

Calibration_Readme.txt

Purpose: Calibrate indicators for health conditions for community sample for 5 multiply imputed datasets

Run programs in the following order:

1. calibr1_mcbs09.SAS - calibrate conditions with self-report (SR) in NHANES.
2. calibr2_mcbs09.SAS - calibrate conditions without SR in NHANES.
3. calibr3_mcbs09.SAS - disentangle cerebrovascular disease 57_58_59 into 3 distinct indicators, corresponding to hemorrhagic, ischemic stroke and TIA
4. save_anal_mcbs09.SAS - Create table of prevalence estimates of SR in NHANES and SR, calibrated and claim-based prevalences in MCBS community population.

Additional code required:

1. mc_nh09.set
2. mc_mc09.set

IVEWare is required to run these data processing steps.

```
/******  
Program: 'calibr1_mcbs09.sas'  
Purpose: Calibrate conditions where self-report disease variable is  
available in NHANES  
Data in: Insert path for input datasets  
Data out: Insert path for output dataset  
*****/  

```

```
Libname g "Insert file path";  
%include "Insert file path\xdelete.sas";  
%include "Insert file path\formats.sas";  
%xdelete(_ALL_);  
options symbolgen mprint ;
```

```
proc printto;  
Run;  
%xdelete(_ALL_);
```

```
%macro strings;  
%global me_mcbs;  
%global me_nh;  
%global mcbs_nh;  
%global nhanes_sr;  
%global meps;  
%global mcbs;  
%global me_mcbs_sr;  
%global me_nh_sr;  
%global mcbs_nh_sr;  
%global cat_me_mcbs_sr;  
%global cat_me_nh_sr;  
%global cat_mcbs_nh_sr;  
%global mcbs_sr;  
%global cat_mcbs_sr;  
%global sr_mcbs;  
%global sr_meps;  
  
%global n_me_mcbs;  
%global n_me_nh;  
%global n_mcbs_nh;  
%global numcalibr;  
%global nsr_mcbs;  
%global claim_me1;  
%global claim_me2;  
%global nclaim_me1;  
%global nclaim_me2;  
%global claim_me3;
```

```

%global nclaim_me3;

%global cgar_nhanes;
%global cgar_me_mc;
%global nclaim_me;
%global ncggar_nhanes;
%global ncggar_me_mc;
%global nsr_meps;
%global nsr_me_nh;
%global nsr_mcbs_nh;
%global sr_me_nh;
%global sr_mcbs_nh;

%let me_mcbs=;
%let me_nh=;
%let mcbs_nh=;
%let nhanes_sr=;
%let meps=;
%let mcbs=;
%let me_mcbs_sr=;
%let me_nh_sr=;
%let mcbs_nh_sr=;
%let cat_me_mcbs_sr=;
%let cat_me_nh_sr=;
%let cat_mcbs_nh_sr=;
%let numcalibr=;
%let mcbs_sr=;
%let sr_mcbs=;
%let sr_meps=;

%let nsr_mcbs=;
%let nsr_meps=;

%let cat_mcbs_sr=;
%let claim_me1=;
%let claim_me2=;
%let claim_me3=;

%let cgar_nhanes=;
%let cgar_me_mc=;
%let nsr_me_nh=;
%let nsr_mcbs_nh=;
%let sr_me_nh=;
%let sr_mcbs_nh=;
proc contents data=g.MCBS_ni_i09 (drop = type faccost instcost ) out=mcbs;
Run;

```

```
proc contents data=g.meps65_i09 out=meps;  
Run;
```

```
proc contents data=g.nhanes65_i0910 out=nhanes;  
Run;
```

```
Proc sort data=g.claim_meps09 out=meps_claims05;  
by dupersid;  
Run;
```

```
Data meps_claims05;  
  Set meps_claims05;  
  cgar57_58=max(cgar57,cgar58);  
drop cgar26 cgar28 cgar29 cgar30 cgar35 cgar53 cgar54 cgar81 cgar83 cgar78  
cgar79 cgar100 cgar101 cgar58 ;  
Run;
```

```
proc contents data=meps_claims05 out=claim_me;  
Run;
```

```
Data claim_me2;  
  Set claim_me;  
  var=lowcase(name);  
  If substr(var,1,4)='cgar';  
  if var not in('cgar57_58');  
Run;
```

```
Data claim_me1;  
  Set claim_me;  
  var=lowcase(name);  
  If substr(var,1,4)='cgar';  
  if var not in('cgar57_58', /*the following 4 CGARS are year  
specific*/'cgar1', 'cgar3', 'cgar24', 'cgar84', 'cgar86', 'cgar94');  
Run;
```

```
Data claim_me3;  
  Set claim_me;  
  var=lowcase(name);  
  If substr(var,1,4)='cgar';  
Run;
```

```
Data meps;  
  Set meps;  
  var=lowcase(name);  
Run;
```

```

Data mcbs;
  Set mcbs;
  var=lowercase(name);
Run;

```

```

Data nhanes;
  Set nhanes;
  var=lowercase(name);
Run;

```

```

Proc sort data=meps;
by var;
Proc sort data=mcbs;
by var;
Proc sort data=nhanes;
by var;
Data g.comcov;
Merge nhanes(rename=(memname=nha)) meps(rename=(memname=meps))
mcbs(rename=(memname=mcbs));
by var;
keep var nha mcbs meps;
Run;

```

```

Data me_mcbs me_nh mcbs_nh sr_mcbs nhanes_sr meps mcbs me_mcbs_sr me_nh_sr
mcbs_nh_sr mcbs_sr sr_meps sr_me_nh sr_mcbs_nh;
  Set g.comcov;
  If var not in('_mult_', 'baseid', 'psu', 'strat', 'wgt');
  If meps^='' and mcbs^='' then output me_mcbs;
  If meps^='' and nha^='' then output me_nh;
  If nha^='' and mcbs^='' and var not in('cgarsr82', 'cgarsr89', 'cgarsr8')
then output mcbs_nh;
  If nha^='' and substr(var,1,6)='cgarsr' then output nhanes_sr;
  If mcbs^='' and substr(var,1,6)='cgarsr' then output sr_mcbs;
  If meps^='' and substr(var,1,6)='cgarsr' then output sr_meps;

  If meps^='' then output meps;
  If mcbs^='' then output mcbs;
  If mcbs^='' and substr(var,1,6)^='cgarsr' then output mcbs_sr;
  If meps^='' and mcbs^='' and substr(var,1,6)^='cgarsr' then output
me_mcbs_sr;

  If meps^='' and nha^='' and substr(var,1,6)^='cgarsr' then output
me_nh_sr;
  If nha^='' and mcbs^='' and substr(var,1,6)^='cgarsr'

```

```

    and var not in('asthma_emphysema') then output mcbs_nh_sr;

    If meps^='' and nha^='' and substr(var,1,6)='cgarsr' then output sr_me_nh;
    If nha^='' and mcbs^='' and substr(var,1,6)='cgarsr' and var not
in('cgarsr82','cgarsr89','cgarsr8') then output sr_mcbs_nh;
Run;

data _null_;
    Set me_mcbs;
    call symput('me_mcbs', trim(resolve('&me_mcbs'))||' '||trim(var));
run;

data _null_;
    Set me_nh;
    call symput('me_nh', trim(resolve('&me_nh'))||' '||trim(var));
run;

data _null_;
    Set mcbs_nh;
    call symput('mcbs_nh', trim(resolve('&mcbs_nh'))||' '||trim(var));
run;

data _null_;
    Set nhanes_sr;
    call symput('nhanes_sr', trim(resolve('&nhanes_sr'))||'
' ||trim(var));
run;

data _null_;
    Set mcbs;
    call symput('mcbs', trim(resolve('&mcbs'))||' '||trim(var));
run;

data _null_;
    Set meps;
    call symput('meps', trim(resolve('&meps'))||' '||trim(var));
run;
data _null_;
    Set me_mcbs_sr;
    call symput('me_mcbs_sr', trim(resolve('&me_mcbs_sr'))||' '||trim(var));
run;

data _null_;
    Set me_nh_sr;
    call symput('me_nh_sr', trim(resolve('&me_nh_sr'))||' '||trim(var));
run;

```

```

data _null_;
  Set sr_me_nh;
  call symput('sr_me_nh', trim(resolve('&sr_me_nh'))||' '||trim(var));
run;

data _null_;
  Set mcbs_nh_sr;
  call symput('mcbs_nh_sr', trim(resolve('&mcbs_nh_sr'))||'
'||trim(var));
run;

data _null_;
  Set sr_mcbs_nh;
  call symput('sr_mcbs_nh', trim(resolve('&sr_mcbs_nh'))||'
'||trim(var));
run;

data _null_;
  Set mcbs_sr;
  call symput('mcbs_sr', trim(resolve('&mcbs_sr'))||' '||trim(var));
run;

data _null_;
  Set sr_mcbs;
  call symput('sr_mcbs', trim(resolve('&sr_mcbs'))||' '||trim(var));
run;

data _null_;
Set sr_meps;
  call symput('sr_meps', trim(resolve('&sr_meps'))||' '||trim(var));
run;

Proc sql;
  Select n(var)into:n_me_mcbs from me_mcbs;
Quit;

Proc sql;
  Select n(var)into:n_me_nh from me_nh;
Quit;

Proc sql;
  Select n(var)into:n_mcbs_nh from mcbs_nh;
Quit;

Proc sql;

