&ltr;
   by pfulllong;

%* double check:  assign flag for when to use PES-C and when to use PES-A.;

data pesind (keep = pfulllong outmove pesind);
   set p&alt.dat&num.&ltr;
      by pfulllong;

   if moverper = 3 and rprob > 0 and tesfinwt > 0 then
      outmove + 1;

   if last.pfulllong then do;
      if outmove ge 10 then
         pesind = 'C';
      else
         pesind = 'A';
      output;
      outmove = 0;
   end;

proc sort data = p&alt.dat&num.&ltr;
   by moverper;

%* get weighted count of residents by mover status and collapsed poststrata.;

proc freq data = p&alt.dat&num.&ltr;
   by moverper;
   tables pfulllong * res / list missing out = freqdatr noprint;
   weight wgtr;
   title 'P-Sample ';

%* get weighted count of matches by mover status and collapsed poststrata.;

proc freq data = p&alt.dat&num.&ltr;
   by moverper;
   tables pfulllong * mat / list missing out = freqdatm noprint;
   weight wgtm;
   title 'P-Sample ';

%* create a separate file for each mover category so that I can get Pnm, Pom, Pim,
   etc. on the same record through the merge (below).;

data freqdatr1 (drop = count percent moverper moverper2 moverper3 Pim&rep Pom&rep)
     freqdatr2 (drop = count percent moverper moverper1 moverper3 Pnm&rep Pom&rep)
     freqdatr3 (drop = count percent moverper moverper1 moverper2 Pnm&rep Pim&rep);
   set freqdatr;

   if moverper = 1 then do;
      moverper1 = 1;
      Pnm&rep = count;
      output freqdatr1;
   end;
   else if moverper = 2 then do;
      moverper2 = 2;
      Pim&rep = count;
      output freqdatr2;
   end;
   else if moverper = 3 then do;
      moverper3 = 3;
      Pom&rep = count;
      output freqdatr3;
   end;

%* create a separate file for each mover category so that I can get Mnm, Mom,
   etc. on the same record through the merge (below).;

data freqdatm1 (drop = count percent moverper moverper2 moverper3 Mim&rep Mom&rep)
     freqdatm2 (drop = count percent moverper moverper1 moverper3 Mnm&rep Mom&rep)
     freqdatm3 (drop = count percent moverper moverper1 moverper2 Mnm&rep Mim&rep);
   set freqdatm;

   if moverper = 1 then do;
      moverper1 = 1;
      Mnm&rep = count;
      output freqdatm1;
   end;
   else if moverper = 2 then do;
      moverper2 = 2;
      Mim&rep = count;
      output freqdatm2;
   end;
   else if moverper = 3 then do;
      moverper3 = 3;
      Mom&rep = count;
      output freqdatm3;
   end;

proc sort data = freqdatr1;
   by pfulllong;

proc sort data = freqdatr2;
   by pfulllong;

proc sort data = freqdatr3;
   by pfulllong;

proc sort data = freqdatm1;
   by pfulllong;

%* freqdatm2 should be empty;

proc print data = freqdatm2;
   title1 "PROBLEM...PROBLEM...FREQDATM2 SHOULD BE EMPTY...PROBLEM...PROBLEM";
   title2 "samp = P alt = &alt num = &num ltr = &ltr rep = &rep";
title1 ' ';
title2 ' ';

%*proc sort data = freqdatm2;
%*   by newpost;

proc sort data = freqdatm3;
   by pfulllong;

proc sort data = pesind;
   by pfulllong;

%* freqdatm2 is an empty data set;

data rev.mergeP&rep (keep = pfulllong mrate&rep pcnt pesind)
     rev.mzero&rep (keep = pfulllong mrate&rep pcnt pesind);
   merge freqdatr1 (in = a) freqdatr2 (in = b)  freqdatr3 (in = c)
         freqdatm1 (in = d) freqdatm3 (in = e) pesind (in = f);
   by pfulllong;

*some poststrata are empty;

   pcnt + 1;

   if not a then do;
      Pnm&rep = 0;
      moverper1 = 1;
   end;

   if not b then do;
      Pim&rep = 0;
      moverper2 = 2;
   end;

   if not c then do;
      Pom&rep = 0;
      moverper3 = 3;
   end;

   if not d then do;
      Mnm&rep = 0;
      moverper1 = 1;
   end;

   if not e then do;
      Mom&rep = 0;
      moverper3 = 3;
   end;

   if pesind = 'C' then do;
      if pnm&rep ne 0 or pim&rep ne 0 then
         Mrate&rep = (Mnm&rep + (Mom&rep/Pom&rep) * Pim&rep)/(Pnm&rep + Pim&rep);
      else
         Mrate&rep = 0;
   end;
   else if pesind = 'A' then do;
      if pnm&rep ne 0 or pom&rep ne 0 then
         Mrate&rep =  (Mnm&rep + Mom&rep)/(Pnm&rep + Pom&rep) ;
      else
         Mrate&rep = 0;
   end;
   if Mrate&rep = 0 then
      output rev.mergeP&rep rev.mzero&rep;
   else
      output rev.mergeP&rep;

proc sort data = rev.p512;
   by pfulllong;

data rev.mergeP&rep ;
   merge rev.p512 (in = a) rev.mergeP&rep (in = b);
      by pfulllong;

   if a ;

*proc print data = rev.mergeP&rep (where=(Mrate&rep = 0));
*   var pfulllong p512 Mrate&rep ;
*   title 'should be none missing.';

proc datasets memtype = data ;
   delete p&alt.dat&num.&ltr;

%**************************************** end P-sample ******************;

%*************************************** begin E-sample *****************;

%* read E-sample person file;

data e&alt.dat&num.&ltr;
   infile "/vol/srdu8.files4/more-p1eval2k/reACE/realt&alt.e&num.&ltr..dat"
   truncover lrecl=336;
   input   @1 cst $char2.
                @3 region $char1.
                @4 div $char1.
                @5 aicind 1.
                @6 sampstr 1.
                @7 tea 1.
                @8 ccou $char3.
                @11 tract $char6.
                @17 cluster $char6.
                @23 block $char6.
                @29 lco $char4.
                @33 eclusurb 1.
                @34 tesind 1.
                @35 puid $char12.
                @47 pseq $char2.
                @49 pnc $char5.
                @54 mpsind $char1.
                @55 mmspot $char5.
                @60 munitid $char4.
                @64 mperid $char4.
                @68 dtcode 2.
                @70 nesamp 2.
                @72 inp 4.
                @76 rtenure 1.
                @77 page 3.
                @80 cpage 3.
                @83 race 2.
                @85 psex 1.
                @86 pspan 1.
                @87 prelsup 2.
                @89 numedup 1.
                @90 numdup 2.
                @92 nru 1.
```

```
    @93 cmpmat $char2.
    @95 clemat $char2.
    @97 bfumat $char2.
    @99 afumat $char2.
    @101 finmat $char2.
    @103 fuflag $char1.
    @104 tesper 1.
    @105 mateind 1.
    @106 wpinit 1.
    @107 wpfin 1.
    @108 addcde 1.
    @109 strcde 1.
    @110 ewght 14.6
    @124 fage 1.
    @125 frace 1.
    @126 ftenure 1.
    @127 fspan 1.
    @128 fsex 1.
    @129 frel 1.
    @130 teswgt 7.4
    @137 afnmathu $char2.
    @139 prcode 1.
    @140 mhumspot $char5.
    @145 mhuwmsid $char4.
    @149 st $char2.
    @151 cou $char3.
@154 fu_code1 $char1.
@155 fu_code2 1.
    @159 tenure2 1.
    @160 age2 1.
    @161 race2 1.
    @163 relate2 1.
    @164 tenifl 1.
    @165 ageifl 1.
    @166 sexifl 1.
    @167 hispif 1.
    @168 raceif 1.
    @169 amtimp 1.
    @170 bfugp 2.
    @172 fes 5.3
    @177 ceprobi 10.8
    @187 ceprobf 11.8
    @198 tesfinwt 14.6
    @212 fipsmsa $char4.
    @216 msasize $char1.
    @217 msaname $char60.
    @277 retrate 5.1
    @282 rrnum 10.
    @292 rrden 10.
    @302 domain 1.
    @303 cutoff 5.1
    @308 rratind 1.
    @309 msatea 1.
    @310 agesex 1.
    @311 poststr $char3.
    @314 cpoststr $char3.
    @317 srhhmem $char1.
    @318 spft $char2.
    @320 srsource $char2.
    @322 srepdate $char4.
    @326 efulllong $char7.
    @333 efullshort $char4.
    @1 wholerec $char336.  ;
```

```
%* if length(wholerec) = 22 then it is a 0 person block cluster only added for
variance
   estimation for original missing data evaluation (ESCAP II report 12).;

   if length(wholerec) ne 22;

%* assign indicator 1/0 for when a person should be counted as in E-sample (ine) or
when
   a person is a correct enumeration (ce) and correct weights.;

      ine = 1;
      wgtce = tesfinwt * ceprobf;
      wgte = tesfinwt;

*   newpost = cpoststr;

   if finmat in ('M ',' M','MR','CE') then do;
      ce = 1;
   end;
   else if finmat in ('DE','EE','FE','GE','KE','MN') then do;
      ce = 0;
   end;
   else if finmat in ('GU','MU','P ',' P','UE') then do;
      ce = 1;
   end;

%* estimate CEs by collapsed poststrata;

proc freq data = e&alt.dat&num.&ltr;
   tables efulllong * ce / list missing out = freqdatce noprint;
   weight wgtce;
   title 'E-sample:  CEs';

