Appendix 2

Hot decking Diary CEX to WAPOP survey multiple imp with donor wgt adj.sas

```
**************************************************************************;
* program:    Hot decking Diary CEX to WAPOP survey multiple imp          ;
*             with donor wgt adj.sas[i]                                    ;
* programmer: Rick Peterson                                               ;
* project:    Washington Excise Tax Microsimulation Model                 ;
* date:       March 21, 2002                                              ;
*                                                                         ;
* purpose:    Match households in WA pop survey with similar households   ;
*             in Consumer Expenditure Survey. Matching variables: Income, ;
*             housing tenure, housing building type, presence of person   ;
*             over 64 in household, presence of person under 18 in        ;
*             household, and family size.                                 ;
*                                                                         ;
*-------------------------------------------------------------------------;
* libraries: popsur - location of WAPOP survey data                       ;
*            mdimpute - destination of matching files                     ;
*                                                                         ;
*-------------------------------------------------------------------------;
* incoming:   From Diary CEX: fmly99.1, fmly99.2, fmly99.3 fmly99.4       ;
*             From WA population survey: sps00f04                         ;
*                                                                         ;
*-------------------------------------------------------------------------;
* formats:   Format for WAPOP survey                                        ;
*                                                                         ;
*-------------------------------------------------------------------------;
* outgoing:   DImpute1, DImpute2, DImpute3, DImpute4: containing household;
*             matches.                                                    ;
*             dqtrsmatched: containing number of quarters matched for     ;
*             each household - used in later program to adjust consumption;
*             data.                                                       ;
*                                                                         ;
*-------------------------------------------------------------------------;
* reports:   None                                                         ;
*                                                                         ;
*-------------------------------------------------------------------------;
* changes:                                                                ;
*                                                                         ;
*-------------------------------------------------------------------------;
* notes:     Each Washington household is matched to a CEX household for  ;
*            each of the 4 quarter of CEX data.  Matching is done in      ;
*            three rounds.  If a match is not found then matching         ;
*            criteria are relaxed for next round.  When more than one     ;
*            donor is available then the donor is selected randomly.      ;
*            The likelihood of donor selection is proportional to the     ;
*            relative weight of the donor within the group of potential   ;
*            donors.                                                      ;
*                                                                         ;
*                                                                         ;
**************************************************************************;

*-------------------------------------------------------------------------;
*Read in donor pool from CEX diary data sets;
*-------------------------------------------------------------------------;

%let y = 99;
filename dfmly1 "c:\data\diary survey\fmlyd&y.1.txt";
filename dfmly2 "c:\data\diary survey\fmlyd&y.2.txt";
filename dfmly3 "c:\data\diary survey\fmlyd&y.3.txt";
filename dfmly4 "c:\data\diary survey\fmlyd&y.4.txt";

%macro loop1;
%do x= 1 %to 4 %by 1;
data dfmly&x;
   infile dfmly&x lrecl=1549;
   input @1 newid 8.
            @148 finlwt21 11.3
            @43 cutenure $1.
            @78 Fam_size 2.
        @544 perslt18 2.
            @582 respstat $1.
            @36 age_ref 2.
            @547 perot64 2.
            @139 fincbtax 8.
            @45 Descrip $2.
            @83 fbsnsx 8.
            @103 ffarmx 8.
            @584 roomx 8.
            @526 othrntx 8.
            @423 jfs_amt 8.
            ;
*Eliminate obs without complete reporting of income;
if respstat ne '1' then delete;
*Create a variable to merge with WAPOP survey observation;
Flag = 1;
*Create variable to match housing tenure;
if cutenure in ('1', '2','3') then own = 1;