```

```

        Select n(var)into:numcalibr from nhanes_sr;
Quit;

Proc sql;
    Select n(var)into:nsr_mcbs from sr_mcbs;
Quit;

Proc sql;
    Select n(var)into:nsr_meps from sr_meps;
Quit;

Proc sql;
    Select n(var)into:nsr_mcbs_nh from sr_mcbs_nh;
Quit;

Proc sql;
    Select n(var)into:nsr_me_nh from sr_me_nh;
Quit;

Proc freq data=g.nhanes65_i0910 nlevels;
table &mcbs_nh_sr;
ods output nlevels=mcbs_nh_sr;
Run;

Data mcbs_nh_sr1;
    Set mcbs_nh_sr;
    catvar=tablevar;
    If nlevels<=4;
Run;

data _null_;
    Set mcbs_nh_sr1;
        call symput('cat_mcbs_nh_sr', trim(resolve('&cat_mcbs_nh_sr'))||'
'||trim(catvar));
run;

Proc freq data=g.nhanes65_i0910 nlevels;
table &me_nh_sr;
ods output nlevels=me_nh_sr;
Run;

Data me_nh_sr1;
    Set me_nh_sr;
    catvar=tablevar;
    If nlevels<=4;
Run;

```



```

data _null_;
  Set me_nh_sr1;
  call symput('cat_me_nh_sr', trim(resolve('&cat_me_nh_sr'))||'
'||trim(catvar));
run;

Proc freq data=g.MCBS_ni_i09 nlevels;
table &me_mcbs_sr;
ods output nlevels=me_mcbs_sr;
Run;

Data me_mcbs_sr1;
  Set me_mcbs_sr;
  catvar=tablevar;
  If nlevels<=4;
  Run;

data _null_;
  Set me_mcbs_sr1;
  call symput('cat_me_mcbs_sr', trim(resolve('&cat_me_mcbs_sr'))||'
'||trim(catvar));
run;

Proc freq data=g.MCBS_ni_i09 nlevels;
table &mcbs_sr;
ods output nlevels=mcbs_sr;
Run;

Data mcbs_sr1;
  Set mcbs_sr;
  catvar=tablevar;
  If nlevels<=4;
  Run;

data _null_;
  Set mcbs_sr1;
  call symput('cat_mcbs_sr', trim(resolve('&cat_mcbs_sr'))||'
'||trim(catvar));
run;

*****Creates a list of diseases to be calibrated to
NHANES*****;
data _null_;
  Set claim_me2;
  call symput('claim_me2', trim(resolve('&claim_me2'))||' '||trim(var));

```

```

run;

Proc sql;
  Select n(var)into:nclaim_me2 from claim_me2;
  Quit;

data _null_;
  Set claim_me3;
  call symput('&claim_me3', trim(resolve('&claim_me3'))||' '||trim(var));
run;

Proc sql;
  Select n(var)into:nclaim_me3 from claim_me3;
  Quit;

data _null_;
  Set claim_me1;
  call symput('&claim_me1', trim(resolve('&claim_me1'))||' '||trim(var));
run;

Proc sql;
  Select n(var)into:nclaim_me1 from claim_me1;
  Quit;

Data cgar_nhanes;
  Set nhanes_sr;
  var='cgar' || substr(var,7);
run;

data _null_;
  Set cgar_nhanes;
  call symput('&cgar_nhanes', trim(resolve('&cgar_nhanes'))||'
' ||trim(var));
run;

Proc sql;
  Select n(var)into:ncgar_nhanes from cgar_nhanes;
  Quit;

*****Creates a list of MEPS diseases to be calibrated to
MCBS*****;
proc sort data=claim_me2;
by var;
proc sort data=cgar_nhanes;
by var;
Data cgar_me_mc;

```

```

Merge claim_me2(in=a) cgar_nhanes(in=b);
by var;
If a and not b;
If var^='cgar57_58';
Run;

data _null_;
  Set cgar_me_mc;
  call symput('cgar_me_mc', trim(resolve('&cgar_me_mc'))||' '||trim(var));
run;

Proc sql;
  Select n(var)into:ncgar_me_mc from cgar_me_mc;
  Quit;

%put me_mcbs=&me_mcbs;
%put n_me_mcbs=&n_me_mcbs;

%put me_nh= &me_nh;
%put n_me_nh= &n_me_nh;

%put mcbs_nh=&mcbs_nh;
%put n_mcbs_nh=&n_mcbs_nh;

%put nhanes_sr= &numcalibr &nhanes_sr;
%put mcbs_sr=&mcbs_sr;
%put cat_mcbs_sr=&cat_mcbs_sr;

%put meps=&meps;
%put mcbs=&mcbs;
%put me_mcbs_sr=&me_mcbs_sr;
%put me_nh_sr=&me_nh_sr;
%put mcbs_nh_sr=&mcbs_nh_sr;
%put cat_me_mcbs_sr=&cat_me_mcbs_sr;
%put cat_me_nh_sr=&cat_me_nh_sr;
%put cat_mcbs_nh_sr=&cat_mcbs_nh_sr;
%put sr_mcbs=&sr_mcbs;
%put claim_me1=&claim_me1;
%put claim_me2=&claim_me2;
%put claim_me3=&claim_me3;

%put cgar_nhanes=&cgar_nhanes;
%put ncgar_me_mc=&ncgar_me_mc;

%put cgar_me_mc=&cgar_me_mc;
%put sr_meps=&sr_meps;

```

```
%put sr_me_nh=&sr_me_nh;
%put sr_mcbs_nh=&sr_mcbs_nh;
%mend;
```

```
%strings;
```

```
Proc sort data=g.MCBS_ni_i09 out=MCBS_ni0/*(drop=faccost instcost)*/;
by baseid;
```

```
Proc sort data=g.mcbs_ni09 out=MCBS_claims09(keep= baseid cgar1-cgar98
cgar100 cgar102-cgar105 cgar122-cgar125);
```

```
by baseid;
```

```
Data mcbs65;
```

```
  Merge MCBS_ni0(in=a) MCBS_claims09;
```

```
  by baseid;
```

```
  If a;
```

```
  If cgar57=1 or cgar58=1 then cgar57=1;
```

```
  Else cgar57=0;
```

```
  cgar58=0;
```

```
  Run;
```

```
Data al;
```

```
  Set g.nhanes65_i0910(in=a) mcbs65(in=b);
```

```
  If a then donor=1;
```

```
  If b then donor=0;
```

```
  age_cont=age;
```

```
if 75>age_cont >= 65 then age= 1;
```

```
  else if 85>age_cont >= 75 and age< 85 then age= 2;
```

```
  else if age_cont>= 85 then age= 3;
```

```
  Run;
```

```
proc printto log='Insert file path\calibrated1_09.log'
```

```
  print='Insert file path\calibrated1_09.output';
```

```
%macro calibr1(numiter=,nimp=,numcgars=);
```

```
Data all(keep=baseid donor wgt psu strat
```

```
cgar1-cgar98 cgar100 cgar102-cgar105 cgar122-cgar125 mcgar1-mcgar98 mcgar100
```

```
mcgar102-mcgar105 mcgar122-mcgar125
```

```
&mcbs &nhanes_sr _mult_ age_cont);
```

```
  Set al;
```

```
  If donor=1 then do;
```

```
  %Do n=1 %to &numcgars;
```

```
  %IF &n ne 99 and &n ne 101 and &n ne 106 and &n ne 107 and &n ne 108 and &n
ne 109 and &n ne 110 and &n ne 111 and
```

```

&n ne 112 and &n ne 113 and &n ne 114 and &n ne 115
and &n ne 116 and &n ne 117 and &n ne 118 and &n ne 119 and /*added PI
5/31/2017*/
&n ne 120 and &n ne 121 %then %do;
  cgar&n=cgarsr&n;
%End;
%End;
End;

%Do n=1 %to &numcgars;
%IF &n ne 99 and &n ne 101 and &n ne 106 and &n ne 107 and &n ne 108 and &n
ne 109 and &n ne 110 and &n ne 111 and
&n ne 112 and &n ne 113 and &n ne 114 and &n ne 115
and &n ne 116 and &n ne 117 and &n ne 118 and &n ne 119 and /*added PI
5/31/2017*/
&n ne 120 and &n ne 121 %then %do;
  If cgar&n=. and donor=0 then cgar&n=0;
  If cgar&n=0 then mcgar&n=0;
  Else mcgar&n=1;
%End;
%End;
If male=1 then do;
  cgar10=0;
  cgar9=0;
  mcgar10=2;
  mcgar9=2;
End;
If male=0 then do;
  cgar11=0;
  mcgar11=2;
End;
cgar58=0;
Run;

%do iter=1 %to &numiter;

%do n=1 %to &numcalibr;
%let cg=%substr(%qscan(&nhanes_sr,&n),7);
%let cigorder=%sysfunc(findw(&sr_mcbs_nh, cgarsr&cg, ' ', 'I E'));

Data imp&cg&&iter;
  Set all;
  If (mcgar&cg=0 and donor=0) then cgar&cg=.;
%if &cg=17 or &cg=19 or &cg=50 %then %do;
  If donor=0 then do;
    If cgar%eval(&cg-1)=1 then do;