%* estimate E-sample people by collapsed poststrata;

proc freq data = e&alt.dat&num.&ltr;
   tables efulllong * ine / list missing out = freqdate noprint;
   weight wgte;
   title 'E-Sample:  Es';

%* rename freq output variable COUNT.;

data freqdate (drop = count percent);
  set freqdate;

   E&rep = count;

data freqdatce (drop = count percent);
   set freqdatce;

   CE&rep = count;

proc sort data = freqdate;
   by efulllong;

proc sort data = freqdatce;
   by efulllong;

%* put all E-sample estimates on one file.;

data rev.mergeE&rep (keep = efulllong CErate&rep flag ecnt);
   merge freqdate (in = a) freqdatce (in = b);
   by efulllong;
```

```
* using flag as a merging variable.;
   flag = 1;

   ecnt + 1;

%* guard against having an empty poststrata.  They would not have an
  entry in the freq output;

   if not a then
      E&rep = 0;

   if not b then
      CE&rep = 0;

   CErate&rep = (CE&rep/E&rep);

proc sort data = rev.e744;
   by efulllong;

data rev.mergeE&rep;
   merge rev.e744 (in = a) rev.mergeE&rep (in = b);
      by efulllong;

proc datasets memtype = data;
   delete e&alt.dat&num.&ltr;

%********************************** end E-sample *********************;
%************************** begin fimpute(k) calc *******************;

proc sort data = rev.e744pfin;
   by pfulllong;

proc sort data = rev.mergeP&rep;
   by pfulllong;

data epM pcnt (keep = em e m) eppost (keep = e744short p512short
   efullshort pfullshort);
   merge rev.e744pfin (in = a) rev.mergeP&rep (in = b) end = lastrec;
      by pfulllong;

   if a and b then
      em + 1;
   else if a then
      e + 1;
   else
      m + 1;

   if lastrec then
      output epM pcnt eppost;
   else
      output epM eppost;

proc print data = pcnt;
   title 'does em = 7584?';
run;

*proc sort data = eppost;
*   by e744short p512short;

*proc print data = eppost;
*   title 'eppost';

*data epMtemp;
```

```
*   set epM;

*   if substr(e744short,1,2) = '09';

*proc sort data = epMtemp;
*   by e744short p512short;

*proc print data = epMtemp;
*   var e744short p512short;
*   title 'epMtemp';

*proc print data = epM (obs = 500);
*   title 'epM:  what''s missing?';

proc sort data = epM;
   by efulllong;

proc sort data = rev.mergeE&rep;
   by efulllong;

*proc sort data = epM out = epMsort;
*   by efullshort;

*proc sort data = rev.mergeE&rep out = mergeE1sort;
*   by efullshort;

*proc print data = epMsort (obs = 500);
*   var e744short p512short efullshort pfullshort;
*   title 'epM';

*proc print data = mergeE&rep.sort (obs = 500);
*   var e744short efullshort;
*   title 'mergee1';

data epCE_M ecnt (keep = ec e c) eppost2 (keep = e744short p512short
   efullshort pfullshort);
   merge rev.mergeE&rep (in = b) epM (in = a) end = lastrec ;
      by efulllong;

   if a and b then do;
      ec + 1;
      output epCE_M eppost2;
   end;
   else if a then do;
      e + 1;
      output epCE_M eppost2;
   end;
   else
      c + 1;

   if lastrec then
      output ecnt;

proc print data = ecnt;
   title 'does ec = 7584? e and c = 0? ';
run;
*/
/*
proc sort data = eppost2;
   by e744short p512short;

data cdatfull (keep = e744short p512short);
   set rev.cdatfull;
```

```
proc sort data = cdatfull;
   by e744short p512short;

data merge12c ep1 ep2 c;
   merge eppost (in = a) eppost2 (in = b) cdatfull (in = c);
      by e744short p512short;

   if a and b and c then
      output merge12c;
   else if a then
      output ep1;
   else if b then
      output ep2;
   else
      output c;

proc print data = ep1;
   var e744short p512short efullshort pfullshort;
   title 'ep1';

proc print data = ep2;
   var e744short p512short efullshort pfullshort;
   title 'ep2';

proc print data = c;
   var e744short p512short efullshort pfullshort;
   title 'c';
*/
*proc print data = eppost2;
*   title 'eppost2';

*proc print data = epCE_M (obs = 500);
*   title 'epCE_M:  any missing?';

* want full 7584 on file.;

*proc sort data = testcol nodupkey out = rev.epCE_Mfin&rep;
*   by twopostcol;


proc sort data = epCE_M;
   by e744short p512short;

*proc contents data = epCE_M;
*proc contents data = rev.Cdatfull;

*proc freq data = epCE_M;
*   tables e744short p512short / missing list;
*   title 'epCE_M';

*proc freq data = rev.Cdatfull;
*   tables e744short p512short / missing list;
*   title 'Cdatfull';

proc sort data = rev.Cdatfull;
   by e744short p512short;

data rev.epc&rep cnt (keep = cnt7584 cntepc cntC) epcprob Cprob;
   merge rev.Cdatfull (in = a) epCE_M (in = b) end = lastrec;
      by e744short p512short;

   if a and b then do;
      cnt7584 + 1;
      output rev.epc&rep;
   end;
   else if b then do;
      cntepc + 1;
      output epcprob;
   end;
   else do;
      cntC + 1;
      output Cprob;
   end;

   if lastrec then
      output cnt;

proc print data = cnt;
   title 'totals?';

proc print data= epcprob;
   var e744short p512short ;
   title 'epcprob';

proc print data = Cprob;
   var e744short p512short;
   title 'Cprob';

*proc contents data = epcprob;
*proc contents data = Cprob;

run;
*/
data rev2.ccf&rep (keep = CErate&rep Mrate&rep ddstar cstar ddla nddla
                          e744 ecnt efulllong
                          efullshort revefullshort p512 pcnt pfulllong pfullshort
                          ccf&rep dse&rep e744short p512short);
   set rev.epc&rep;

   if cstar = 0 and ddla = 0 and nddla = 0 then do;
      ccf&rep = 0;
      dse&rep = 0;
   end;
   else do;
      ccf&rep = (ddstar/(cstar + ddla + nddla)) * (CErate&rep / Mrate&rep);

      dse&rep = (ddstar) * (CErate&rep / Mrate&rep);
   end;

*proc print data = rev2.ccf1;
*   title 'ccf1';

run;

%mend vector;
%*%vector(alt=1,num=1,ltr= ,rep=1);

* *****************************start ************************;
%* Don labeled original alternative output files with alt, num, and ltr.;
%* I add a rep number (1-128) for ease in using proc iml.;

%*vector(alt=1,num=1,ltr= ,rep=1);

%macro mac1;
%do n = 1 %to 16;
%let rep = &n;
```

```
%vector(alt=1,num=&n,ltr= ,rep=&rep);
%end;
%mend mac1;

%mac1;

%macro mac2;
%do n = 1 %to 16;
%let rep = %eval(&n + 16);
%vector(alt=3,num=&n,ltr= ,rep=&rep);
%end;
%mend mac2;

%mac2;

%macro mac3;

%let start = 33;

%do n = 1 %to 6;
   %if &n = 1 %then
      %let l = a;
   %else %if &n = 2 %then
      %let l = b;
   %else %if &n = 3 %then
      %let l = c;
   %else %if &n = 4 %then
      %let l = d;
   %else %if &n = 5 %then
      %let l = e;
   %else %if &n = 6 %then
      %let l = f;

   %let rep = &start;
   %vector(alt=4,num=1,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=4,num=2,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=4,num=4,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=4,num=5,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=4,num=7,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=4,num=9,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=4,num=12,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=4,num=16,ltr=&l ,rep=&rep);

   %let rep = %eval(&rep + 6);
   %vector(alt=5,num=3,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=5,num=6,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=5,num=8,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=5,num=10,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=5,num=11,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=5,num=13,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=5,num=14,ltr=&l ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %vector(alt=5,num=15,ltr=&l ,rep=&rep);

   %let start = %eval(&start + 1);

%end;
%mend mac3;

%mac3;

run;



/*
proc transpose data = rev.mergeP out=mergePT name = mrate&rep prefix = ps;
               id pcnt;

data mergePT;
   set mergePT;

   flag = 1;

proc print data = mergePT;
   title 'mergePT';

run;
*/
```

psamphighlow.sas **ATTACHMENT 5**

```
* psamphighlow.sas ;


options ls = 80 missing = ' ';
libname dmba01 '/home/akearney/p1eval2k/dmba01';

data pdseus;
   set dmba01.pdseus;

   if rsc = 'N' then
      rsv_v = 1;
   else if rsc = 'I' then
      rsc_v = 2;
   else if rsc = 'O' then
      rsc_v = 3;
   else if rsc = 'R' then
      rsc_v = 4;
   else if rsc = 'U' then
      rsc_v = 5;

   stratum = 10 * fipsstat + sampstr;

   if 0 < mprob < 1 then do;
      mprobhigh = 0;
      mproblow = 1;
   end;
   else do;
      mprobhigh = mprob;
      mproblow = mprob;
   end;

   if 0 < rprob < 1 then do;
      rprobhigh = 1;
      rproblow = 0;
   end;
   else do;
      rprobhigh = rprob;
      rproblow = rprob;
   end;

   if finmat in ('KI', 'KP', 'P ', ' P') then do; * unresolved match;
      mprobhigh2 = 0;
      mproblow2 = 1;
   end;
   else do;
      mprobhigh2 = mprob;
      mproblow2 = mprob;
   end;

   if finmat in ('MU','NU','KI','KP','P ',' P','IU') then do;
      rprobhigh2 = 1;
      rproblow2 = 0;
   end;
   else do;
      rprobhigh2 = rprob;
      rproblow2 = rprob;
   end;

proc format;
   value probfmt
   0 = '0'
   1 = '1'
   . = 'miss'
   other = 'unres';

proc print data = pdseus (where=(mprobhigh ne mprobhigh2 or
          mproblow ne mproblow2 or rprobhigh ne rprobhigh2 or
          rproblow ne rproblow2));
   var finmat mprobhigh mprobhigh2 mproblow mproblow2 rprob rprobhigh
          rprobhigh2 rproblow rproblow2;
   title 'pdseus';