else if cutenure eq '4' then own =2;
else if cutenure eq '5' then own =3;
else own =4;
*Create variable to match building type;
if Descrip in ('01') then bld = 1;
else if Descrip in ('02','03','04') then bld = 5;
else if Descrip in ('05', '06') then bld = 3;
else bld = 5;
*Cap household size at 6 people;
if fam_size >5 then fam_size = 6;
*Create variable to match households with persons under 18;
if perslt18>0 then under18 = 1;
   else under18=0;
*Create variable to match households with persons over age 64;
if perot64>0 then over64 = 1;
   else over64 = 0;
*Remove business losses and value of food stamps from income;
   if fbsnsx ne . and fbsnsx <0 then fincbtax = fincbtax - fbsnsx;
   if ffarmx ne . and ffarmx <0 then fincbtax = fincbtax - ffarmx;
   if roomx ne . and roomx <0 then fincbtax = fincbtax - roomx;
   if othrntx ne . and othrntx <0 then fincbtax = fincbtax - othrntx;
   if jfs_amt ne . then fincbtax = fincbtax - jfs_amt;
*Drop household that is in fmly data set but has no consumption information
   in expn data set;
   %if &x=1 %then %do;
    if newid=189172 then delete;
    if newid=189171 then delete;
   %end;
   %if &x=2 %then %do;
    if newid=204201 then delete;
    if newid=204202 then delete;
   %end;
run;
%end;
%mend Loop1;
%Loop1;

*-------------------------------------------------------------------------;
*Read in data from WAPOP survey;
*-------------------------------------------------------------------------;

Data one;
set popsur.sps00f04;
*Create variable for presence of person over64 in household;
if age >64 then over64popsur =1;
else over64popsur =0;
keep id age over64popsur;
run;

proc summary data=one;
by id;
var over64popsur;
output out=two sum=;
run;

Data A;
set popsur.sps00f04;
where pnum=1;
Flag = 1;
*Cap household size at 6 people;
if peopl >5 then peopl = 6;
*Create variable to match persons in household under 18;
if chldrn18>0 then WAsurvey18 = 1;
   else WAsurvey18=0;
*Change building type variable to match CEX response possibilities;
   if q3p1 in (1,2) then bldpopsurvey = 1;
   else if q3p1 = 3 then bldpopsurvey = 3;
   else bldpopsurvey = 5;
keep Flag ID pnum age bldpopsurvey q3p2 fnlwgt hhinc peopl WAsurvey18;
run;

*Create "Missing" - the  WAPOP data set with household information but
without consumption data;
*Merge in variable for presence of person over 64;
Data missing;
merge A two;
by id;
if over64popsur >0 then over64popsur=1;
else over64popsur=0;
drop _type_ _freq_;
run;

*-------------------------------------------------------------------------;
*FIRST ROUND OF IMPUTATON - Restrict potential donors by tenure,
property type, family size, age and income;
```

```
*-------------------------------------------------------------------------;
*
Turn off writing to the SAS log;
options nonotes nosource nosource2 errors=0;

*Create macro loop to find a match for the WAPOP household in each
 of the 4 CEX diary data sets;

%macro loop;
%do x = 1 %to 4;

*Count the number of obs in missing;

proc means data = missing noprint;
var flag;
output out=amiss sum=;
run;

*Delete file which will contain imputed values;

Proc delete data=mdimpute.ADImpute&x;
run;

*Create file which will contain imputed values;

Data mdimpute.ADImpute&x;
set missing;
keep ID;
Proc sort;
by ID;
run;

*Create macro variable equal to number of obs with missing CEX data;

Data num;
set amiss;
call symput('num',left(_freq_));

*Macro function to do imputation;

%macro Impute(a=);

*Loop which starts with first obs with missing data and ends with last one;

%do %until (&a gt &num);

*Pull the next obs with missing data;

Data two;
Set missing;
If _N_ =&a;

*Merge Donor data set with missing data set;
*Keep only Obs from Donor data set with similar characteristics;

Data threea;
Merge Dfmly&x Two; By Flag;
```

```
if own ne q3p2 and q3p2 in (1,2,3)
then delete;                              *Requires exact match for tenure;
if bld ne bldpopsurvey and bldpopsurvey in (1,3,5)
then delete;                              *Requires exact match for bld type;

if over64popsur ne over64 then delete; *Requires presence of person over64;

if hhinc <100000 then do;
   if fincbtax gt (hhinc +7500)
   or fincbtax lt (hhinc-7500) then delete; *Requires income to be within
7,500 if income below 100,000;
   end;
if 100000<= hhinc <= 130000 then do;
      if fincbtax gt (hhinc+12500)
      or fincbtax lt (hhinc-12500) then delete;*Requires income to be within
12,500 if income above 100,000 and below 130,000;
      end;
if hhinc > 130000 then do;
      if fincbtax gt (hhinc+20000)
      or fincbtax lt (hhinc-20000) then delete;*Requires income to be within
20,000 if income above 130,000;
      end;

if fam_size ne peopl then delete;          *Requires household size to be the
same;
if under18 ne WAsurvey18 then delete;      *Requires households to have
children under 18;

_type_=0; *Create variable for merging with output of a proc sum;


*Count number and total weight of candidate donors;

proc summary data =threea;