```

```

        mcgar&cg=2;
        cgar&cg=0;
    end;
    Else do;
        mcgar&cg=0;
        cgar&cg=.;
    End;
End;
%end;
Run;
proc sort data=imp&cg&iter;
by _mult_;
proc logistic data=imp&cg&iter;
by _mult_ ;
class male race marital;
model donor= &mcbs_nh_sr race*ed5 race*povcat race*age race*ed5
    %if( &iter^=1 or &n^=1) %then %do;
        costi
        %do b=1 %to %eval(&cg-1);
            %IF &b ne 99 and &b ne 101 and &b ne 106 and &b ne 107 and &b ne
108 and &b ne 109 and &b ne 110 and &b ne 111 and
                &b ne 112 and &b ne 113 and &b ne 114 and &b ne 115
and &b ne 116 and &b ne 117 and &b ne 118 and &b ne 119 and /*added PI
5/31/2017*/
                    &b ne 120 and &b ne 121 %then %do;
                cgar&b
            %end;
        %end;
    %end;
    %if &iter^=1 %then %do;
        %do j=%eval(&cg+1) %to &numcgars;
            %IF &j ne 99 and &j ne 101 and &j ne 106 and &j ne 107 and &j ne
108 and &j ne 109 and &j ne 110 and &j ne 111 and
                &j ne 112 and &j ne 113 and &j ne 114 and &j ne 115
and &j ne 116 and &j ne 117 and &j ne 118 and &j ne 119 and /*added PI
5/31/2017*/
                    &j ne 120 and &j ne 121 %then %do;
                cgar&j
            %end;
        %end;
    %end;
    /lackfit selection=backward slstay=.99;
ods output LackFitChiSq=g.mcbs09_hs&iter;
output out=PsNHA_&cg&iter xbeta=psnhanes ;
run ;

```

```

%macro calibrate(ngroup=);

proc rank data=PsNHA_&cg&&iter(drop=_level_) group=&ngroup out=PsNHA&cg&&iter
;
    by _mult_;
    var psnhanes ;
    ranks pscat ; *assigns propensity scores into quartiles ;
run ;

proc sort data=PsNHA&cg&&iter;
by _mult_ pscat ;
proc freq data=PsNHA&cg&&iter ;
by _mult_;
table pscat*donor*(&cat_mcbs_nh_sr )/cmh;
ods output cmh=bychi;
Run;

Data g.matchcat_mcbs03&iter;
Set bychi;
If statistic=3 and prob<0.05;
Run;

proc sort data=PsNHA&cg&&iter;
by _mult_ pscat;

run ; title " ";
proc means data=PsNHA&cg&&iter noprint missing;
    by _mult_ pscat ;
    var Mcgar&cg;
    where donor=0 and mcgar&cg^=2;
    weight wgt;
title "where donor =0" ;
output out=outstats0 mean=p0 n=freq0;
run ; title " ";

proc means data=PsNHA&cg&&iter noprint;
    by _mult_ pscat;
    var cgar&cg;
    weight wgt;
    where donor=1 and mcgar&cg^=2; ;
    title "where donor=1";
    output out=outstats1 mean=p1 n=freq1;
Run;

```

```

data g.stats&cg&&iter;
    merge outstats0 outstats1;
    by _mult_ pscat ;
    imputerate=(p1-p0)/(1-p0);
title " Strata-specific rates" ;
run ;title " ";

**imputation;
data clcgar&cg&&iter ;
    merge g.stats&cg&&iter PsNHA&cg&&iter(in=inranks);
    by _mult_ pscat ;
    if inranks ;
        If p0>=p1 then do;
            if donor=0 and cgar&cg = . then cgar&cg = 0;
        End;
    If p0<p1 then do;
        w=ranuni(-1);
        if donor=0 and cgar&cg =. then cgar&cg=(imputerate)>w;
    end ;
run;

%mend calibrate;
%calibrate(ngroup=5);

%do im=1 %to &nimp;
    %let impp=_&im;

    Data back;
    Set clcgar&cg&&iter;
    If cost=0 then lcost=0;
    Else lcost=log(cost);
    If _mult_=&im;
        imp=_mult_;
    If donor=1 then do;
        %do f=1 %to &numcgars;
            %if %sysfunc(indexw(&nhanes_sr, cgarsr&f))=0 %then %do;
                cgar&f=.;
            %end;
        %end;
    end;
;
    cgar58=0;
    drop p0 p1 w imputerate pscat costi;
Run;

```



```

options set = SRCLIB "Insert file path" sasautos = ('!SRCLIB' sasautos)
mautosource;
%impute(name=mc_nh09, dir=Insert file path\Calibrated, setup=old);

Data back&impp;
  Set back2;
  If lcost=0 then costi=0;
  Else costi=exp(lcost);
  If male=1 then do;
  cgar84=0;
  cgar85=0;
  End;
  drop lcost psnhanes;
Run;
%end;

data all;
  Set %do h=1 %to &nimp;
    back_&h
  %end;
;
  _mult_=imp;
  drop imp;
Run;

Data g.where_mcbs09;
  iter=&iter;
  cgar=&cg;
Run;

%xdelete(back_1 back_2 back_3 back_4 back_5 stat outstat1 outstat0 outstat3
PsNHA_&cg&&iter PsNHA&cg&&iter clcgar&cg&&iter imp&cg&&iter) ;

%If &n=&numcalibr %then %do;
proc sort data=all;
by _mult_ donor;

proc surveyfreq data=all missing;
by _mult_ donor;
  table cgar1-cgar98 cgar100 cgar102-cgar105 cgar122-cgar125;
  ods output OneWay=mepscl;
  strata STRat;
  cluster psu;
  weight WGT;
Run;

```

```

Data meps_clbr2(keep=cgar donor _mult_ percent stderr);
  Set mepscl;;
  cgar=substr(table,7);

%do kk=1 %to &numcgars;
  If cgar&kk=1 then output;
%end;

Run;

Proc sql;
  Create table parest_meps_clbr as
  select mean(percent) as percent, var(percent) as bvar, mean(stderr**2) as
wvar,cgar,donor
  from meps_clbr2
  group by cgar, donor ;
Quit;

Data parest_meps_clbr0(keep=cgar MICl_meps donor);
  Set parest_meps_clbr;
  mivar= wvar+bvar*6/5;
  mistderr=sqrt(mivar);
  MICl_meps=trim(left(put(percent,5.2)))||'
('||trim(left(put(mistderr,4.2)))||')');
Run;
proc sort data=parest_meps_clbr0 ;
by cgar;
proc transpose data=parest_meps_clbr0 out=tr&iter;
by cgar;
id donor;
var micl_meps;
Run;

proc print data=tr&iter;
Title "MCBS09: step 4(iter= &iter)";
Run;
  %end;

%If &n=&numcalibr and &iter=&numiter %then %do;
  data g.calibr1_MC_NH09;
  Set all;
Run;
%end;

%end;

```

```
%end;  
title " ";  
%mend calibr1;  
  
%calibr1(numiter=5,nimp=5,numcgars=125);
```

```
/******  
Program: 'calibr2_mcbs09.sas'  
Purpose: Calibrate conditions where self-report disease variable is NOT  
available in NHANES  
Data in: Insert path for input datasets  
Data out: Insert path for output dataset  
*****/  