/*
proc freq data = pdseus;
   tables mprob mprobhigh mprobhigh2 mproblow mproblow2 rprob rprobhigh
          rprobhigh2 rproblow rproblow2 / missing;
   format mprob mprobhigh mprobhigh2 mproblow mproblow2 rprob rprobhigh
          rprobhigh2 rproblow rproblow2 probfmt.;
   title 'Do high/low = high2/low2?';
*/
data _null_;
   set pdseus;
   file '/home/akearney/p1eval2k/reACE/pdseushigh.dat' lrecl = 367;
   put   @1 fipsstat $char2.
                @3 region $char1.
                @4 div $char1.
                @5 aicind 1.
                @6 sampstr 1.
                @7 tea 1.
                @8 ccou $char3.
                @11 tract $char6.
                @17 cluster $char6.
                @23 block $char6.
                @29 lco $char4.
                @33 eclusurb 1.
                @34 tesindo 1.
                @35 ilmapspo $char5.
                @40 unitid $char4.
                @44 finoutc 2.
                @46 finouti 2.
                @48 cst $char2.
                @50 cou $char3.
                @54 perid $char4.
                @58 st $char2.
                @60 mid $char12.
                @72 mseq $char2.
                @74 mpnc $char5.
                @79 dtcode 2.
                @81 prepospc 2.
                @83 prenddc 2.
                @85 posspsc 2.
                @87 ndatdfc 2.
                @89 posspsi 2.
                @91 ndatdfi 2.
                @93 popct 2.
                @95 tenure 1.
                @96 toba 1.
                @97 age 3.
                @100 race 2.
                @102 sex 1.
                @103 hisp 1.
                @104 relate 2.
                @106 prxyin 1.
                @107 moverper 1.
                @108 moverhh 1.
                @109 rsc $char1.
                @113 cmpmat $char2.
                @115 clemat $char2.
                @117 bfumat $char2.
                @119 afumat $char2.
                @121 finmat $char2.
                @123 fuflag $char1.
                @124 mateind 1.
                @125 tesper 1.
                @126 wpinit 1.
                @127 wpfin 1.
                @128 addcde 1.
                @129 teswgt 7.4
            @136 fu_code1 $char1.
            @137 fu_code2 1.
                @146 tenure2 1.
                @147 age2 1.
                @148 race2 1.
                @150 hisp2 1.
                @151 rtype 1.
                @152 relat2 1.
                @153 tenifl 1.
                @154 ageifl 1.
                @155 sexifl 1.
                @156 hispif 1.
                @157 raceif 1.
                @158 amtimp 1.
                @159 bfugp 2.
                @161 frs 5.3
                @166 rprobhigh 10.8
                @176 mprobhigh 11.8
                @187 pwght 14.6
                @201 niwgto 15.6
                @216 niwgti 15.6
                @231 tesfinwt 15.6
                @246 fipsmsa $char4.
                @250 msasize $char1.
                @251 msaname $char60.
                @311 retrate 5.1
                @316 rrnum 10.
                @326 rrden 10.
                @336 domain 1.
                @337 cutoff 5.1
                @342 rratind 1.
                @343 msatea 1.
                @344 agesex 1.
                @345 poststr $char3.
                @348 cpoststr $char3.
                @364 rsc_v 1.
                @365 stratum 3.       ;


data _null_;
   set pdseus;
   file '/home/akearney/p1eval2k/reACE/pdseuslow.dat' lrecl = 367;
   put   @1 fipsstat $char2.
                @3 region $char1.
                @4 div $char1.
                @5 aicind 1.
                @6 sampstr 1.
                @7 tea 1.
                @8 ccou $char3.
```

```
        @11 tract $char6.
        @17 cluster $char6.
        @23 block $char6.
        @29 lco $char4.
        @33 eclusurb 1.
        @34 tesindo 1.
        @35 ilmapspo $char5.
        @40 unitid $char4.
        @44 finoutc 2.
        @46 finouti 2.
        @48 cst $char2.
        @50 cou $char3.
        @54 perid $char4.
        @58 st $char2.
        @60 mid $char12.
        @72 mseq $char2.
        @74 mpnc $char5.
        @79 dtcode 2.
        @81 prepospc 2.
        @83 prenddc 2.
        @85 posspsc 2.
        @87 ndatdfc 2.
        @89 posspsi 2.
        @91 ndatdfi 2.
        @93 popct 2.
        @95 tenure 1.
        @96 toba 1.
        @97 age 3.
        @100 race 2.
        @102 sex 1.
        @103 hisp 1.
        @104 relate 2.
        @106 prxyin 1.
        @107 moverper 1.
        @108 moverhh 1.
        @109 rsc $char1.
        @113 cmpmat $char2.
        @115 clemat $char2.
        @117 bfumat $char2.
        @119 afumat $char2.
        @121 finmat $char2.
        @123 fuflag $char1.
        @124 mateind 1.
        @125 tesper 1.
        @126 wpinit 1.
        @127 wpfin 1.
        @128 addcde 1.
        @129 teswgt 7.4
    @136 fu_code1 $char1.
    @137 fu_code2 1.
        @146 tenure2 1.
        @147 age2 1.
        @148 race2 1.
        @150 hisp2 1.
        @151 rtype 1.
        @152 relat2 1.
        @153 tenifl 1.
        @154 ageifl 1.
        @155 sexifl 1.
        @156 hispif 1.
        @157 raceif 1.
        @158 amtimp 1.
        @159 bfugp 2.
```

```
                @161 frs 5.3
                @166 rproblow 10.8
                @176 mproblow 11.8
                @187 pwght 14.6
                @201 niwgto 15.6
                @216 niwgti 15.6
                @231 tesfinwt 15.6
                @246 fipsmsa $char4.
                @250 msasize $char1.
                @251 msaname $char60.
                @311 retrate 5.1
                @316 rrnum 10.
                @326 rrden 10.
                @336 domain 1.
                @337 cutoff 5.1
                @342 rratind 1.
                @343 msatea 1.
                @344 agesex 1.
                @345 poststr $char3.
                @348 cpoststr $char3.
                @364 rsc_v 1.
                @365 stratum 3.             ;

run;

/********* BEGIN REVISED METHODOLOGY REVISED POSTSTRATA *************/

data pfullsam;
infile '/home/akearney/p1eval2k/dmba01/pfullsam.dat' missover lrecl = 514;
input   @17 cluster $char6.
        @35 msn $char5.
        @40 wmsn $char4.
        @54 per $char4.
        @121 finmat $char2.
        @166 rprob 10.8
        @176 mprob 11.8
        @362 pfullshort $char4.
        @362 pfulld12 $char2.
        @364 pfulld3 $char1.
        @365 pfulld4 $char1.
        @505 pt1 10.8;

   if 0 < mprob < 1 then do;
      mprobhigh = 0;
      mproblow = 1;
   end;
   else do;
      mprobhigh = mprob;
      mproblow = mprob;
   end;

   if 0 < rprob < 1 then do;
      rprobhigh = 1;
      rproblow = 0;
   end;
   else do;
      rprobhigh = rprob;
      rproblow = rprob;
   end;

   if finmat in ('KI', 'KP', 'P ', ' P') then do; * unresolved match;
      mprobhigh2 = 0;
      mproblow2 = 1;
   end;
   else do;
      mprobhigh2 = mprob;
      mproblow2 = mprob;
   end;

   if finmat in ('MU','NU','KI','KP','P ',' P','IU') then do;
      rprobhigh2 = 1;
      rproblow2 = 0;
   end;
   else do;
      rprobhigh2 = rprob;
      rproblow2 = rprob;
   end;

proc format;
   value probfmt
   0 = '0'
   1 = '1'
   . = 'miss'
   other = 'unres';

proc print data = pfullsam (where=(mprobhigh ne mprobhigh2 or
          mproblow ne mproblow2 or rprobhigh ne rprobhigh2 or
          rproblow ne rproblow2));
   var finmat mprobhigh mprobhigh2 mproblow mproblow2 rprob rprobhigh
          rprobhigh2 rproblow rproblow2;
   title 'pfullsam';

data _null_;
   set pfullsam;
   file '/home/akearney/p1eval2k/reACE/pfullsamhigh.dat' lrecl = 514;
   put   @17 cluster $char6.
        @35 msn $char5.
        @40 wmsn $char4.
        @54 per $char4.
        @121 finmat $char2.
        @166 rprobhigh 10.8
        @176 mprobhigh 11.8
        @362 pfullshort $char4.
        @362 pfulld12 $char2.
        @364 pfulld3 $char1.
        @365 pfulld4 $char1.
        @505 pt1 10.8;

data _null_;
   set pfullsam;
   file '/home/akearney/p1eval2k/reACE/pfullsamlow.dat' lrecl = 514;
   put   @17 cluster $char6.
        @35 msn $char5.
        @40 wmsn $char4.
        @54 per $char4.
        @121 finmat $char2.
        @166 rproblow 10.8
        @176 mproblow 11.8
        @362 pfullshort $char4.
        @362 pfulld12 $char2.
        @364 pfulld3 $char1.
        @365 pfulld4 $char1.
        @505 pt1 10.8;

run;
```

esamphighlow.sas **ATTACHMENT 6**

```
options ls = 80 missing = ' ';
libname dmba01 '/home/akearney/p1eval2k/dmba01';