var finlwt21;
output out=threeb sum=wgttotal;
run;


*Select 7 donors for multiple imputation;

%do i = 1 %to 7 %by 1;

*Calc a score equal to wgt of each donor relative to all donors in this
group. Spread the scores over the interval 0 to 1 with width equal
to each donor's relative weight. Pick a random number between 0 and 1
and select a donor;
*Repeat 7 times;

data three;
merge threea threeb;
by _type_;
rename _freq_ = total;
drop _type_;
score = finlwt21/wgttotal;
if _n_=1 then
a = ranuni(0);
```

```
retain a;
retain score0 score1 0;
score0 = score1;
score1 + score;
newid&i = newid;
Keep ID newid&i _freq_;
if score0< a <= score1 then output three;


*Add imputed value to 1st round data set;

Data  mdimpute.ADImpute&x;
merge mdimpute.ADImpute&x three;
by ID;
I_round='a';
attrib I_round label ='Imputation Round' format =$1.0;
%end; *Ends multiple imputation rounds;
%let a=%eval(&a+1); *Increments 'a' to next missing obs;
%end;
%mend Impute;
%Impute(a=1);
run;

*---------------------------------------------------------------------------;
*SECOND ROUND OF IMPUTATON - Relax family size and income restrictions;
*---------------------------------------------------------------------------;

*Select obs where less than 1 match was found in first round;

Data b;
set mdimpute.ADImpute&x;
Where total lt 1;
run;
Data bmissing;
merge b(in=in1) missing(in=in2) ;
by id; if in1;
run;

*Count the number of obs missing data set;

proc means data = bmissing noprint;
var flag;
output out=bmiss sum=;
run;

*Delete file which will contain imputed values;

Proc delete data=mdimpute.BDImpute&x;
run;

*Create file which will contain imputed values;

Data mdimpute.BDImpute&x;
set bmissing;
keep ID;
Proc sort;
by ID;
```

```
run;

*Create macro variable equal to number of obs with missing AV;

Data bnum;
set bmiss;
call symput('bnum',left(_freq_));

*Macro function to do imputation;

%macro BImpute(b=);

*Loop which starts with first obs and ends with last one;

%do %until (&b gt &bnum);

*Pull the next obs with missing data;

Data two;
Set bmissing;
If _N_ =&b;

*Merge Donor data set with missing data set;
*Keep only Obs from Donor data set with similar characteristics;

Data threea;
Merge dfmly&x Two; By Flag;
if own ne q3p2 and q3p2 in (1,2,3)
then delete;                              *Requires exact match for tenure;

if bld ne bldpopsurvey and bldpopsurvey in (1,3,5)
then delete;                              *Requires exact match for bld type;

if over64popsur ne over64 then delete; *Requires presence of person over64;

if hhinc <100000 then do;
   if fincbtax gt (hhinc +15000)
   or fincbtax lt (hhinc-15000) then delete; *Requires income to be within
15,000 if income below 100,000;
   end;
if 100000<= hhinc <= 130000 then do;
      if fincbtax gt (hhinc+25000)
      or fincbtax lt (hhinc-25000) then delete;*Requires income to be within
25,000 if income above 100,000 and below 130,000;
      end;
if hhinc > 130000 then do;
      if fincbtax gt (hhinc+35000)
      or fincbtax lt (hhinc-35000) then delete;*Requires income to be within
35,000 if income above 130,000;
      end;

if fam_size gt (peopl+1)
or fam_size lt (peopl-1) then delete; *Requires household size to be within
one;

if under18 ne WAsurvey18 then delete;      *Requires households to have
children under 18;
```

```
_type_=0; *Create variable for merging with output of a proc sum;


*Count number and total weight of candidate donors;

proc summary data =threea;
var finlwt21;
output out=threeb sum=wgttotal;
run;


*Select 7 donors for multiple imputation;

%do i = 1 %to 7 %by 1;

*Calc a score equal to wgt of each donor relative to all donors in this
group. Spread the scores over the interval 0 to 1 with width equal
to each donor's relative weight. Pick a random number between 0 and 1
and select a donor;
*Repeat 7 times;

data three;
merge threea threeb;
by _type_;
rename _freq_ = total;
drop _type_;
score = finlwt21/wgttotal;
if _n_=1 then
a = ranuni(0);
retain a;
retain score0 score1 0;
score0 = score1;
score1 + score;
newid&i = newid;
Keep ID newid&i _freq_;
if score0< a <= score1 then output three;


*Add imputed value to 2st round data set;

Data  mdimpute.BDImpute&x;
merge mdimpute.BDImpute&x three;
by ID;
I_round='b';
attrib I_round label ='Imputation Round' format =$1.0;
%end; *Ends multiple imputation rounds;
%let b=%eval(&b+1);*Increments 'b' to next missing obs;
%end;
%mend BImpute;
%BImpute(b=1);
run;

*-----------------------------------------------------------------------;
*THIRD ROUND OF IMPUTATON - eliminate bld type restriction
and relax family size restriction;
*-----------------------------------------------------------------------;
```

```
*Select obs where less than 1 match was found in second round;

Data c;
set mdimpute.BDImpute&x;
Where total lt 1;
run;
Data cmissing;
merge c(in=in1) missing(in=in2) ;
by id; if in1;
run;

*Count the number of obs with missing AV;

proc means data = cmissing noprint;
var flag;
output out=cmiss sum=total;