```

```
libname claims "Insert file path" ;  
libname g "Insert file path";  
%include "Insert file path\xdelete.sas";  
%include "Insert file path\formats.sas";  
%xdelete(_ALL_);  
options set = SRCLIB "Insert file path\srclib" sasautos = (!SRCLIB'  
sasautos) mautosource;  
options symbolgen mprint ;
```

```
%macro strings;  
%global me_mcbs;  
%global me_nh;  
%global mcbs_nh;  
%global nhanes_sr;  
%global meps;  
%global mcbs;  
%global me_mcbs_sr;  
%global me_nh_sr;  
%global mcbs_nh_sr;  
%global cat_me_mcbs_sr;  
%global cat_me_nh_sr;  
%global cat_mcbs_nh_sr;  
%global mcbs_sr;  
%global cat_mcbs_sr;  
%global sr_mcbs;  
%global sr_meps;
```

```
%global n_me_mcbs;  
%global n_me_nh;  
%global n_mcbs_nh;  
%global numcalibr;  
%global nsr_mcbs;  
%global claim_me1;  
%global claim_me2;  
%global nclaim_me1;  
%global nclaim_me2;  
%global claim_me3;  
%global nclaim_me3;
```

```

%global cgar_nhanes;
%global cgar_me_mc;
%global nclaim_me;
%global ncgar_nhanes;
%global ncgar_me_mc;
%global nsr_meps;
%global nsr_me_nh;
%global nsr_mcbs_nh;
%global sr_me_nh;
%global sr_mcbs_nh;

%let me_mcbs=;
%let me_nh=;
%let mcbs_nh=;
%let nhanes_sr=;
%let meps=;
%let mcbs=;
%let me_mcbs_sr=;
%let me_nh_sr=;
%let mcbs_nh_sr=;
%let cat_me_mcbs_sr=;
%let cat_me_nh_sr=;
%let cat_mcbs_nh_sr=;
%let numcalibr=;
%let mcbs_sr=;
%let sr_mcbs=;
%let sr_meps=;

%let nsr_mcbs=;
%let nsr_meps=;

%let cat_mcbs_sr=;
%let claim_me1=;
%let claim_me2=;
%let claim_me3=;

%let cgar_nhanes=;
%let cgar_me_mc=;
%let nsr_me_nh=;
%let nsr_mcbs_nh=;
%let sr_me_nh=;
%let sr_mcbs_nh=;
proc contents data=g.MCBS_ni_i09 (drop = type faccost instcost ) out=mcbs;
Run;

```

```
proc contents data=g.meps65_i09 out=meps;  
Run;
```

```
proc contents data=g.nhanes65_i0910 out=nhanes;  
Run;
```

```
Proc sort data=g.claim_meps09 out=meps_claims05;  
by dustersid;  
Run;
```

```
Data meps_claims05;  
Set meps_claims05;  
cgar57_58=max(cgar57,cgar58);  
drop cgar26 cgar28 cgar29 cgar30 cgar35 cgar53 cgar54 cgar81 cgar83 cgar78  
cgar79 cgar100 cgar101 cgar58 ;  
Run;
```

```
proc contents data=meps_claims05 out=claim_me;  
Run;
```

```
Data claim_me2;  
Set claim_me;  
var=lowcase(name);  
If substr(var,1,4)='cgar';  
if var not in('cgar57_58');  
Run;
```

```
Data claim_me1;  
Set claim_me;  
var=lowcase(name);  
If substr(var,1,4)='cgar';  
if var not in('cgar57_58', /*the following 4 CGARS are year  
specific*/'cgar1', 'cgar3', 'cgar24', 'cgar84', 'cgar86', 'cgar94');  
Run;
```

```
Data claim_me3;  
Set claim_me;  
var=lowcase(name);  
If substr(var,1,4)='cgar';  
Run;
```

```
Data meps;  
Set meps;  
var=lowcase(name);  
Run;
```

```
Data mcbs;  
  Set mcbs;  
  var=lowercase(name);  
Run;
```

```
Data nhanes;  
  Set nhanes;  
  var=lowercase(name);  
Run;
```

```
Proc sort data=meps;  
by var;  
Proc sort data=mcbs;  
by var;  
Proc sort data=nhanes;  
by var;  
Data g.comcov;  
Merge nhanes(rename=(memname=nha)) meps(rename=(memname=meps))  
mcbs(rename=(memname=mcbs));  
by var;  
keep var nha mcbs meps;  
Run;
```

```
Data me_mcbs me_nh mcbs_nh sr_mcbs nhanes_sr meps mcbs me_mcbs_sr me_nh_sr  
mcbs_nh_sr mcbs_sr sr_meps sr_me_nh sr_mcbs_nh;  
  Set g.comcov;  
  If var not in('_mult_', 'baseid', 'psu', 'strat', 'wgt');  
  If meps^='' and mcbs^='' then output me_mcbs;  
  If meps^='' and nha^='' then output me_nh;  
  If nha^='' and mcbs^='' and var not in('cgarsr82', 'cgarsr89', 'cgarsr8')  
then output mcbs_nh;  
  If nha^='' and substr(var,1,6)='cgarsr' then output nhanes_sr;  
  If mcbs^='' and substr(var,1,6)='cgarsr' then output sr_mcbs;  
  If meps^='' and substr(var,1,6)='cgarsr' then output sr_meps;  
  
  If meps^='' then output meps;  
  If mcbs^='' then output mcbs;  
  If mcbs^='' and substr(var,1,6)^='cgarsr' then output mcbs_sr;  
  If meps^='' and mcbs^='' and substr(var,1,6)^='cgarsr' then output  
me_mcbs_sr;  
  
  If meps^='' and nha^='' and substr(var,1,6)^='cgarsr' then output  
me_nh_sr;  
  If nha^='' and mcbs^='' and substr(var,1,6)^='cgarsr'  
and var not in('asthma_emphysema') then output mcbs_nh_sr;
```

```
    If meps^='' and nha^='' and substr(var,1,6)='cgarsr' then output sr_me_nh;  
    If nha^='' and mcbs^='' and substr(var,1,6)='cgarsr' and var not  
in('cgarsr82','cgarsr89','cgarsr8') then output sr_mcbs_nh;
```

```
Run;
```

```
data _null_;  
    Set me_mcbs;  
    call symput('me_mcbs', trim(resolve('&me_mcbs'))||' '||trim(var));  
run;
```

```
data _null_;  
    Set me_nh;  
    call symput('me_nh', trim(resolve('&me_nh'))||' '||trim(var));  
run;
```

```
data _null_;  
    Set mcbs_nh;  
    call symput('mcbs_nh', trim(resolve('&mcbs_nh'))||' '||trim(var));  
run;
```

```
data _null_;  
    Set nhanes_sr;  
    call symput('nhanes_sr', trim(resolve('&nhanes_sr'))||'  
'||trim(var));  
run;
```

```
data _null_;  
    Set mcbs;  
    call symput('mcbs', trim(resolve('&mcbs'))||' '||trim(var));  
run;
```

```
data _null_;  
    Set meps;  
    call symput('meps', trim(resolve('&meps'))||' '||trim(var));  
run;
```

```
data _null_;  
    Set me_mcbs_sr;  
    call symput('me_mcbs_sr', trim(resolve('&me_mcbs_sr'))||' '||trim(var));  
run;
```

```
data _null_;  
    Set me_nh_sr;  
    call symput('me_nh_sr', trim(resolve('&me_nh_sr'))||' '||trim(var));  
run;
```



```

data _null_;
  Set sr_me_nh;
  call symput('sr_me_nh', trim(resolve('&sr_me_nh'))||' '||trim(var));
run;

data _null_;
  Set mcbs_nh_sr;
  call symput('mcbs_nh_sr', trim(resolve('&mcbs_nh_sr'))||'
' ||trim(var));
run;
data _null_;
  Set sr_mcbs_nh;
  call symput('sr_mcbs_nh', trim(resolve('&sr_mcbs_nh'))||'
' ||trim(var));
run;

data _null_;
  Set mcbs_sr;
  call symput('mcbs_sr', trim(resolve('&mcbs_sr'))||' ' ||trim(var));
run;

data _null_;
  Set sr_mcbs;
  call symput('sr_mcbs', trim(resolve('&sr_mcbs'))||' ' ||trim(var));
run;
data _null_;
Set sr_meps;
  call symput('sr_meps', trim(resolve('&sr_meps'))||' ' ||trim(var));
run;

Proc sql;
  Select n(var)into:n_me_mcbs from me_mcbs;
  Quit;
Proc sql;
  Select n(var)into:n_me_nh from me_nh;
  Quit;
Proc sql;
  Select n(var)into:n_mcbs_nh from mcbs_nh;
  Quit;

Proc sql;
  Select n(var)into:numcalibr from nhanes_sr;
  Quit;

Proc sql;
  Select n(var)into:nsr_mcbs from sr_mcbs;

```

```

Quit;

Proc sql;
  Select n(var)into:nsr_meps from sr_meps;
Quit;

Proc sql;
  Select n(var)into:nsr_mcbs_nh from sr_mcbs_nh;
Quit;

Proc sql;
  Select n(var)into:nsr_me_nh from sr_me_nh;
Quit;

Proc freq data=g.nhanes65_i0910 nlevels;
table &mcbs_nh_sr;
ods output nlevels=mcbs_nh_sr;
Run;

Data mcbs_nh_sr1;
  Set mcbs_nh_sr;
  catvar=tablevar;
  If nlevels<=4;
Run;

data _null_;
  Set mcbs_nh_sr1;
  call symput('cat_mcbs_nh_sr', trim(resolve('&cat_mcbs_nh_sr'))||'
'||trim(catvar));
run;

Proc freq data=g.nhanes65_i0910 nlevels;
table &me_nh_sr;
ods output nlevels=me_nh_sr;
Run;

Data me_nh_sr1;
  Set me_nh_sr;
  catvar=tablevar;
  If nlevels<=4;
Run;

data _null_;
  Set me_nh_sr1;
  call symput('cat_me_nh_sr', trim(resolve('&cat_me_nh_sr'))||'
'||trim(catvar));