/************** this code is for the high/low estimates using production
                DSE methodology and production
                poststrata

                We ended up not using the half-high, half-low estimates.
               ************************************************/

data edseus;
   set dmba01.edseus;

   if 0 < ceprobi < 1 then do;
      ceprobihigh = 1;
      ceprobilow = 0;
   end;
   else do;
      ceprobihigh = ceprobi;
      ceprobilow = ceprobi;
   end;

   ceprobfhigh = ceprobihigh * (1 + numedup) / (1 + numedup + numdup) ;
   ceprobflow = ceprobilow * (1 + numedup) / (1 + numedup + numdup);

* if numdup > 0 then ceprobf not in (1,0) ;

   if finmat in ('GU','MU','P ',' P','UE') and (bfumat ne 'KE') then do;
      ceprobihigh2 = 1;
      ceprobilow2 = 0;
      class = 'neKE';
   end;
   else if finmat in ('GU','MU','P ',' P','UE') and (bfugp = 10) then do;
      ceprobihigh2 = 1;
      ceprobilow2 = 0;
      class = 'gp10';
   end;
   else do;
      ceprobihigh2 = ceprobi;
      ceprobilow2 = ceprobi;
      class = 'othr';
   end;

   ceprobfhigh2 = ceprobihigh2 * (1 + numedup) / (1 + numedup + numdup) ;
   ceprobflow2 = ceprobilow2 * (1 + numedup) / (1 + numedup + numdup);

proc format;
   value probfmt
   0 = '0'
   1 = '1'
   . = 'miss'
   other = 'unres';
/*
proc freq data = edseus (where=(class='othr' and finmat in ('GU','MU',
                         'P ',' P','UE')));
*   tables class / missing ;
   tables class bfugp bfumat finmat ceprobihigh2 ceprobilow2 ceprobi
          ceprobfhigh2 ceprobflow2 / missing list ;
   format ceprobihigh2 ceprobilow2 ceprobi ceprobfhigh2 ceprobflow2 probfmt.;
   title 'edseus:  unresolved ceprobi';
```

```
proc freq data = edseus;* (where=(0<ceprobfhigh2<1));
   tables ceprobi ceprobihigh ceprobihigh2 ceprobilow
          ceprobilow2 ceprobf ceprobfhigh ceprobfhigh2 ceprobflow
          ceprobflow2 / missing ;
   format ceprobi ceprobihigh ceprobihigh2 ceprobilow
          ceprobilow2 ceprobf ceprobfhigh ceprobfhigh2 ceprobflow
          ceprobflow2 probfmt.;
*   tables finmat / missing;
   title 'edseus high and low';
*/
data _null_;
   set edseus ;
   file '/home/akearney/p1eval2k/reACE/edseushigh.dat' lrecl = 316;
   put   @1 cst $char2.
                @3 region $char1.
                @4 div $char1.
                @5 aicind 1.
                @6 sampstr 1.
                @7 tea 1.
                @8 ccou $char3.
                @11 tract $char6.
                @17 cluster $char6.
                @23 block $char6.
                @29 lco $char4.
                @33 eclusurb 1.
                @34 tesind 1.
                @35 puid $char12.
                @47 pseq $char2.
                @49 pnc $char5.
                @54 mpsind $char1.
                @55 mmspot $char5.
                @60 munitid $char4.
                @64 mperid $char4.
                @68 dtcode 2.
                @70 nesamp 2.
                @72 inp 4.
                @76 rtenure 1.
                @77 page 3.
                @80 cpage 3.
                @83 race 2.
                @85 psex 1.
                @86 pspan 1.
                @87 prelsup 2.
                @89 numedup 1.
                @90 numdup 2.
                @92 nru 1.
                @93 cmpmat $char2.
                @95 clemat $char2.
                @97 bfumat $char2.
                @99 afumat $char2.
                @101 finmat $char2.
                @103 fuflag $char1.
                @104 tesper 1.
                @105 mateind 1.
                @106 wpinit 1.
                @107 wpfin 1.
                @108 addcde 1.
                @109 strcde 1.
                @110 ewght 14.6
                @124 fage 1.
                @125 frace 1.
                @126 ftenure 1.
                @127 fspan 1.
                @128 fsex 1.
                @129 frel 1.
                @130 teswgt 7.4
                @137 afnmathu $char2.
                @139 prcode 1.
                @140 mhumspot $char5.
                @145 mhuwmsid $char4.
                @149 st $char2.
                @151 cou $char3.
            @154 fu_code1 $char1.
            @155 fu_code2 1.
                @159 tenure2 1.
                @160 age2 1.
                @161 race2 1.
                @163 relate2 1.
                @164 tenifl 1.
                @165 ageifl 1.
                @166 sexifl 1.
                @167 hispif 1.
                @168 raceif 1.
                @169 amtimp 1.
                @170 bfugp 2.
                @172 fes 5.3
                @177 ceprobihigh 10.8
                @187 ceprobfhigh 11.8
                @198 tesfinwt 14.6
                @212 fipsmsa $char4.
                @216 msasize $char1.
                @217 msaname $char60.
                @277 retrate 5.1
                @282 rrnum 10.
                @292 rrden 10.
                @302 domain 1.
                @303 cutoff 5.1
                @308 rratind 1.
                @309 msatea 1.
                @310 agesex 1.
                @311 poststr $char3.
                @314 cpoststr $char3.     ;

run;

data _null_;
   set edseus ;
   file '/home/akearney/p1eval2k/reACE/edseuslow.dat' lrecl = 316;
   put   @1 cst $char2.
                @3 region $char1.
                @4 div $char1.
                @5 aicind 1.
                @6 sampstr 1.
                @7 tea 1.
                @8 ccou $char3.
                @11 tract $char6.
                @17 cluster $char6.
                @23 block $char6.
                @29 lco $char4.
                @33 eclusurb 1.
                @34 tesind 1.
                @35 puid $char12.
                @47 pseq $char2.
                @49 pnc $char5.
                @54 mpsind $char1.
                @55 mmspot $char5.
```

```
    @60 munitid $char4.
    @64 mperid $char4.
    @68 dtcode 2.
    @70 nesamp 2.
    @72 inp 4.
    @76 rtenure 1.
    @77 page 3.
    @80 cpage 3.
    @83 race 2.
    @85 psex 1.
    @86 pspan 1.
    @87 prelsup 2.
    @89 numedup 1.
    @90 numdup 2.
    @92 nru 1.
    @93 cmpmat $char2.
    @95 clemat $char2.
    @97 bfumat $char2.
    @99 afumat $char2.
    @101 finmat $char2.
    @103 fuflag $char1.
    @104 tesper 1.
    @105 mateind 1.
    @106 wpinit 1.
    @107 wpfin 1.
    @108 addcde 1.
    @109 strcde 1.
    @110 ewght 14.6
    @124 fage 1.
    @125 frace 1.
    @126 ftenure 1.
    @127 fspan 1.
    @128 fsex 1.
    @129 frel 1.
    @130 teswgt 7.4
    @137 afnmathu $char2.
    @139 prcode 1.
    @140 mhumspot $char5.
    @145 mhuwmsid $char4.
    @149 st $char2.
    @151 cou $char3.
@154 fu_code1 $char1.
@155 fu_code2 1.
    @159 tenure2 1.
    @160 age2 1.
    @161 race2 1.
    @163 relate2 1.
    @164 tenifl 1.
    @165 ageifl 1.
    @166 sexifl 1.
    @167 hispif 1.
    @168 raceif 1.
    @169 amtimp 1.
    @170 bfugp 2.
    @172 fes 5.3
    @177 ceprobilow 10.8
    @187 ceprobflow 11.8
    @198 tesfinwt 14.6
    @212 fipsmsa $char4.
    @216 msasize $char1.
    @217 msaname $char60.
    @277 retrate 5.1
    @282 rrnum 10.
```

```
                @292 rrden 10.
                @302 domain 1.
                @303 cutoff 5.1
                @308 rratind 1.
                @309 msatea 1.
                @310 agesex 1.
                @311 poststr $char3.
                @314 cpoststr $char3.     ;

run;

/*    ********** END PRODUCTION METHODOLOGY, PRODUCTION POSTSTRATA ****/

/*    ********* BEGIN REVISED DSE, REVISED METHODOLOGY **************/

data efullsam;
   infile '/home/akearney/p1eval2k/dmba01/efullsam.dat' lrecl = 491;
   input   @17 cluster $char6.
        @35 puid $char12.
        @47 pseq $char2.
        @49 pnc $char5.
        @55 msn $char5.
        @60 wmsn $char4.
        @64 per $char4.
        @77 age 3.
        @85 sex 1.
        @89 numedup 1.
        @90 numdup 2.
        @97 bfumat $char2.
        @101 finmat $char2.
        @170 bfugp 2.
        @177 ceprobi 10.8
        @187 ceprobf 11.8
        @198 tesfinwt 14.6
        @390 efulllong $char7.
        @397 efull12d $char2.
        @399 efull3d $char1.
        @400 efull4d $char1.
        @397 efullshort $char4.
        @452 zt1 10.8
        @462 zt2 10.8
        @472 pt1 10.8
        @482 pt2 10.8;

   if 0 < ceprobi < 1 then do;
      ceprobihigh = 1;
      ceprobilow = 0;
   end;
   else do;
      ceprobihigh = ceprobi;
      ceprobilow = ceprobi;
   end;

   ceprobfhigh = ceprobihigh * (1 + numedup) / (1 + numedup + numdup) ;
   ceprobflow = ceprobilow * (1 + numedup) / (1 + numedup + numdup);