run;

*Delete file which will contain imputed values;

Proc delete data=mdimpute.CDImpute&x;
run;

*Create file which will contain imputed values;

Data mdimpute.CDImpute&x;
set cmissing;
keep ID;
Proc sort;
by ID;
run;

*Create macro variable equal to number of obs with missing data;

Data bnum;
set cmiss;
call symput('cnum',left(_freq_));

*Macro function to do imputation;

%macro CImpute(c=);

*Loop which starts with first obs with missing AV and ends with last one;

%do %until (&c gt &cnum);

*Pull the next obs with missing data;

Data two;
Set cmissing;
If _N_ =&c;

*Merge Donor data set with obs with missing data;
*Keep only Obs from Donor data set with similar characteristics;

Data threea;
```

```
Merge dfmly&x Two; By Flag;
if own ne q3p2 and q3p2 in (1,2,3)
then delete;                             *Requires exact match for tenure;

*if bld ne bldpopsurvey and bldpopsurvey in (1,3,5)
then delete;                             *Requires exact match for bld type;

if over64popsur ne over64 then delete; *Requires presence of person over64;

if hhinc <100000 then do;
   if fincbtax gt (hhinc +15000)
   or fincbtax lt (hhinc-15000) then delete; *Requires income to be within
15,000 if income below 100,000;
   end;
if 100000<= hhinc <= 130000 then do;
      if fincbtax gt (hhinc+25000)
      or fincbtax lt (hhinc-25000) then delete;*Requires income to be within
25,000 if income above 100,000 and below 130,000;
      end;
if hhinc > 130000 then do;
      if fincbtax gt (hhinc+35000)
      or fincbtax lt (hhinc-35000) then delete;*Requires income to be within
35,000 if income above 130,000;
      end;

if fam_size gt (peopl+2)
or fam_size lt (peopl-2) then delete; *Requires household size to be within
two;

if under18 ne WAsurvey18 then delete;      *Requires households to have
children under 18;

_type_=0; *Create variable for merging with output of a proc sum;


*Count number and total weight of candidate donors;

proc summary data =threea;
var finlwt21;
output out=threeb sum=wgttotal;
run;


*Select 7 donors for multiple imputation;

%do i = 1 %to 7 %by 1;

*Calc a score equal to wgt of each donor relative to all donors in this
group. Spread the scores over the interval 0 to 1 with width equal
to each donor's relative weight. Pick a random number between 0 and 1
and select a donor;
*Repeat 7 times;

data three;
merge threea threeb;
by _type_;
rename _freq_ = total;
```

```
drop _type_;
score = finlwt21/wgttotal;
if _n_=1 then
a = ranuni(0);
retain a;
retain score0 score1 0;
score0 = score1;
score1 + score;
newid&i = newid;
Keep ID newid&i _freq_;
if score0< a <= score1 then output three;

*Add imputed value to 3rd round data set;

Data  mdimpute.CDImpute&x;
merge mdimpute.CDImpute&x three;
by ID;
I_round='c';
attrib I_round label ='Imputation Round' format =$1.0;
%end; *Ends multiple imputation rounds;
%let c=%eval(&c+1);*Increments 'c' to next missing obs;
%end;
%mend CImpute;
%CImpute(c=1);
run;
%end;
%mend Loop;
%Loop;
run;


*-------------------------------------------------------------------------;
*Combine the data sets from the 3 imputation rounds;
*Drop any obs without a match;
*-------------------------------------------------------------------------;

%macro loop1;
%do x=1 %to 4 %by 1;
*Delete file which will contain imputed values;
Proc delete data=mdimpute.DImpute&x;
run;

data mdimpute.DImpute&x;
merge mdimpute.ADImpute&x mdimpute.BDImpute&x mdimpute.CDImpute&x;
by id;
if newid1 = . then delete;
run;
%end;
%mend Loop1;
%loop1;

*-------------------------------------------------------------------------;
*Determine number of quarters that each id has a match;
*-------------------------------------------------------------------------;

data mdimpute.dqtrsmatched;
merge mdimpute.dimpute1(in=a) mdimpute.dimpute2(in=b)
```

```
mdimpute.dimpute3(in=c) mdimpute.dimpute4(in=d);
by id;
if a then qtr1=1; else qtr1=0;
if b then qtr2=1; else qtr2=0;
if c then qtr3=1; else qtr3=0;
if d then qtr4=1; else qtr4=0;
Dqtrs=qtr1+qtr2+qtr3+qtr4;
keep id Dqtrs;
proc sort data=mdimpute.dqtrsmatched;
by id;
run;

options notes source source2 errors=20;

*-------------------------------------------------------------------------;
*Check to see if a newid is missing when other iterations contain values;
*-------------------------------------------------------------------------;

proc delete data=test;
run;
%macro loop2;
%do w=1 %to 4 %by 1;
data zzz;
set mdimpute.AdImpute&w;
drop I_round&w;
if newid1 = . and sum(newid2, newid3, newid4, newid5, newid6,
newid7)>0 then output;
if newid2 = . and sum(newid1, newid3, newid4, newid5, newid6,
newid7)>0 then output;
if newid3 = . and sum(newid2, newid1, newid4, newid5, newid6,
newid7)>0 then output;
if newid4 = . and sum(newid2, newid3, newid1, newid5, newid6,
newid7)>0 then output;
if newid5 = . and sum(newid2, newid3, newid4, newid1, newid6,
newid7)>0 then output;
if newid6 = . and sum(newid2, newid3, newid4, newid5, newid1,
newid7)>0 then output;