```

```

run;

Proc freq data=g.MCBS_ni_i09 nlevels;
table &me_mcbs_sr;
ods output nlevels=me_mcbs_sr;
Run;

Data me_mcbs_sr1;
  Set me_mcbs_sr;
  catvar=tablevar;
  If nlevels<=4;
  Run;

data _null_;
  Set me_mcbs_sr1;
  call symput('cat_me_mcbs_sr', trim(resolve('&cat_me_mcbs_sr'))||'
'||trim(catvar));
run;

Proc freq data=g.MCBS_ni_i09 nlevels;
table &mcbs_sr;
ods output nlevels=mcbs_sr;
Run;

Data mcbs_sr1;
  Set mcbs_sr;
  catvar=tablevar;
  If nlevels<=4;
  Run;

data _null_;
  Set mcbs_sr1;
  call symput('cat_mcbs_sr', trim(resolve('&cat_mcbs_sr'))||'
'||trim(catvar));
run;

*****Creates a list of cgar to be calibrated to
NHANES*****;
data _null_;
  Set claim_me2;
  call symput('claim_me2', trim(resolve('&claim_me2'))||' '||trim(var));
run;

Proc sql;
  Select n(var)into:nclaim_me2 from claim_me2;
  Quit;

```

```
data _null_;
  Set claim_me3;
  call symput('claim_me3', trim(resolve('&claim_me3'))||' '||trim(var));
run;
```

```
Proc sql;
  Select n(var)into:nclaim_me3 from claim_me3;
  Quit;
data _null_;
  Set claim_me1;
  call symput('claim_me1', trim(resolve('&claim_me1'))||' '||trim(var));
run;
```

```
Proc sql;
  Select n(var)into:nclaim_me1 from claim_me1;
  Quit;
```

```
Data cgar_nhanes;
  Set nhanes_sr;
  var='cgar'||substr(var,7);
run;
```

```
data _null_;
  Set cgar_nhanes;
  call symput('cgar_nhanes', trim(resolve('&cgar_nhanes'))||'
'||trim(var));
run;
```

```
Proc sql;
  Select n(var)into:ncgar_nhanes from cgar_nhanes;
  Quit;
```

```
*****Creates a list of MEPS cgar to be calibrated to
MCBS*****;
```

```
proc sort data=claim_me2;
by var;
proc sort data=cgar_nhanes;
by var;
Data cgar_me_mc;
Merge claim_me2(in=a) cgar_nhanes(in=b);
by var;
If a and not b;
If var^='cgar57_58';
Run;
```

```

data _null_;
  Set cgar_me_mc;
  call symput('cgar_me_mc', trim(resolve('&cgar_me_mc'))||' '||trim(var));
run;

Proc sql;
  Select n(var)into:ncgar_me_mc from cgar_me_mc;
  Quit;

%put me_mcbs=&me_mcbs;
%put n_me_mcbs=&n_me_mcbs;

%put me_nh= &me_nh;
%put n_me_nh= &n_me_nh;

%put mcbs_nh=&mcbs_nh;
%put n_mcbs_nh=&n_mcbs_nh;

%put nhanes_sr= &numcalibr &nhanes_sr;
%put mcbs_sr=&mcbs_sr;
%put cat_mcbs_sr=&cat_mcbs_sr;

%put meps=&meps;
%put mcbs=&mcbs;
%put me_mcbs_sr=&me_mcbs_sr;
%put me_nh_sr=&me_nh_sr;
%put mcbs_nh_sr=&mcbs_nh_sr;
%put cat_me_mcbs_sr=&cat_me_mcbs_sr;
%put cat_me_nh_sr=&cat_me_nh_sr;
%put cat_mcbs_nh_sr=&cat_mcbs_nh_sr;
%put sr_mcbs=&sr_mcbs;
%put claim_me1=&claim_me1;
%put claim_me2=&claim_me2;
%put claim_me3=&claim_me3;

%put cgar_nhanes=&cgar_nhanes;
%put ncgar_me_mc=&ncgar_me_mc;

%put cgar_me_mc=&cgar_me_mc;
%put sr_meps=&sr_meps;
%put sr_me_nh=&sr_me_nh;
%put sr_mcbs_nh=&sr_mcbs_nh;
%mend;

%strings;

```

```

%macro calibr_no_nh_sr(datain=,ncgars=, ngroups=,nimp=);
Data clmcbs;
  Set &datain;

  %do f=1 %to &ncgars;
    %if %sysfunc(indexw(&nhanes_sr,cgarsr&f))=0 %then %do;
      cgar&f=.;
    %end;
  %end;
  %do mm=1 %to &ncgars;
    %IF &mm ne 99 and &mm ne 101 and &mm ne 106 and &mm ne 107 and
&mm ne 108 and &mm ne 109 and &mm ne 110 and &mm ne 111 and
      &mm ne 112 and &mm ne 113 and &mm ne 114 and &mm ne
115 and &mm ne 116 and &mm ne 117 and &mm ne 118 and &mm ne 119 and /*added
PI 5/31/2017*/
      &mm ne 120 and &mm ne 121 %then %do;
      rename cgar&mm=cgar_ccl&mm;;
    %end;
  %end;
age=age_cont;
if donor=0;
Run;
Data claim_mcbs09;
Set claims.claimb_mcbs09;;
cgar57u=max(0,cgar57);
Run;

Proc sort data=clmcbs;
by baseid;
Data see (keep=baseid _mult_ wgt psu strat &mcbs age_cont
cgar1-cgar98 cgar100 cgar102-cgar105 cgar122-cgar125 cgar_ccl1-cgar_ccl98
cgar_ccl100 cgar_ccl102-cgar_ccl105
cgar_ccl122-cgar_ccl125
cgar57u) ;
Merge clmcbs(in=a) claim_mcbs09;
by baseid;
if a;
Run;

data see;
set see;
If cgar57=1 or cgar58=1 then cgar57=1;
Else cgar57=0;
Run;
Data males females;
Set see;

```

```

calclaim=0;
claim=0;
p1=0;
If male=1 then output males;
If male=0 then output females;
keep _mult_ strat psu wgt baseid condition rank1 cost mark calclaim claim
p1;
Run;

```

```

%Do i=1 %to &ncgars;
  %IF &i ne 99 and &i ne 101 and &i ne 106 and &i ne 107 and &i ne 108
and &i ne 109 and &i ne 110 and &i ne 111 and
&i ne 112 and &i ne 113 and &i ne 114 and &i ne 115
and &i ne 116 and &i ne 117 and &i ne 118 and &i ne 119 and /*added PI
5/31/2017*/
&i ne 120 and &i ne 121 %then %do;

```

```

Data see&i;
Set see;
condition=&i;
claim=cgar&i;
calclaim=cgar_ccl&i;;
%if &i=102 or &i=105 %then %do;
  If male=0;
%end;
%if &i=104 %then %do;
  If male=1;
%end;
Run;

```

```

Data see&i;
Set see&i;
If claim=. then claim=0;
keep _mult_ &mcbs baseid strat wgt psu condition calclaim claim;
Run;

```

```

proc sort data=see&i;
by _mult_ ;
proc princomp data=see&i out=out_pc&i noprint;
by _mult_ ;
var &mcbs_sr ;
run;

```

```

Proc logistic data=out_pc&i;
by _mult_ ;
model claim=%do b=1 %to 30;
prin&b

```

```

                %end; /lackfit ;
ods output LackFitChiSq=lackfit&i;
Output out=out&i p=p1;
Run;
%end;
%end;

Data out;
    set
        %Do i=1 %to 16;
    out&i
        %end;
    out18
        %Do i=20 %to 49;
    out&i
        %end;
    %Do i=51 %to 57;
    out&i
        %end;
    %Do i=59 %to 98; /**exclude 99,101, 106-121, include 122-125*/
    out&i
        %end;
    out100
        %Do i=102 %to 105;
    out&i
        %end;
    %Do i=122 %to 125;
    out&i
        %end;
;
Run;

```

```

Proc sort data=out ;
by _mult_;
Proc rank data=out out=outt1 groups=&ngroups;
by _mult_;
var p1;
ranks rank1;
Run;

```

```

%xdelete(out);

```

```

proc sort data=outt1;
by _mult_ rank1;
proc freq data=outt1;
by _mult_ rank1;