* if numdup > 0 then ceprobf not in (1,0) ;

   if finmat in ('GU','MU','P ',' P','UE') and (bfumat ne 'KE') then do;
      ceprobihigh2 = 1;
      ceprobilow2 = 0;
      class = 'neKE';
   end;
   else if finmat in ('GU','MU','P ',' P','UE') and (bfugp = 10) then do;
      ceprobihigh2 = 1;
      ceprobilow2 = 0;
      class = 'gp10';
   end;
   else do;
      ceprobihigh2 = ceprobi;
      ceprobilow2 = ceprobi;
      class = 'othr';
   end;

   ceprobfhigh2 = ceprobihigh2 * (1 + numedup) / (1 + numedup + numdup) ;
   ceprobflow2 = ceprobilow2 * (1 + numedup) / (1 + numedup + numdup);

proc format;
   value probfmt
   0 = '0'
   1 = '1'
   . = 'miss'
   other = 'unres';

proc print data = efullsam (where=(ceprobihigh ne ceprobihigh2
                                 or ceprobilow ne ceprobilow2));
   var puid pseq pnc ceprobi numedup numdup ceprobihigh
       ceprobihigh2 ceprobilow ceprobilow2 ceprobi finmat bfumat;
   title 'efullsam:  ceprobihigh ne ceprobhigh2 or ceprobilow ne ceprobilow2';

data _null_;
   set efullsam ;
   file '/home/akearney/p1eval2k/reACE/efullsamhigh.dat' lrecl = 491;
   put   @17 cluster $char6.
        @35 puid $char12.
        @47 pseq $char2.
        @49 pnc $char5.
        @55 msn $char5.
        @60 wmsn $char4.
        @64 per $char4.
        @77 age 3.
        @85 sex 1.
        @89 numedup 1.
        @90 numdup 2.
        @97 bfumat $char2.
        @101 finmat $char2.
        @170 bfugp 2.
        @177 ceprobihigh 10.8
        @187 ceprobfhigh 11.8
        @198 tesfinwt 14.6
        @390 efulllong $char7.
        @397 efull12d $char2.
        @399 efull3d $char1.
        @400 efull4d $char1.
        @397 efullshort $char4.
        @452 zt1 10.8
        @462 zt2 10.8
        @472 pt1 10.8
        @482 pt2 10.8;

data _null_;
   set efullsam ;
   file '/home/akearney/p1eval2k/reACE/efullsamlow.dat' lrecl = 491;
   put   @17 cluster $char6.
        @35 puid $char12.
        @47 pseq $char2.
        @49 pnc $char5.
        @55 msn $char5.
        @60 wmsn $char4.
        @64 per $char4.
        @77 age 3.
        @85 sex 1.
        @89 numedup 1.
        @90 numdup 2.
        @97 bfumat $char2.
        @101 finmat $char2.
        @170 bfugp 2.
        @177 ceprobilow 10.8
        @187 ceprobflow 11.8
        @198 tesfinwt 14.6
        @390 efulllong $char7.
        @397 efull12d $char2.
        @399 efull3d $char1.
        @400 efull4d $char1.
        @397 efullshort $char4.
        @452 zt1 10.8
        @462 zt2 10.8
        @472 pt1 10.8
        @482 pt2 10.8;

run;
```

psampreace.sas

```
* psampreace.sas:  this program modifies mprob and rprob to get both the
  highest DSE and the lowest DSE, and also the halfhigh and halflow.  There
   is one output file, the sas dataset pvmdplus.  ;
* I calculate the high/low probs 2 ways to make sure there is agreement.
  There should be the same number of people with bestf2 = unres. and mprob
  (or rprob) between 0 and 1.;

option ls = 80 missing = ' ';
libname dmba01 '/home/akearney/p1eval2k/dmba01';
libname revmd '/home/akearney/p1eval2k/reACE';

data prevsam;
   infile '/home/akearney/p1eval2k/dmba01/prevsam.dat' truncover lrecl = 1119;
   input   @1 clust $char6.
        @7 msn $char5.
        @12 wmsn $char4.
        @16 per $char4.
        @101 sex 1.
        @106 bmovper 1.
        @114 bfinmat $char2.
        @178 pmprob 11.8
        @203 prprob 10.8
        @287 agesex 1.
        @341 bacestat $char1.
        @709 code_final $char2.
        @717 rsc_final $char1.
        @718 moverper_final 1.
        @719 mprob_final 11.8
        @730 niwgti_final 15.6
        @760 tesfinwt_final 15.6
        @775 rprob_final 10.8
        @785 iprob_final 10.8
        @800 tesfinwt_new 15.6
        @1117 newage 3.;

* 399 people have code_final = NU and rsc_final = I.  they have rprob=0
  and mprob = 0;

   if code_final in ('KI','KP','P ',' P','NU','MU') and
      rsc_final ne 'I' then do;
      rprobhigh = 1;
      rproblow = 0;
      rprobhalfhigh = rprob_final + (1 - rprob_final)/2 ;
      rprobhalflow = rprob_final/2;
   end;
   else do;
      rprobhigh = rprob_final;
      rproblow = rprob_final;
      rprobhalfhigh = rprob_final;
      rprobhalflow = rprob_final;
   end;

   if 0 < rprob_final < 1 and code_final not in ('N ',' N')then do;
      rprobhigh2 = 1;
      rproblow2 = 0;
      rprobhalfhigh2 = rprob_final + (1 - rprob_final)/2 ;
      rprobhalflow2 = rprob_final/2;
   end;
   else do;
      rprobhigh2 = rprob_final;
      rproblow2 = rprob_final;
      rprobhalfhigh2 = rprob_final;
      rprobhalflow2 = rprob_final;
   end;

   if 0 < prprob < 1 then do;
      prprobhigh = 1;
      prproblow = 0;
   end;
   else do;
      prprobhigh = prprob;
      prproblow = prprob;
   end;

   if bfinmat in ('MU','NU','KI','KP','P ',' P','IU') then do;
      prprobhigh2 = 1;
      prproblow2 = 0;
   end;
   else do;
      prprobhigh2 = prprob;
      prproblow2 = prprob;
   end;
***************** extreme mprob section ***************************;

   if 0 < pmprob < 1 then do;
      pmprobhigh = 0;
      pmproblow = 1;
   end;
   else do;
      pmprobhigh = pmprob;
      pmproblow = pmprob;
   end;

   if bfinmat in ('KI', 'KP', 'P ', ' P') then do; * unresolved match;
      pmprobhigh2 = 0;
      pmproblow2 = 1;
   end;
   else do;
      pmprobhigh2 = pmprob;
      pmproblow2 = pmprob;
   end;

   if code_final in ('KI', 'KP', 'P ', ' P') then do; * unresolved match;
      mprobhigh = 0;
      mproblow = 1;
      mprobhalfhigh = mprob_final/2 ;
      mprobhalflow = mprob_final + (1 - mprob_final)/2;
   end;
   else do;
      mprobhigh = mprob_final;
      mproblow = mprob_final;
      mprobhalfhigh = mprob_final;
      mprobhalflow = mprob_final;
   end;

   if 0 < mprob_final < 1 then do;
      mprobhigh2 = 0;
      mproblow2 = 1;
      mprobhalfhigh2 = mprob_final/2;
      mprobhalflow2 = mprob_final + (1 - mprob_final)/2;
   end;
   else do;
      mprobhigh2 = mprob_final;
      mproblow2 = mprob_final;
      mprobhalfhigh2 = mprob_final;
      mprobhalflow2 = mprob_final;
   end;
/*
proc print data = prevsam (where=(mprobhigh ne mprobhigh2
                                 or mproblow ne mproblow2
                                 or mprobhalfhigh ne mprobhalfhigh2
                                 or mprobhalflow ne mprobhalflow2
                                 or pmprobhigh ne pmprobhigh2
                                 or pmproblow ne pmproblow2));
   var clust msn wmsn per mprob_final code_final mprobhigh
       mprobhigh2 mproblow mproblow2 mprobhalfhigh mprobhalfhigh2
       mprobhalflow mprobhalflow2;
   title 'prevsam:  mprobhigh ne mprobhigh2 mproblow ne mproblow2
        mprobhalfhigh ne mprobhalfhigh2 mprobhalflow ne mprobhalflow2';

proc print data = prevsam (where=(rprobhigh ne rprobhigh2
                                 or rproblow ne rproblow2
                                 or rprobhalfhigh ne rprobhalfhigh2
                                 or rprobhalflow ne rprobhalflow2
                                 or prprobhigh ne prprobhigh2
                                 or prproblow ne prproblow2));
   var clust msn wmsn per rprob_final code_final rprobhigh
       rprobhigh2 rproblow rproblow2 rprobhalfhigh rprobhalfhigh2
       rprobhalflow rprobhalflow2;
   title 'prevsam:  rprobhigh ne rprobhigh2 rproblow ne rproblow2
        rprobhalfhigh ne rprobhalfhigh2 rprobhalflow ne rprobhalflow2';