if newid7 = . and sum(newid2, newid3, newid4, newid5, newid6,
newid1)>0 then output;
run;
proc append base = test Data=zzz;
run;
%end;
%mend loop2;
%loop2;
```

```
**************************************************************************;
* program:    Hot decking Interview CEX to WAPOP survey multiple imp      ;
*             with donor wgt adj.sas                                      ;
* programmer: Rick Peterson                                               ;
* project:    Washington Excise Tax Microsimulation Model                 ;
* date:       March 21, 2002                                              ;
*                                                                         ;
* purpose:    Match households in WA pop survey with similar households   ;
*             in Consumer Expenditure Survey. Matching variables: Income, ;
*             housing tenure, housing building type, presence of person   ;
*             over 64 in household, presence of person under 18 in        ;
*             household, and family size.                                 ;
*                                                                         ;
*-------------------------------------------------------------------------;
* libraries: popsur - location of WAPOP survey data                       ;
*            miimpute - destination of matching files                     ;
*                                                                         ;
*-------------------------------------------------------------------------;
* incoming:   From Interview CEX: fmly99.1, fmly99.2, fmly99.3 fmly99.4   ;
*             From WA population survey: sps00f04                         ;
*                                                                         ;
*-------------------------------------------------------------------------;
* formats:   Format for WAPOP survey                                        ;
*                                                                         ;
*-------------------------------------------------------------------------;
* outgoing:   IImpute1, IImpute2, IImpute3, IImpute4: containing household;
*             matches.                                                    ;
*             iqtrsmatched: containing number of quarters matched for     ;
*             each household - used in later program to adjust consumption;
*             data.                                                       ;
*                                                                         ;
*-------------------------------------------------------------------------;
* reports:   None                                                         ;
*                                                                         ;
*-------------------------------------------------------------------------;
* changes:                                                                ;
*                                                                         ;
*-------------------------------------------------------------------------;
* notes:     Each Washington household is matched to a CEX household for  ;
*            each of the 4 quarter of CEX data.  Matching is done in      ;
*            three rounds.  If a match is not found then matching         ;
*            criteria are relaxed for next round.  When more than one     ;
*            donor is available then the donor is selected randomly.      ;
*            The likelihood of donor selection is proportional to the     ;
*            relative weight of the donor within the group of potential   ;
*            donors.                                                      ;
*                                                                         ;
*                                                                         ;
**************************************************************************;

*-------------------------------------------------------------------------;
*Read in donor pool from Interview data sets;
*-------------------------------------------------------------------------;

%let y = 99;
filename ifmly1 "c:\data\interview survey\fmlyi&y.1x.txt";
filename ifmly2 "c:\data\interview survey\fmlyi&y.2.txt";
```

Hot decking Interview CEX to WAPOP survey multiple imp with donor wgt adj.sas

```
filename ifmly3 "c:\data\interview survey\fmlyi&y.3.txt";
filename ifmly4 "c:\data\interview survey\fmlyi&y.4.txt";

%macro loop1;
%do x= 1 %to 4 %by 1;
data ifmly&x;
   infile ifmly&x lrecl=3461;
   input @1 newid 8.
            @331 finlwt21 11.3
            @216 cutenure $1.
            @242 Fam_size 2.
        @633 perslt18 2.
            @636 persot64 2.
            @245 Fam_type $1.
            @556 no_earner 2.
            @3324 inclass $2.
            @681 respstat $1.
            @11 age_ref 2.
            @303 fincbtax 9.
            @61 building $2.
            @351 fnonfrmx 9.
            @274 ffrmincx 9.
            @456 inclossa 8.
            @465 inclossb 8.
            @506 jfdstmpa 8.