```



```
table calclaim/out=promlem_&ngroups;
Run;
```

```
Data prb(keep=_mult_ rank1 mark);
  Set promlem_&ngroups;
  If percent=100 and calclaim=0;
  Mark=1;
Run;
```

```
data outt2;
  Merge outt1 prb;
  by _mult_ rank1;
  Run;
```

```
Data
%do k=1 %to &nimp;
%do j=1 %to &ngroups;
ps&k&&j
%end;
%end;
;
Set outt2;
  If mark=. then mark=0;
keep p1 claim calclaim rank1 _mult_ strat psu wgt baseid condition rank1 cost
mark;
%do k=1 %to &nimp;
%do j=1 %to &ngroups;
If _mult_=&k and rank1=%eval(&j-1) then output ps&k&&j;
%end;
%end;
Run;
```

```
%do n=1 %to &ngroups;
%do v=1 %to &nimp;
%let m=;
Data ps;
  Set ps&v&&n;
  Run;
```

```
proc sql;
  Select mean(mark) into: m
  from ps;
Quit;
```

```
%if &m=1 %then %do;
  Data ps;
```

```

        Set ps;
        If calclaim=. then calclaim=0;
Run;
%end;
%if &m=0 %then %do;

        %impute(name=mc_mc09, setup=old, dir=Insert file path\Calibrated);
%end;
%if &i=102 or &i=105 %then %do;
        Data f;
        Set f males;
        Run;
%end;
%if &i=104 %then %do;
        Data f;
        Set f females;
        Run;
%end;
Data f&v&&n;
        Set f;
        _mult_=&v;
Run;
%xdelete(ps f seelong );

%end;
%end;

Data al1;
        Set %do x=1 %to &ngroups;
                %do y=1 %to &nimp;
                        f&y&&x
                %end;
        %end;

;
Run;

Proc sort data=al1 out=al1;
by _mult_ baseid;
proc transpose data=al1 out=alli prefix=cgar_ccl;
var calclaim;
id condition;
by _mult_ baseid;
Run;

proc sort data=al1 out=al1;
by _mult_ condition;

```

```

proc surveyfreq data=all;
by _mult_ condition;
table claim calclaim;
ods output OneWay=psprev;
weight wgt;
strata strat;
cluster psu;
Run;

```

```

Data imp2(keep=_mult_ condition cgar percent stderr);
  Set psprev ;
  cgar=substr(table,7);
  If claim=1 then output;
  If calclaim=1 then output;
Run;

```

```

Proc sql;
  Create table parest as
  select mean(percent) as percent, var(percent) as bvar, mean(stderr**2) as
wvar,condition,cgar
  from imp2
  group by cgar,condition ;
Quit;

```

```

Data parest1(keep=cgar condition sumvar);
  Set parest;
  mivar= wvar+bvar*6/5;
  mistderr=sqrt(mivar);
  sumvar=trim(left(put(percent,5.2)))||'
('||trim(left(put(mistderr,4.2)))||')');
Run;

```

```

proc sort data=parest1;
by condition;
proc transpose data=parest1 out=estim;
var sumvar;
by condition;
id cgar;
Run;

```

```

ods rtf body='Insert file path\calibr2_mcbs09.rtf';
proc print;
run;
ods rtf close;

```

```

ods rtf close;

```

```

proc sort data=see;
  by _mult_ baseid;

Data g.Calibr2_mc_nh09a;
  Merge see alli;
  by _mult_ baseid;
  cgar57=cgar57u;
  %do b=1 %to &ncgars;
    If cgar&b=. then cgar&b=0;
  %end;
  drop cgar57u;
  cgar_ccl57_58=cgar_ccl57;
  drop cgar_ccl57 cgar_ccl58 ;
  label _mult_='Imputation #' cost='Cost' cgarsr57='Stroke';
  Run;
%mend;
  %calibr_no_nh_sr(datain=g.calibr1_MC_NH09,ncgars=125, ngroups=10,nimp=5);

proc means data=g.Calibr2_mc_nh09a;
var cgar1-cgar105 cgar_ccl1-cgar_ccl56 cgar_ccl59-cgar_ccl105;
Run;
Data g.Calibr2_mc_nh09;
set g.Calibr2_mc_nh09a;
Run;

```

```
/******  
Program: 'calibr3_mcbs09.sas'  
Purpose: Dissentagle cerebrovascular disease 57_58_59 into 3  
distict indicators, corresponding to hemorrhagic, ischemic stroke and TIA  
Data in: Insert path for input datasets  
Data out: Insert path for output dataset  
*****/  

```

```
Libname g "Insert file path";  
%include "Insert file path\xdelete.sas";  
%include "Insert file path\formats.sas";  
%xdelete(_ALL_);
```

```
%macro strings;
```

```
%global me_mcbs;  
%global me_nh;  
%global mcbs_nh;  
%global nhanes_sr;  
%global meps;  
%global mcbs;  
%global me_mcbs_sr;  
%global me_nh_sr;  
%global mcbs_nh_sr;  
%global cat_me_mcbs_sr;  
%global cat_me_nh_sr;  
%global cat_mcbs_nh_sr;  
%global mcbs_sr;  
%global cat_mcbs_sr;  
%global sr_mcbs;  
%global sr_meps;
```

```
%global n_me_mcbs;  
%global n_me_nh;  
%global n_mcbs_nh;  
%global numcalibr;  
%global nsr_mcbs;  
%global claim_me1;  
%global claim_me2;  
%global nclaim_me1;  
%global nclaim_me2;  
%global claim_me3;  
%global nclaim_me3;
```

```
%global cgar_nhanes;  
%global cgar_me_mc;  
%global nclaim_me;
```

```

%global nugar_nhanes;
%global nugar_me_mc;
%global nsr_meps;
%global nsr_me_nh;
%global nsr_mcbs_nh;
%global sr_me_nh;
%global sr_mcbs_nh;

%let me_mcbs=;
%let me_nh=;
%let mcbs_nh=;
%let nhanes_sr=;
%let meps=;
%let mcbs=;
%let me_mcbs_sr=;
%let me_nh_sr=;
%let mcbs_nh_sr=;
%let cat_me_mcbs_sr=;
%let cat_me_nh_sr=;
%let cat_mcbs_nh_sr=;
%let numcalibr=;
%let mcbs_sr=;
%let sr_mcbs=;
%let sr_meps=;

%let nsr_mcbs=;
%let nsr_meps=;

%let cat_mcbs_sr=;
%let claim_me1=;
%let claim_me2=;
%let claim_me3=;

%let cgar_nhanes=;
%let cgar_me_mc=;
%let nsr_me_nh=;
%let nsr_mcbs_nh=;
%let sr_me_nh=;
%let sr_mcbs_nh=;
proc contents data=g.MCBS_ni_i09 (drop = type faccost instcost ) out=mcbs;
Run;

proc contents data=g.meps65_i09 out=meps;
Run;

proc contents data=g.nhanes65_i0910 out=nhanes;