data _null_;
   set prevsam;
   file '/home/akearney/p1eval2k/reACE/prevsamhigh.dat' lrecl = 1119;
   put   @1 clust $char6.
        @7 msn $char5.
        @12 wmsn $char4.
        @16 per $char4.
        @101 sex 1.
        @106 bmovper 1.
        @114 bfinmat $char2.
        @178 pmprobhigh 11.8
        @203 prprobhigh 10.8
        @287 agesex 1.
        @341 bacestat $char1.
        @709 code_final $char2.
        @717 rsc_final $char1.
        @718 moverper_final 1.
        @719 mprobhigh 11.8
        @730 niwgti_final 15.6
        @760 tesfinwt_final 15.6
        @775 rprobhigh 10.8
        @785 iprob_final 10.8
        @800 tesfinwt_new 15.6
        @1117 newage 3.;
*/
data _null_;
   set prevsam;
   file '/home/akearney/p1eval2k/reACE/prevsamlow.dat' lrecl = 1119;
   put   @1 clust $char6.
        @7 msn $char5.
        @12 wmsn $char4.
        @16 per $char4.
        @101 sex 1.
        @106 bmovper 1.
        @114 bfinmat $char2.
        @178 pmproblow 11.8
        @203 prproblow 10.8
        @287 agesex 1.
        @341 bacestat $char1.
        @709 code_final $char2.
        @717 rsc_final $char1.
        @718 moverper_final 1.
        @719 mproblow 11.8
        @730 niwgti_final 15.6
        @760 tesfinwt_final 15.6
        @775 rproblow 10.8
        @785 iprob_final 10.8
        @800 tesfinwt_new 15.6
        @1117 newage 3.;

run;
```

```
* esampreace.sas:  this program modifies ceprobi to get both the highest DSE
  and the lowest DSE, and also the halfhigh and halflow.  There are 4 output
  files.;
* I calculate the high/low probs 2 ways to make sure there is agreement.
  There should be the same number of people with bestf = unres. and ceprobi
  between 0 and 1.  I do a check since there was a slight variation in
  production.;
* the file erevsam.sas7bdat and prevsam.sas7bdat are the same as files
  emd_final.sas7bdat and pmd_final.sas7bdat.  Dawn copied the files from
  ace2kres:[reace.md] to estimation$disk:[revace.missingdata] and renamed
  them.;

option ls = 80 missing = ' ';
libname dmba01 '/home/akearney/p1eval2k/dmba01';
libname revmd '/home/akearney/p1eval2k/reACE';

data erevsam;
   infile '/home/akearney/p1eval2k/dmba01/erevsam.dat' truncover lrecl = 642;
   input @1 clust $char6. @7 cid $char12. @19 seq $char2. @21 per $char5.
         @35 bbfumat $char2. @416 age 3. @421 sex 1. @425 bdupcnt1 1.
         @426 bdupcnt2 2.
         @433 bfinmat $char2. @509 pceprobi 10.8 @519 pceprobf 11.8
         @592 ceprobi_final @602 ceprobf_final 11.8 @613 tesfinwt_final 15.6
         @628 tesfinwt_new 15.6;

   if code_final in ('P ',' P','UE') then do;
      ceprobihigh = 1;
      ceprobilow = 0;
      ceprobihalfhigh = ceprobi_final + (1 - ceprobi_final)/2 ;
      ceprobihalflow = ceprobi_final/2;
   end;
   else do;
      ceprobihigh = ceprobi_final;
      ceprobilow = ceprobi_final;
      ceprobihalfhigh = ceprobi_final;
      ceprobihalflow = ceprobi_final;
   end;
/*  still one unresolved pceprobi (btwn 0 and 1) with bfinmat=UE and
bbfumat=KE
    this case was treated uniquely.  use classification below.

   if bfinmat in (' P','P ','NE','MU','UE','GU') and bbfumat ne 'KE' then do;
      pceprobihigh = 1;
      pceprobilow = 0;
      pceprobihalfhigh = pceprobi + (1 - pceprobi)/2;
      pceprobihalflow = pceprobi / 2;
   end;
   else do;
      pceprobihigh = pceprobi;
      pceprobilow = pceprobi;
      pceprobihalfhigh = pceprobi;
      pceprobihalflow = pceprobi;
   end;
*/
   ceprobfhigh = ceprobihigh * (1 + bdupcnt1)/(1 + bdupcnt1 + bdupcnt2);
   ceprobflow = ceprobilow * (1 + bdupcnt1)/(1 + bdupcnt1 + bdupcnt2);
   ceprobfhalfhigh = ceprobihalfhigh * (1 + bdupcnt1)/(1 + bdupcnt1 +
bdupcnt2);
   ceprobfhalflow = ceprobihalflow * (1 + bdupcnt1)/(1 + bdupcnt1 + bdupcnt2);
/*
   pceprobfhigh = pceprobihigh * (1 + bdupcnt1)/(1 + bdupcnt1 + bdupcnt2);
   pceprobflow = pceprobilow * (1 + bdupcnt1)/(1 + bdupcnt1 + bdupcnt2);
   pceprobfhalfhigh = pceprobihalfhigh * (1 + bdupcnt1)/(1 + bdupcnt1 +
bdupcnt2);
   pceprobfhalflow = pceprobihalflow * (1 + bdupcnt1)/(1 + bdupcnt1 +
bdupcnt2);
*/
   if (0 < ceprobi_final < 1) and code_final not in ('N ',' N') then do;
      ceprobihigh2 = 1;
      ceprobilow2 = 0;
      ceprobihalfhigh2 = ceprobi_final + (1 - ceprobi_final)/2 ;
      ceprobihalflow2 = ceprobi_final/2;
   end;
   else do;
      ceprobihigh2 = ceprobi_final;
      ceprobilow2 = ceprobi_final;
      ceprobihalfhigh2 = ceprobi_final;
      ceprobihalflow2 = ceprobi_final;
   end;

   if (0 < pceprobi < 1) then do;
      pceprobihigh2 = 1;
      pceprobilow2 = 0;
      pceprobihalfhigh2 = pceprobi + (1 - pceprobi)/2;
      pceprobihalflow2 = pceprobi / 2;
   end;
   else do;
      pceprobihigh2 = pceprobi;
      pceprobilow2 = pceprobi;
      pceprobihalfhigh2 = pceprobi;
      pceprobihalflow2 = pceprobi;
   end;

   ceprobfhigh2 = ceprobihigh2 * (1 + bdupcnt1)/(1 + bdupcnt1 + bdupcnt2);
   ceprobflow2 = ceprobilow2 * (1 + bdupcnt1)/(1 + bdupcnt1 + bdupcnt2);
   ceprobfhalfhigh2 = ceprobihalfhigh2 * (1 + bdupcnt1)/(1 + bdupcnt1 +
bdupcnt2);
   ceprobfhalflow2 = ceprobihalflow2 * (1 + bdupcnt1)/(1 + bdupcnt1 +
bdupcnt2);

   pceprobfhigh2 = pceprobihigh2 * (1 + bdupcnt1)/(1 + bdupcnt1 + bdupcnt2);
   pceprobflow2 = pceprobilow2 * (1 + bdupcnt1)/(1 + bdupcnt1 + bdupcnt2);
   pceprobfhalfhigh2 = pceprobihalfhigh2 * (1 + bdupcnt1)/(1 + bdupcnt1 +
bdupcnt2);
   pceprobfhalflow2 = pceprobihalflow2 * (1 + bdupcnt1)/(1 + bdupcnt1 +
bdupcnt2);

proc print data = erevsam (where=(ceprobihigh ne ceprobihigh2
                                 or ceprobilow ne ceprobilow2
                                 or ceprobihalfhigh ne ceprobihalfhigh2
                                 or ceprobihalflow ne ceprobihalflow2
                             /*    or pceprobilow ne pceprobilow2
                                 or pceprobihigh ne pceprobihigh2
                                 or pceprobihalfhigh ne pceprobihalfhigh2
                                 or pceprobihalflow ne pceprobihalflow2*/));
   var cid seq per pceprobi bdupcnt1 bdupcnt2/* ceprobihigh
       ceprobihigh2 ceprobilow ceprobilow2 ceprobihalfhigh ceprobihalfhigh2
       ceprobihalflow ceprobihalflow2*/ pceprobi bfinmat bbfumat;
   title 'ereace:  ceprobihigh ne ceprobhigh2 or ceprobilow ne ceprobilow2
          or ceprobihalfhigh ne ceprobhalfhigh2 or ceprobihalflow ne
          ceprobihalflow2 or pceprobi didn''t work.' ;

data _null_;
   set erevsam;
   file '/home/akearney/p1eval2k/reACE/erevsamhigh.dat' lrecl = 642;
   put @1 clust $char6. @7 cid $char12. @19 seq $char2. @21 per $char5.
         @416 age 3. @421 sex 1. @425 bdupcnt1 1. @426 bdupcnt2 2.
         @433 bfinmat $char2. @509 pceprobihigh2 10.8 @519 pceprobfhigh2 11.8
         @592 ceprobihigh @602 ceprobfhigh 11.8 @613 tesfinwt_final 15.6
         @628 tesfinwt_new 15.6;

data _null_;
   set erevsam;
   file '/home/akearney/p1eval2k/reACE/erevsamlow.dat' lrecl = 642;
   put @1 clust $char6. @7 cid $char12. @19 seq $char2. @21 per $char5.
         @416 age 3. @421 sex 1. @425 bdupcnt1 1. @426 bdupcnt2 2.
         @433 bfinmat $char2. @509 pceprobilow2 10.8 @519 pceprobflow2 11.8
         @592 ceprobilow @602 ceprobflow 11.8 @613 tesfinwt_final 15.6
         @628 tesfinwt_new 15.6;

run;
```

calceta.sas **ATTACHMENT 9**

```
*calceta.sas : this program takes the high and low estimates of the DSEs for
the original and ReACE methodologies and gets the eta = (delta prime) /
(delta);

options ls = 80 mlogic mprint formdlim = '-';
libname rev2 '/vol/srdu8.files4/more-p1eval2k/reACE';
libname rev '/home/akearney/p1eval2k/reACE';

data dsehi (keep = totcb_dsehi);
   infile '/home/akearney/p1eval2k/reACE/dse_hi.dat' truncover end = lastrec;
   input @1 e744short $char4. @6 p512short $char4. @11 efullshort $char4.
         @16 pfullshort $char4. @21 cb_dsehi 25.15 @47 cb_ccfhi 25.15 ;