            ;
*Eliminate obs without complete reporting of income;
if respstat ne '1' then delete;
*Create a variable to merge with WAPOP survey observation;
Flag = 1;
*Create variable to match housing tenure;
if cutenure in ('1', '2','3') then own = 1;
else if cutenure eq '4' then own =2;
else if cutenure eq '5' then own =3;
else own =4;
*Create variable to match building type;
if building in ('01') then bld = 1;
else if building in ('02','03','04','05','06','07','08') then bld = 2;
else if building in ('09') then bld = 3;
else bld = 5;
*Cap household size at 6 people;
if fam_size >5 then fam_size = 6;
*Create variable to match persons in household under 18;
if perslt18>0 then under18 = 1;
   else under18=0;
*Create variable to match persons in household over 64;
   if persot64>0 then over64 = 1;
   else over64 = 0;
*Remove business losses and value of food stamps from income;
   if fnonfrmx ne . and fnonfrmx <0 then fincbtax = fincbtax - fnonfrmx;
   if ffrmincx ne . and ffrmincx <0 then fincbtax = fincbtax - ffrmincx;
   if inclossa ne . and inclossa <0 then fincbtax = fincbtax - inclossa;
   if inclossb ne . and inclossb <0 then fincbtax = fincbtax - inclossb;
   if jfdstmpa ne . then fincbtax = fincbtax - jfdstmpa;
run;
%end;
%mend Loop1;
```

```
%Loop1;

*-------------------------------------------------------------------------;
*Read in data from WAPOP survey;
*-------------------------------------------------------------------------;

Data one;
set popsur.sps00f04;
*Create variable for presence of person over64 in household;
if age >64 then over64popsur =1;
else over64popsur =0;
keep id age over64popsur;
run;

proc summary data=one;
by id;
var over64popsur;
output out=two sum=;
run;

Data A;
set popsur.sps00f04;
where pnum=1;
Flag = 1;
*Cap household size at 6 people;
if peopl >5 then peopl = 6;
*Create variable to match persons in household under 18;
if chldrn18>0 then WAsurvey18 = 1;
   else WAsurvey18=0;

keep Flag ID pnum age q3p1 q3p2 fnlwgt hhinc peopl WAsurvey18;
run;

*Create "Missing" - the  WAPOP data set with household information but
without consumption data;
*Merge in variable for presence of person over 64;
Data missing;
merge A two;
by id;
if over64popsur >0 then over64popsur=1;
else over64popsur=0;
drop _type_ _freq_;
run;

*-------------------------------------------------------------------------;
*FIRST ROUND OF IMPUTATON - Restrict potential donors by tenure, property
type, family size, age and income
*-------------------------------------------------------------------------;

*Turn off writing to the SAS log;
options nonotes nosource nosource2 errors=0;

*Create macro loop to find a match for the WAPOP household in each
 of the 4 CEX interview data sets;

%macro loop;
%do x = 1 %to 4;
```

```
*Count the number of obs in missing;

proc means data = missing noprint;
var flag;
output out=amiss sum=;
run;

*Delete file which will contain imputed values;

Proc delete data=miimpute.AIimpute&x;
run;

*Create file which will contain imputed values;

Data miimpute.AIImpute&x;
set missing;
keep ID;
Proc sort;
by ID;
run;

*Create macro variable equal to number of obs with missing CEX data;

Data num;
set amiss;
call symput('num',left(_freq_));

*Macro function to do imputation;

%macro Impute(a=);

*Loop which starts with first obs with missing data and ends with last one;

%do %until (&a gt &num);

*Pull the next obs with missing data;

Data two;
Set missing;
If _N_ =&a;


*Merge Donor data set with missing data set;
*Keep only Obs from Donor data set with similar characteristics;

Data threea;
Merge ifmly&x Two; By Flag;
if own ne q3p2 and q3p2 in (1,2,3)
then delete;                          *Requires exact match for tenure;
if bld ne q3p1 and q3p1 in (1,2,3,5)
then delete;                          *Requires exact match for bld type;

if over64popsur ne over64 then delete; *Requires presence of person over64;

if hhinc <100000 then do;
   if fincbtax gt (hhinc +7500)
   or fincbtax lt (hhinc-7500) then delete; *Requires income to be within
7,500 if income below 100,000;
   end;
if 100000<= hhinc <= 130000 then do;
      if fincbtax gt (hhinc+12500)
      or fincbtax lt (hhinc-12500) then delete;*Requires income to be within
12,500 if income above 100,000 and below 130,000;
      end;
if hhinc > 130000 then do;
      if fincbtax gt (hhinc+20000)
      or fincbtax lt (hhinc-20000) then delete;*Requires income to be within
20,000 if income above 130,000;
      end;

if fam_size ne peopl then delete;          *Requires household size to be the
same;
if under18 ne WAsurvey18 then delete;       *Requires households to have
children under 18;

_type_=0; *Create variable for merging with output of a proc sum;


*Count number and total weight of candidate donors;

proc summary data =threea;
var finlwt21;
output out=threeb sum=wgttotal;
run;


*Select 7 donors for multiple imputation;

%do i = 1 %to 7 %by 1;

*Calc a score equal to wgt of each donor relative to all donors in this
group. Spread the scores over the interval 0 to 1 with width equal