```

```
Run;
```

```
Proc sort data=g.claim_meps09 out=meps_claims05;  
by dupersid;  
Run;
```

```
Data meps_claims05;  
  Set meps_claims05;  
  cgar57_58=max(cgar57,cgar58);  
drop cgar26 cgar28 cgar29 cgar30 cgar35 cgar53 cgar54 cgar81 cgar83 cgar78  
cgar79 cgar100 cgar101 cgar58 ;  
Run;
```

```
proc contents data=meps_claims05 out=claim_me;  
Run;
```

```
Data claim_me2;  
  Set claim_me;  
  var=lowcase(name);  
  If substr(var,1,4)='cgar';  
  if var not in('cgar57_58');  
Run;
```

```
Data claim_me1;  
  Set claim_me;  
  var=lowcase(name);  
  If substr(var,1,4)='cgar';  
  if var not in('cgar57_58', /*the following 4 CGARS are year  
specific*/'cgar1', 'cgar3', 'cgar24', 'cgar84', 'cgar86', 'cgar94');  
Run;
```

```
Data claim_me3;  
  Set claim_me;  
  var=lowcase(name);  
  If substr(var,1,4)='cgar';  
Run;
```

```
Data meps;  
  Set meps;  
  var=lowcase(name);  
Run;
```

```
Data mcbs;  
  Set mcbs;  
  var=lowcase(name);  
Run;
```

```
Data nhanes;  
  Set nhanes;  
  var=lowercase(name);  
Run;
```

```
Proc sort data=meps;  
by var;  
Proc sort data=mcbs;  
by var;  
Proc sort data=nhanes;  
by var;  
Data g.comcov;  
Merge nhanes(rename=(memname=nha)) meps(rename=(memname=meps))  
mcbs(rename=(memname=mcbs));  
by var;  
keep var nha mcbs meps;  
Run;
```

```
Data me_mcbs me_nh mcbs_nh sr_mcbs nhanes_sr meps mcbs me_mcbs_sr me_nh_sr  
mcbs_nh_sr mcbs_sr sr_meps sr_me_nh sr_mcbs_nh;
```

```
  Set g.comcov;  
  If var not in('_mult_', 'baseid', 'psu', 'strat', 'wgt');  
  If meps^='' and mcbs^='' then output me_mcbs;  
  If meps^='' and nha^='' then output me_nh;  
  If nha^='' and mcbs^='' and var not in('cgarsr82', 'cgarsr89', 'cgarsr8')  
then output mcbs_nh;  
  If nha^='' and substr(var,1,6)='cgarsr' then output nhanes_sr;  
  If mcbs^='' and substr(var,1,6)='cgarsr' then output sr_mcbs;  
  If meps^='' and substr(var,1,6)='cgarsr' then output sr_meps;
```

```
  If meps^='' then output meps;  
  If mcbs^='' then output mcbs;  
  If mcbs^='' and substr(var,1,6)^='cgarsr' then output mcbs_sr;  
  If meps^='' and mcbs^='' and substr(var,1,6)^='cgarsr' then output  
me_mcbs_sr;
```

```
  If meps^='' and nha^='' and substr(var,1,6)^='cgarsr' then output  
me_nh_sr;
```

```
  If nha^='' and mcbs^='' and substr(var,1,6)^='cgarsr'  
  and var not in('asthma_emphysema') then output mcbs_nh_sr;
```

```
  If meps^='' and nha^='' and substr(var,1,6)='cgarsr' then output sr_me_nh;  
  If nha^='' and mcbs^='' and substr(var,1,6)='cgarsr' and var not  
in('cgarsr82', 'cgarsr89', 'cgarsr8') then output sr_mcbs_nh;
```



```

Run;

data _null_;
  Set me_mcbs;
  call symput('me_mcbs', trim(resolve('&me_mcbs'))||' '||trim(var));
run;

data _null_;
  Set me_nh;
  call symput('me_nh', trim(resolve('&me_nh'))||' '||trim(var));
run;

data _null_;
  Set mcbs_nh;
  call symput('mcbs_nh', trim(resolve('&mcbs_nh'))||' '||trim(var));
run;

data _null_;
  Set nhanes_sr;
  call symput('nhanes_sr', trim(resolve('&nhanes_sr'))||'
'||trim(var));
run;

data _null_;
  Set mcbs;
  call symput('mcbs', trim(resolve('&mcbs'))||' '||trim(var));
run;

data _null_;
  Set meps;
  call symput('meps', trim(resolve('&meps'))||' '||trim(var));
run;
data _null_;
  Set me_mcbs_sr;
  call symput('me_mcbs_sr', trim(resolve('&me_mcbs_sr'))||' '||trim(var));
run;

data _null_;
  Set me_nh_sr;
  call symput('me_nh_sr', trim(resolve('&me_nh_sr'))||' '||trim(var));
run;

data _null_;
  Set sr_me_nh;
  call symput('sr_me_nh', trim(resolve('&sr_me_nh'))||' '||trim(var));
run;

```

```

data _null_;
  Set mcbs_nh_sr;
  call symput('mcbs_nh_sr', trim(resolve('&mcbs_nh_sr'))||'
' ||trim(var));
run;

data _null_;
  Set sr_mcbs_nh;
  call symput('sr_mcbs_nh', trim(resolve('&sr_mcbs_nh'))||'
' ||trim(var));
run;

data _null_;
  Set mcbs_sr;
  call symput('mcbs_sr', trim(resolve('&mcbs_sr'))||' ' ||trim(var));
run;

data _null_;
  Set sr_mcbs;
  call symput('sr_mcbs', trim(resolve('&sr_mcbs'))||' ' ||trim(var));
run;

data _null_;
Set sr_meps;
  call symput('sr_meps', trim(resolve('&sr_meps'))||' ' ||trim(var));
run;

Proc sql;
  Select n(var)into:n_me_mcbs from me_mcbs;
Quit;

Proc sql;
  Select n(var)into:n_me_nh from me_nh;
Quit;

Proc sql;
  Select n(var)into:n_mcbs_nh from mcbs_nh;
Quit;

Proc sql;
  Select n(var)into:numcalibr from nhanes_sr;
Quit;

Proc sql;
  Select n(var)into:nsr_mcbs from sr_mcbs;

```

```

Quit;

Proc sql;
  Select n(var)into:nsr_meps from sr_meps;
Quit;

Proc sql;
  Select n(var)into:nsr_mcbs_nh from sr_mcbs_nh;
Quit;

Proc sql;
  Select n(var)into:nsr_me_nh from sr_me_nh;
Quit;

Proc freq data=g.nhanes65_i0910 nlevels;
table &mcbs_nh_sr;
ods output nlevels=mcbs_nh_sr;
Run;

Data mcbs_nh_sr1;
  Set mcbs_nh_sr;
  catvar=tablevar;
  If nlevels<=4;
Run;

data _null_;
  Set mcbs_nh_sr1;
  call symput('cat_mcbs_nh_sr', trim(resolve('&cat_mcbs_nh_sr'))||'
'||trim(catvar));
run;

Proc freq data=g.nhanes65_i0910 nlevels;
table &me_nh_sr;
ods output nlevels=me_nh_sr;
Run;

Data me_nh_sr1;
  Set me_nh_sr;
  catvar=tablevar;
  If nlevels<=4;
Run;

data _null_;
  Set me_nh_sr1;
  call symput('cat_me_nh_sr', trim(resolve('&cat_me_nh_sr'))||'
'||trim(catvar));

```

```

run;

Proc freq data=g.MCBS_ni_i09 nlevels;
table &me_mcbs_sr;
ods output nlevels=me_mcbs_sr;
Run;

Data me_mcbs_sr1;
  Set me_mcbs_sr;
  catvar=tablevar;
  If nlevels<=4;
  Run;

data _null_;
  Set me_mcbs_sr1;
  call symput('cat_me_mcbs_sr', trim(resolve('&cat_me_mcbs_sr'))||'
'||trim(catvar));
run;

Proc freq data=g.MCBS_ni_i09 nlevels;
table &mcbs_sr;
ods output nlevels=mcbs_sr;
Run;

Data mcbs_sr1;
  Set mcbs_sr;
  catvar=tablevar;
  If nlevels<=4;
  Run;

data _null_;
  Set mcbs_sr1;
  call symput('cat_mcbs_sr', trim(resolve('&cat_mcbs_sr'))||'
'||trim(catvar));
run;

*****Creates a list of diseases to be calibrated to
NHANES*****;
data _null_;
  Set claim_me2;
  call symput('claim_me2', trim(resolve('&claim_me2'))||' '||trim(var));
run;

Proc sql;
  Select n(var)into:nclaim_me2 from claim_me2;
  Quit;

```

```

data _null_;
  Set claim_me3;
  call symput('claim_me3', trim(resolve('&claim_me3'))||' '||trim(var));
run;

Proc sql;
  Select n(var)into:nclaim_me3 from claim_me3;
  Quit;

data _null_;
  Set claim_me1;
  call symput('claim_me1', trim(resolve('&claim_me1'))||' '||trim(var));
run;

Proc sql;
  Select n(var)into:nclaim_me1 from claim_me1;
  Quit;

Data cgar_nhanes;
  Set nhanes_sr;
  var='cgar' || substr(var,7);
run;

data _null_;
  Set cgar_nhanes;
  call symput('cgar_nhanes', trim(resolve('&cgar_nhanes'))||'
' ||trim(var));
run;

Proc sql;
  Select n(var)into:ncgar_nhanes from cgar_nhanes;
  Quit;

*****Creates a list of MEPS diseases to be calibrated to
MCBS*****;
proc sort data=claim_me2;
by var;
proc sort data=cgar_nhanes;
by var;
Data cgar_me_mc;
Merge claim_me2(in=a) cgar_nhanes(in=b);
by var;
If a and not b;
If var^='cgar57_58';
Run;