*   if cb_dsehi = 0 then do;
*      count + 1;
*      output;
*   end;

   totcb_dsehi + cb_dsehi;

   if lastrec then
      output;

data dselo (keep = totcb_dselo);
   infile '/home/akearney/p1eval2k/reACE/dse_lo.dat' truncover end = lastrec ;
   input @1 e744short $char4. @6 p512short $char4. @11 efullshort $char4.
         @16 pfullshort $char4. @21 cb_dselo 25.15 @47 cb_ccflo 25.15 ;

*   if cb_dselo = 0 then do;
*      count + 1;
*      output;
*   end;

   totcb_dselo + cb_dselo;

   if lastrec then
      output;

data dseoldhi (keep = totdseoldhi);
   infile '/home/akearney/p1eval2k/reACE/dse_old_hi.dat' truncover
end=lastrec;
   input @10 oldpost $char3. @15 dseoldhi 20.8 ;

   totdseoldhi + dseoldhi;

   if lastrec then
      output;

data dseoldlo (keep = totdseoldlo);
   infile '/home/akearney/p1eval2k/reACE/dse_old_lo.dat' truncover
end=lastrec;
   input @10 oldpost $char3. @15 dseoldlo 20.8 ;

   totdseoldlo + dseoldlo;

   if lastrec then
      output;
/*
proc sort data = dsehi;
   by e744short p512short;
```

```
proc sort data = dselo;
   by e744short p512short;

proc sort data = dseoldhi;
   by oldpost;

proc sort data = dseoldlo;
   by oldpost;

proc print data = dsehi;
*   var e744short p512short cb_dsehi;
*   sum cb_dsehi;
   title 'dsehi';

proc print data = dselo;
*   var e744short p512short cb_dselo;
*   sum cb_dselo;
   title 'dselo';

proc print data = dseoldhi;
*   var ps416 dseoldhi;
*   sum dseoldhi;
   title 'dseoldhi';

proc print data = dseoldlo;
*   var ps416 dseoldlo;
*   sum dseoldlo;
   title 'dseoldlo';
*/

data allhilo;
   merge dsehi dselo dseoldhi dseoldlo;

   delta = totdseoldhi - totdseoldlo;
   deltaprime = totcb_dsehi - totcb_dselo;

   eta = deltaprime / delta;

proc print data = allhilo;
   format totdseoldhi totdseoldlo totcb_dsehi totcb_dselo comma20.2
          eta 9.6 deltaprime delta comma14.2;
   title 'allhilo';
run;
```

varDifinnoiml.sas **ATTACHMENT 10**

```
* varDifinnoiml.sas : can't get varDifin.sas to work due to memory problems with
  proc iml.  Try without iml.;

options ls = 80 mlogic mprint;
libname rev2 '/vol/srdu8.files4/more-p1eval2k/reACE';
libname rev '/home/akearney/p1eval2k/reACE';

*  read Production output, need for C*, DD*, ddla, and nddla, see memo Q-38;
*  also need C (Cen).  Di = fimputeTCen, then get VimputationDi.;


/*
proc sort data = rev.cdatfull;
   by e744short p512short;

%macro mergeccf();

%do rep = 1 %to 128;

data ccf&rep (keep = e744short p512short ccf&rep dse&rep flag);
   set rev2.ccf&rep;

   flag = 1;

%end;

%do rep = 1 %to 128;
   proc sort data = ccf&rep;
      by flag;
%end;

data rev.ccfall;
merge ccf1 ccf2 ccf3 ccf4 ccf5 ccf6 ccf7 ccf8 ccf9 ccf10 ccf11 ccf12 ccf13 ccf14
ccf15 ccf16 ccf17 ccf18 ccf19 ccf20 ccf21 ccf22 ccf23 ccf24 ccf25 ccf26
ccf27 ccf28 ccf29 ccf30 ccf31 ccf32 ccf33 ccf34 ccf35 ccf36 ccf37 ccf38
ccf39 ccf40 ccf41 ccf42 ccf43 ccf44 ccf45 ccf46 ccf47 ccf48 ccf49 ccf50
ccf51 ccf52 ccf53 ccf54 ccf55 ccf56 ccf57 ccf58 ccf59 ccf60 ccf61 ccf62
ccf63 ccf64 ccf65 ccf66 ccf67 ccf68 ccf69 ccf70 ccf71 ccf72 ccf73 ccf74
ccf75 ccf76 ccf77 ccf78 ccf79 ccf80 ccf81 ccf82 ccf83 ccf84 ccf85 ccf86
ccf87 ccf88 ccf89 ccf90 ccf91 ccf92 ccf93 ccf94 ccf95 ccf96 ccf97 ccf98
ccf99 ccf100 ccf101 ccf102 ccf103 ccf104 ccf105 ccf106 ccf107 ccf108 ccf109
ccf110 ccf111 ccf112 ccf113 ccf114 ccf115 ccf116 ccf117 ccf118 ccf119 ccf120
ccf121   ccf122   ccf123   ccf124   ccf125   ccf126   ccf127   ccf128;
   by flag;

%mend mergeccf;

%mergeccf;
*/

proc iml;
*   create rev.ccftemp var _all_;
   use rev.ccfall;
   read all var{ccf1 ccf2 ccf3 ccf4 ccf5 ccf6 ccf7 ccf8 ccf9 ccf10 ccf11 ccf12
 ccf13 ccf14
ccf15 ccf16 ccf17 ccf18 ccf19 ccf20 ccf21 ccf22 ccf23 ccf24 ccf25 ccf26
ccf27 ccf28 ccf29 ccf30 ccf31 ccf32 ccf33 ccf34 ccf35 ccf36 ccf37 ccf38
ccf39 ccf40 ccf41 ccf42 ccf43 ccf44 ccf45 ccf46 ccf47 ccf48 ccf49 ccf50
ccf51 ccf52 ccf53 ccf54 ccf55 ccf56 ccf57 ccf58 ccf59 ccf60 ccf61 ccf62
ccf63 ccf64 ccf65 ccf66 ccf67 ccf68 ccf69 ccf70 ccf71 ccf72 ccf73 ccf74
ccf75 ccf76 ccf77 ccf78 ccf79 ccf80 ccf81 ccf82 ccf83 ccf84 ccf85 ccf86
ccf87 ccf88 ccf89 ccf90 ccf91 ccf92 ccf93 ccf94 ccf95 ccf96 ccf97 ccf98
ccf99 ccf100 ccf101 ccf102 ccf103 ccf104 ccf105 ccf106 ccf107 ccf108 ccf109
ccf110 ccf111 ccf112 ccf113 ccf114 ccf115 ccf116 ccf117 ccf118 ccf119 ccf120
ccf121   ccf122   ccf123   ccf124   ccf125   ccf126   ccf127   ccf128} into ccf;

%* get row totals ;
ccfsum = ccf[,+];
c = ncol(ccf);
print 'num col ' c;
r = nrow(ccf);
print 'num row ' r;

%******************* CHANGE ***************************;
%* get mean ccf ;

ccfmean = ccfsum / c;
print 'ccfsum ' ccfsum 'ccfmean ' ccfmean;

*create rev.ccfmean from ccfmean;

%************************* CHANGE *********************;

phi = 1.3;
gamma = 1;
eta = 0.410048;

x = shape({0},r,c);
fimpute = shape({0},r,c);

   do k = 1 to c;
      x[,k] = ccf[,k] - ccfmean;
      fimpute[,k] = (phi # gamma # eta # x[,k]) + ccfmean;
   end;

file '/home/akearney/p1eval2k/reACE/fimpute.dat' lrecl = 1250;
do i = 1 to nrow(fimpute);
   do j = 1 to ncol(fimpute);
      put (fimpute[i,j]) 9.6 @;
   end;
   put;
end;

/*
postnum = 'ps1' : 'ps7584';
create rev.fimpute from fimpute[colname=postnum];

fimpute = t(fimpute);
*/
quit;

*proc contents data = rev.fimpute;
run;
*proc transpose data =  out=mergePT name = mrate&rep prefix = ps;
*                id pcnt;
```

calcvarD.sas

```
* calcvarD.sas : matrix is too big for proc iml (128 X 7584).  Will calc
this way.;

options ls = 80 missing = ' ';
libname rev2 '/vol/srdu8.files4/more-p1eval2k/reACE';
libname rev '/home/akearney/p1eval2k/reACE';

proc sort data = rev.cdatfull;
   by e744short p512short;

proc transpose data = rev.fimputet out= fimpute (drop = _name_) prefix = alt;
run;

*proc print data = fimpute (obs = 10);
*    var alt115-alt128;
*    title 'fimpute';

data cdatfull;
   set rev.cdatfull;

   flag = 1;

data fimpute;
   set fimpute;

   flag = 1;

* get a matrix of DSEs instead of CCFs.;

data dse (keep = dse1-dse128);
   merge fimpute cdatfull end = lastrec;
      by flag;

   array alt{128} alt1-alt128;
   array dse{128} dse1-dse128;

   if alt1 = ' ' then
      countalt1miss + 1;

   do i = 1 to 128;
      dse{i} = alt{i} * C;
   end;
/*
proc print data = dse (obs = 10);
   var dse115-dse128  ;
   format dse1-dse5 comma15.2;
   title 'dse';
*/
*   proc contents data = dse;
*   title 'dse:  7584 X 128?';

proc transpose data = dse out= dseT (drop = _name_) prefix = ps;
run;

data sumacrossps (keep = sumps);
   set dseT end = lastrec;

   array dse{7584} ps1-ps7584 ;

   do i = 1 to 7584;
```

```
      sumps + dse{i};
   end;

   output;
   sumps = 0;