to each donor's relative weight. Pick a random number between 0 and 1
and select a donor;
*Repeat 7 times;

data three;
merge threea threeb;
by _type_;
rename _freq_ = total;
drop _type_;
score = finlwt21/wgttotal;
if _n_=1 then
a = ranuni(0);
retain a;
retain score0 score1 0;
score0 = score1;
score1 + score;
newid&i = newid;
Keep ID newid&i _freq_;
if score0< a <= score1 then output three;

*Add imputed value to 1st round data set;
```

```
Data  miimpute.AIImpute&x;
merge miimpute.AIImpute&x three;
by ID;
I_round='a';
attrib I_round label ='Imputation Round' format =$1.0;
%end; *Ends multiple imputation rounds;
%let a=%eval(&a+1); *Increments 'a' to next missing obs;
%end;
%mend Impute;
%Impute(a=1);
run;

*-------------------------------------------------------------------------;
*SECOND ROUND OF IMPUTATON - Relax family size and income restrictions
*-------------------------------------------------------------------------;
*Select obs where less than 1 match was found in first round;

Data b;
set miimpute.AIImpute&x;
Where total lt 1;
run;
Data bmissing;
merge b(in=in1) missing(in=in2) ;
by id; if in1;
run;

*Count the number of obs missing data set;

proc means data = bmissing noprint;
var flag;
output out=bmiss sum=;
run;

*Delete file which will contain imputed values;

Proc delete data=miimpute.BIImpute&x;
run;

*Create file which will contain imputed values;

Data miimpute.BIImpute&x;
set bmissing;
keep ID;
Proc sort;
by ID;
run;

*Create macro variable equal to number of obs with missing AV;

Data bnum;
set bmiss;
call symput('bnum',left(_freq_));

*Macro function to do imputation;

%macro BImpute(b=);
```

```
*Loop which starts with first obs and ends with last one;

%do %until (&b gt &bnum);

*Pull the next obs with missing data;

Data two;
Set bmissing;
If _N_ =&b;

*Merge Donor data set with missing data set;
*Keep only Obs from Donor data set with similar characteristics;

Data threea;
Merge ifmly&x Two; By Flag;
if own ne q3p2 and q3p2 in (1,2,3)
then delete;                            *Requires exact match for tenure;

if bld ne q3p1 and q3p1 in (1,2,3,5)
then delete;                             *Requires exact match for bld
type;

if over64popsur ne over64 then delete; *Requires presence of person over64;

if hhinc <100000 then do;
   if fincbtax gt (hhinc +15000)
   or fincbtax lt (hhinc-15000) then delete; *Requires income to be within
15,000 if income below 100,000;
   end;
if 100000<= hhinc <= 130000 then do;
      if fincbtax gt (hhinc+25000)
      or fincbtax lt (hhinc-25000) then delete;*Requires income to be within
25,000 if income above 100,000 and below 130,000;
      end;
if hhinc > 130000 then do;
      if fincbtax gt (hhinc+35000)
      or fincbtax lt (hhinc-35000) then delete;*Requires income to be within
35,000 if income above 130,000;
      end;

if fam_size gt (peopl+1)
or fam_size lt (peopl-1) then delete; *Requires household size to be within
one;

if under18 ne WAsurvey18 then delete;      *Requires households to have
children under 18;

_type_=0; *Create variable for merging with output of a proc sum;


*Count number and total weight of candidate donors;

proc summary data =threea;
var finlwt21;
output out=threeb sum=wgttotal;
run;
```

```
*Select 7 donors for multiple imputation;

%do i = 1 %to 7 %by 1;

*Calc a score equal to wgt of each donor relative to all donors in this
group. Spread the scores over the interval 0 to 1 with width equal
to each donor's relative weight. Pick a random number between 0 and 1
and select a donor;
*Repeat 7 times;

data three;
merge threea threeb;
by _type_;
rename _freq_ = total;
drop _type_;
score = finlwt21/wgttotal;
if _n_=1 then
a = ranuni(0);
retain a;
retain score0 score1 0;
score0 = score1;
score1 + score;
newid&i = newid;
Keep ID newid&i _freq_;
if score0< a <= score1 then output three;


*Add imputed value to 2st round data set;

Data  miimpute.BIImpute&x;
merge miimpute.BIImpute&x three;
by ID;
I_round='b';
attrib I_round label ='Imputation Round' format =$1.0;
%end; *Ends multiple imputation rounds;
%let b=%eval(&b+1);*Increments 'b' to next missing obs;
%end;
%mend BImpute;
%BImpute(b=1);
run;

*---------------------------------------------------------------------------;
*THIRD ROUND OF IMPUTATON - eliminate bld type restriction
and relax family size restriction;
*---------------------------------------------------------------------------;

*Select obs where less than 1 match was found in second round;

Data c;
set miimpute.BIImpute&x;
Where total lt 1;
run;
Data cmissing;
merge c(in=in1) missing(in=in2) ;
by id; if in1;
```

```
run;

*Count the number of obs with missing AV;

proc means data = cmissing noprint;
var flag;
output out=cmiss sum=total;
run;

*Delete file which will contain imputed values;

Proc delete data=miimpute.CIImpute&x;
run;

*Create file which will contain imputed values;

Data miimpute.CIImpute&x;
set cmissing;
keep ID;
Proc sort;
by ID;
run;

*Create macro variable equal to number of obs with missing data;

Data bnum;
set cmiss;
call symput('cnum',left(_freq_));

*Macro function to do imputation;

%macro CImpute(c=);

*Loop which starts with first obs with missing AV and ends with last one;

%do %until (&c gt &cnum);

*Pull the next obs with missing data;

Data two;
Set cmissing;
If _N_ =&c;

*Merge Donor data set with obs with missing data;
*Keep only Obs from Donor data set with similar characteristics;

Data threea;
Merge ifmly&x Two; By Flag;
if own ne q3p2 and q3p2 in (1,2,3)
then delete;                              *Requires exact match for tenure;

*if bld ne q3p1 and q3p1 in (1,2,3,5)
then delete;                               *Requires exact match for bld
type;

if over64popsur ne over64 then delete; *Requires presence of person over64;
if hhinc <100000 then do;
   if fincbtax gt (hhinc +15000)
   or fincbtax lt (hhinc-15000) then delete; *Requires income to be within
15,000 if income below 100,000;
   end;
if 100000<= hhinc <= 130000 then do;
      if fincbtax gt (hhinc+25000)
      or fincbtax lt (hhinc-25000) then delete;*Requires income to be within
25,000 if income above 100,000 and below 130,000;
      end;
if hhinc > 130000 then do;
      if fincbtax gt (hhinc+35000)
      or fincbtax lt (hhinc-35000) then delete;*Requires income to be within
35,000 if income above 130,000;
      end;

if fam_size gt (peopl+2)
or fam_size lt (peopl-2) then delete; *Requires household size to be within
two;

if under18 ne WAsurvey18 then delete;      *Requires households to have
children under 18;

_type_=0; *Create variable for merging with output of a proc sum;


*Count number and total weight of candidate donors;

proc summary data =threea;
var finlwt21;
output out=threeb sum=wgttotal;
run;


*Select 7 donors for multiple imputation;

%do i = 1 %to 7 %by 1;

*Calc a score equal to wgt of each donor relative to all donors in this
group. Spread the scores over the interval 0 to 1 with width equal
to each donor's relative weight. Pick a random number between 0 and 1
and select a donor;
*Repeat 7 times;

data three;
merge threea threeb;
by _type_;
rename _freq_ = total;
drop _type_;
score = finlwt21/wgttotal;
if _n_=1 then
a = ranuni(0);
retain a;
retain score0 score1 0;
score0 = score1;
score1 + score;
newid&i = newid;
```

```
Keep ID newid&i _freq_;
if score0< a <= score1 then output three;

*Add imputed value to 3rd round data set;

Data  miimpute.CIImpute&x;
merge miimpute.CIImpute&x three;
by ID;
I_round='c';
attrib I_round label ='Imputation Round' format =$1.0;
%end; *Ends multiple imputation rounds;
%let c=%eval(&c+1);*Increments 'c' to next missing obs;
%end;
%mend CImpute;
%CImpute(c=1);
run;

%end;
%mend Loop;
%Loop;
run;

*-------------------------------------------------------------------------;
*Combine the data sets from the 3 imputation rounds;
*Drop any obs without a match;
*-------------------------------------------------------------------------;

*Delete file which will contain imputed values;
%macro loop1;
%do x=1 %to 4 %by 1;
Proc delete data=miimpute.IImpute&x;
run;

data miimpute.IImpute&x;
merge miimpute.AIImpute&x miimpute.BIImpute&x miimpute.CIImpute&x;
by id;
if newid&x = . then delete;
run;
%end;
%mend Loop1;
%loop1;

*-------------------------------------------------------------------------;
*Determine number of quarters that each id has a match;
*-------------------------------------------------------------------------;

data miimpute.iqtrsmatched;
merge miimpute.iimpute1(in=a) miimpute.iimpute2(in=b)
miimpute.iimpute3(in=c) miimpute.iimpute4(in=d);
by id;
if a then qtr1=1; else qtr1=0;
if b then qtr2=1; else qtr2=0;
if c then qtr3=1; else qtr3=0;
if d then qtr4=1; else qtr4=0;
Dqtrs=qtr1+qtr2+qtr3+qtr4;
keep id Dqtrs;
proc sort data=miimpute.iqtrsmatched;
```

```
by id;
run;

options notes source source2 errors=20;

*----------------------------------------------------------------------------;
*Check to see if a newid is missing when other iterations contain values;
*----------------------------------------------------------------------------;

proc delete data=test;
run;
%macro loop2;
%do w=1 %to 4 %by 1;
data zzz;
set miimpute.AIImpute&w;
if newid1 = . and sum(newid2, newid3, newid4, newid5, newid6,
newid7)>0 then output;
if newid2 = . and sum(newid1, newid3, newid4, newid5, newid6,
newid7)>0 then output;
if newid3 = . and sum(newid2, newid1, newid4, newid5, newid6,
newid7)>0 then output;
if newid4 = . and sum(newid2, newid3, newid1, newid5, newid6,
newid7)>0 then output;
if newid5 = . and sum(newid2, newid3, newid4, newid1, newid6,
newid7)>0 then output;
if newid6 = . and sum(newid2, newid3, newid4, newid5, newid1,
newid7)>0 then output;
if newid7 = . and sum(newid2, newid3, newid4, newid5, newid6,
newid1)>0 then output;
run;
proc append base = test Data=zzz;
run;
%end;
%mend loop2;
%loop2;
```

[i] These programs were adapted from James W. McNally's paper – "Generating Hot-Deck Inputation Estimates: Using SAS for Simple and Multiple Imputation Allocation Routines", PSTC Working Paper #97-12 , Sept 1997, Population Studies and Training Center, Brown University.