*proc print data = sumacrossps;* (obs = 10);
*   sum sumps;
*   title 'sumacrossps';

proc iml;

use sumacrossps;
read all var{sumps} into natldse;

mindse = min(natldse);
maxdse = max(natldse);

range = maxdse - mindse;

print 'maxdse = ' maxdse[format=14.1] ' mindse = ' mindse[format=14.1]
      ' range = ' range[format=14.1];

quit;

proc transpose data = sumacrossps out= sumT (drop = _name_) prefix = sumps;
run;

data varD (keep = varD stddev sumsqdiff sumalt avgalt);
   set sumT;

   array sumps{128} sumps1-sumps128;
   array diff{128} diff1-diff128 ;
   retain sumsqdiff 0;

   do i = 1 to 128;
      sumalt + sumps{i};
   end;

   avgalt = sumalt / 128;

   do i = 1 to 128;
      diff{i} = sumps{i} - avgalt;
      sumsqdiff = sumsqdiff + (diff{i} * diff{i});
   end;

   varD = sumsqdiff / 127;
   stddev = sqrt(varD);

proc print data = varD;
   var sumsqdiff varD stddev avgalt sumalt;
   format sumsqdiff varD avgalt sumalt comma20.1;
   title 'VarD';

run;
```

calcvarDescap.sas **ATTACHMENT 12**

```
* calcvarDescap.sas : calculate the squared deviations of Don's ESCAP II ;

options ls = 80 missing = ' ';
libname rev2 '/vol/srdu8.files4/more-p1eval2k/reACE';
libname rev '/home/akearney/p1eval2k/reACE';

/*
%macro read (alt= ,cnum= ,nl=, rep= );

data alt&rep;
   infile "/home/akearney/p1eval2k/dse/apres24aout/a&alt.&cnum.&nl..dat"
          truncover lrecl = 34;
   input @1 altcode $char4. @6 poststr $char3. @11 dse&rep 13.2
         @25 ccf&rep 10.8;

   flag = 1;

%mend read;

%macro mac1;
%do n = 1 %to 16;
%let rep = &n;
%if &n < 10 %then
   %let cnum = 0&n;
%else
   %let cnum = &n  ;
%read(alt=1,cnum=&cnum,nl= ,rep=&rep);
%end;
%mend mac1;

%mac1;

%macro mac2;
%do n = 1 %to 16;
%let rep = %eval(&n + 16);
%if &n < 10 %then
   %let cnum = 0&n;
%else
   %let cnum = &n  ;
%read(alt=3,cnum=&cnum,nl= ,rep=&rep);
%end;
%mend mac2;

%mac2;

%macro mac3;

%let start = 33;

%do n = 1 %to 6;

   %let rep = &start;
   %read(alt=4,cnum=01,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=4,cnum=02,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=4,cnum=04,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=4,cnum=05,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=4,cnum=07,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=4,cnum=09,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=4,cnum=12,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=4,cnum=16,nl=&n ,rep=&rep);

   %let rep = %eval(&rep + 6);
   %read(alt=5,cnum=03,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=5,cnum=06,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=5,cnum=08,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=5,cnum=10,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=5,cnum=11,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=5,cnum=13,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=5,cnum=14,nl=&n ,rep=&rep);
   %let rep = %eval(&rep + 6);
   %read(alt=5,cnum=15,nl=&n ,rep=&rep);

   %let start = %eval(&start + 1);

%end;
%mend mac3;

%mac3;

data rev.ccfallescap;
   merge alt1 alt2 alt3 alt4 alt5 alt6 alt7 alt8 alt9 alt10 alt11 alt12 alt13 alt14
         alt15 alt16 alt17 alt18 alt19 alt20 alt21 alt22 alt23 alt24 alt25 alt26
         alt27 alt28 alt29 alt30 alt31 alt32 alt33 alt34 alt35 alt36 alt37 alt38
         alt39 alt40 alt41 alt42 alt43 alt44 alt45 alt46 alt47 alt48 alt49 alt50
         alt51 alt52 alt53 alt54 alt55 alt56 alt57 alt58 alt59 alt60 alt61 alt62
         alt63 alt64 alt65 alt66 alt67 alt68 alt69 alt70 alt71 alt72 alt73 alt74
         alt75 alt76 alt77 alt78 alt79 alt80 alt81 alt82 alt83 alt84 alt85 alt86
         alt87 alt88 alt89 alt90 alt91 alt92 alt93 alt94 alt95 alt96 alt97 alt98
         alt99 alt100 alt101 alt102 alt103 alt104 alt105 alt106 alt107 alt108 alt109
         alt110 alt111 alt112 alt113 alt114 alt115 alt116 alt117 alt118 alt119 alt120
         alt121 alt122 alt123 alt124 alt125 alt126 alt127 alt128;
   by flag;


*      ccf&rep = (ddstar/(cstar + ddla + nddla)) * (CErate&rep / Mrate&rep);

*      dse&rep = (ddstar) * (CErate&rep / Mrate&rep);
*/
data prepostus;
   infile '/home/akearney/p1eval2k/dmba01/pspreus.dat' truncover lrecl= 255;
   input @1 poststr $char3. @9 cstar 10. @31 ddstar 10. @224 ddla 10.
         @235 nddla 10. @246 C 10. ;

proc iml;

   use prepostus;
   read all var{C} into Cen;

   use prepostus;
   read all var{poststr} into post;
   use rev.ccfallescap;
   read all var{ccf1 ccf2 ccf3 ccf4 ccf5 ccf6 ccf7 ccf8 ccf9 ccf10 ccf11
                   ccf12 ccf13 ccf14 ccf15 ccf16 ccf17 ccf18 ccf19 ccf20
                   ccf21 ccf22 ccf23 ccf24 ccf25 ccf26 ccf27 ccf28 ccf29
                   ccf30 ccf31 ccf32 ccf33 ccf34 ccf35 ccf36 ccf37 ccf38
                   ccf39 ccf40 ccf41 ccf42 ccf43 ccf44 ccf45 ccf46 ccf47
                   ccf48 ccf49 ccf50 ccf51 ccf52 ccf53 ccf54 ccf55 ccf56
                   ccf57 ccf58 ccf59 ccf60 ccf61 ccf62 ccf63 ccf64 ccf65
                   ccf66 ccf67 ccf68 ccf69 ccf70 ccf71 ccf72 ccf73 ccf74
                   ccf75 ccf76 ccf77 ccf78 ccf79 ccf80 ccf81 ccf82 ccf83
                   ccf84 ccf85 ccf86 ccf87 ccf88 ccf89 ccf90 ccf91 ccf92
                   ccf93 ccf94 ccf95 ccf96 ccf97 ccf98 ccf99 ccf100 ccf101
                   ccf102 ccf103 ccf104 ccf105 ccf106 ccf107 ccf108 ccf109
                   ccf110 ccf111 ccf112 ccf113 ccf114 ccf115 ccf116 ccf117
                   ccf118 ccf119 ccf120 ccf121 ccf122 ccf123 ccf124 ccf125
                   ccf126 ccf127 ccf128} into ccfall;

%* get row totals ;
ccfsum = ccfall[,+];
c = ncol(ccfall);
print 'num col ' c;
r = nrow(ccfall);
print 'num row ' r;

%* get mean ccf ;

ccfmean = ccfsum / c;
print 'ccfsum ' ccfsum 'ccfmean ' post '  ' ccfmean;

phi = 1.3;

x = shape({0},r,c);
ccfadj = shape({0},r,c);

   do k = 1 to c;
      x[,k] = ccfall[,k] - ccfmean;
      ccfadj[,k] = (phi # x[,k]) + ccfmean;
   end;

rx = nrow(x);
cx = ncol(x);
radj = nrow(ccfadj);
cadj = ncol(ccfadj);

print 'rx = ' rx ' cx = ' cx ' rcadj = ' radj ' cadj = ' cadj;
colprt = 'ccf1;
print 'post ' post '  ' ccfadj[1] = ' (ccfadj[1:5]) [format=10.8];

D=shape({0},128,1);
%*print 'precalc D = ' D[format=10.1];

D = ccfadj` * Cen ;
rD = nrow(D);
cD = ncol(D);

print 'rows in D ' rD ' columns in D ' cD;
%*print 'D = ' post ' ' D[format=15.1];

%* get row totals ;
Dsum = D[+];
Dbar = Dsum / c;

print' Dsum = ' Dsum[format=11.1]
```

```
      ' Dbar = '  Dbar[format=10.1] ' c = ' c;

%* calculate Vimp(Di) ;

dev=shape({0},128,1);
do i = 1 to c;
   dev[i] = (D[i] - Dbar);
end;

minD = min(D);
maxD = max(D);
rangeD = maxD - minD;

print 'alt '  'D = '  (D[1:128])[format=15.1] ' dev = ' (dev[1:128])[format=8.1];

%*print 'D = '  D[format=15.1] ' dev = ' (dev[1:128])[format=8.1];

VimpDi = ssq(dev) / (c - 1);

stddevVimpDi = sqrt(VimpDi);

print 'VimpDi = ' VimpDi[format=20.1] 'std dev VimpDi = ' stddevVimpDi[format=10.1];
print 'minimum D ' minD[format=comma18.0] ' maximum D ' maxD[format=comma18.0];
print 'range of D = ' rangeD[format=comma12.0];

quit;

run;
```