```

```
data _null_;
  Set cgar_me_mc;
  call symput('cgar_me_mc', trim(resolve('&cgar_me_mc'))||' '||trim(var));
run;
```

```
Proc sql;
  Select n(var)into:ncgar_me_mc from cgar_me_mc;
Quit;
```

```
%put me_mcbs=&me_mcbs;
%put n_me_mcbs=&n_me_mcbs;
```

```
%put me_nh= &me_nh;
%put n_me_nh= &n_me_nh;
```

```
%put mcbs_nh=&mcbs_nh;
%put n_mcbs_nh=&n_mcbs_nh;
```

```
%put nhanes_sr= &numcalibr &nhanes_sr;
%put mcbs_sr=&mcbs_sr;
%put cat_mcbs_sr=&cat_mcbs_sr;
```

```
%put meps=&meps;
%put mcbs=&mcbs;
%put me_mcbs_sr=&me_mcbs_sr;
%put me_nh_sr=&me_nh_sr;
%put mcbs_nh_sr=&mcbs_nh_sr;
%put cat_me_mcbs_sr=&cat_me_mcbs_sr;
%put cat_me_nh_sr=&cat_me_nh_sr;
%put cat_mcbs_nh_sr=&cat_mcbs_nh_sr;
%put sr_mcbs=&sr_mcbs;
%put claim_me1=&claim_me1;
%put claim_me2=&claim_me2;
%put claim_me3=&claim_me3;
```

```
%put cgar_nhanes=&cgar_nhanes;
%put ncgar_me_mc=&ncgar_me_mc;
```

```
%put cgar_me_mc=&cgar_me_mc;
%put sr_meps=&sr_meps;
%put sr_me_nh=&sr_me_nh;
%put sr_mcbs_nh=&sr_mcbs_nh;
%mend;
```

```
%strings;
```

```

Data mult;
  Set g.calibr2_mc_nh09;
  If cgar57=1 then dist57_58=1; /*'57'*/
  If cgar58=1 then dist57_58=2; /*'58'*/
  If cgar57=1 and cgar58=1 then dist57_58=3; /*'57_58'*/
  Run;

```

```

%macro split(cg=);
Proc sort data=mult;
  by _mult_;
  Proc freq data=mult;
  by _mult_;
  where cgar_ccl&cg=1;
  table dist57_58/out=dist&cg;
  Run;
Data c;
  do _mult_=1 to 5;
  do dist57_58=1 to 3;
  output;
  end;
  End;
  Run;
Proc sql;
  Select n(_mult_)/5 into:bign from mult;
Quit;

```

```

data n&cg dist&cg;;
merge c dist&cg;
by _mult_ dist&cg;
p=percent/100;
If dist&cg^=. then output dist&cg;
If dist&cg=. then output n&cg;
Run;

```

```

  Proc sort data=dist&cg ;
  by dist&cg;
  Proc transpose data=dist&cg out=g.tdist&cg prefix=p;;
  by dist&cg;
  id _mult_;
  var p;
  Run;

```

```

Proc transpose data=n&cg out=t_n&cg prefix=n;;
  id _mult_;
  var count;

```



```

End;
Else if cgar_ccl57_58=0 then do;
    cgar_ccl57=0;
    cgar_ccl58=0;
End;
Else if (cgar_ccl57_58=1 and dist57_58=.) then do;
    cgar_ccl57=c57;
    cgar_ccl58=c58;
End;
Run;

proc contents;
Run;

proc freq;
table cgar_ccl57_58*(cgar_ccl57 cgar_ccl58);
Run;

Data calibr3_mcbs09;
Set allclbr;
rename
/*exclude 99,101, 106-121, include 122-125*/
%do k=1 %to 98;
cgar&k=cgar_cl&k
%end;
cgar100=cgar_cl100

%do k=102 %to 105;
cgar&k=cgar_cl&k
%end;

%do k=122 %to 125;
cgar&k=cgar_cl&k
%end;
;
If male=1 then cgar_ccl86=0;
drop c57 c58 ;
Run;
Data g.calibr3_mcbs09;
Set calibr3_mcbs09;
If male=0 then do;
cgar_cl104=0 ;
Cgar_ccl04=0;
end;
If male=1 then do;
cgar_cl102=0 ;

```

```
Cgar_ccl102=0;
cgar_cl105=0 ;
Cgar_ccl105=0;
end;
```

```
***Self-report cgar labels***;
```

```
*MCBS, MEPS and NHANES;
```

```
label cgarsr16 = "Diabetes";
label cgarsr49 = "High blood pressure";
label cgarsr51 = "MI (heart attack)";
label cgarsr52 = "Coronary Atherosclerosis and other heart disease";
label cgarsr57 = "Stroke";
```

```
*MCBS, NHANES only;
```

```
label cgarsr6 = "Colon CA";
label cgarsr7 = "Lung CA";
label cgarsr8 = "Skin cancer";
label cgarsr9 = "Breast CA";
label cgarsr10 = "Cervical CA";
label cgarsr11 = "Prostate CA";
label cgarsr82 = "Prostate Hyperpasia";
label cgarsr88 = "Rheumatoid Arthritis";
label cgarsr89 = "Arthritis: non-rheumatoid";
label cgarsr91 = "Osteoporosis/brittle bones";
label cgarsr96 = "Broken or fractured a hip";
```

```
***Matching covariate labels***;
```

```
*MCBS, MEPS and NHANES;
```

```
label age = "Age (continuous)";
label male = "Gender: 1=male";
label race = "Race";
label ed5 = "Education";
label marital = "Marital status";
label hasjob = "Has job";
label didserv = "Served in Armed Forces";
label povcat = "Poverty Level Category";
label nbrpeopl = "Total number of people in household";
label inpatstays = "Inpatient stays";
label healthstat = "General health status";
label hearing = "Quality of hearing";
label smokenow = "Current smoker";
label havecare = "Have particular place for medical care";
label dif_lift = "Difficulty lifting/carrying 10 pounds";
label dif_walk = "Difficulty walking 1/4 mi";
label hyst = "Had Hysterectomy";
```

*MCBS, NHANES only;

```
label priv_insur = "Private Health Insurance";
label comphealth = "Health compared to 1 year ago";
label asthma_emphysema = "Asthma/Emphysema/COPD (combined in MCBS)";
label cgarsr43 = "Cataract";
label cgarsr18 = "SR HYperlipedimia";
label cgarsr28 = "SR Depression";
label cgarsr56 = "SR Congestive Heart Failure";
label everSmoke = "Ever smoked";
label weightkg = "Weight (kg)";
label height = "Height (cm)";
label pneushot = "Pneumonia vaccination";
label dif_stoop = "Difficulty stooping/crouching/kneeling";
label prb_dres = "Any difficulty dressing";
label prb_eat = "Any difficulty eating";
```

*MCBS, MEPS only;

```
label inpatnights = "Inpatient nights";
label cost = "Expenditure";
label mammogram = "Mammogram/breast X-ray in the last yr";
label pap_smear = "Pap smear in the last yr";
label flushot = "Flu shot last year";
```

*MCBS;

```
label cgarsr37 = "Parkinson's Disease";
label cgarsr39 = "Paralysis";
```

label

```
cgar_CL1 = 'Claim based cgar01 Tuberculosis '
cgar_CL2 = 'Claim based cgar02 STD non-HIV '
cgar_CL3 = 'Claim based cgar03 HIV '
cgar_CL4 = 'Claim based cgar04 Immunizations and screening for infectious
disease '
cgar_CL5 = 'Claim based cgar05 Other Infectious Disease '
cgar_CL6 = 'Claim based cgar06 Colon Cancer '
cgar_CL7 = 'Claim based cgar07 Lung Cancer '
cgar_CL8 = 'Claim based cgar08 Skin Cancer '
cgar_CL9 = 'Claim based cgar09 Breast Cancer '
cgar_CL10 = 'Claim based cgar10 Cervical Cancer '
cgar_CL11 = 'Claim based cgar11 Prostate Cancer '
cgar_CL12 = 'Claim based cgar12 Hematologic Cancer '
cgar_CL13 = 'Claim based cgar13 Benign Neoplasm '
cgar_CL14 = 'Claim based cgar14 Other Cancer '
cgar_CL15 = 'Claim based cgar15 Thyroid disorders '
cgar_CL16 = 'Claim based cgar16 Diabetes Mellitus '
cgar_CL17 = 'Claim based cgar17 Undiagnosed Diabetes Mellitus '
```

cgar_CL18 = 'Claim based cgar18 Hyperlipidemia '
 cgar_CL19 = 'Claim based cgar19 Undiagnosed Hyperlipidemia '
 cgar_CL20 = 'Claim based cgar20 Gout and other crystal arthropathies '
 cgar_CL21 = 'Claim based cgar21 Other Endocrine Diseases '
 cgar_CL22 = 'Claim based cgar22 Anemias '
 cgar_CL23 = 'Claim based cgar23 Other Hematologic Disease '
 cgar_CL24 = 'Claim based cgar24 ETOH Abuse '
 cgar_CL25 = 'Claim based cgar25 Illicit Drug Use '
 cgar_CL26 = 'Claim based cgar26 Tobacco Use '
 cgar_CL27 = 'Claim based cgar27 Dementia '
 cgar_CL28 = 'Claim based cgar28 Depression '
 cgar_CL29 = 'Claim based cgar29 Bipolar Disease '
 cgar_CL30 = 'Claim based cgar30 Schizophrenia '
 cgar_CL31 = 'Claim based cgar31 Anxiety '
 cgar_CL32 = 'Claim based cgar32 Posttraumatic Stress Disorder (PTSD) '
 cgar_CL33 = 'Claim based cgar33 Attention Deficit Hyperactivity Disorder
 ADD-ADHD '
 cgar_CL34 = 'Claim based cgar34 Mental Retardation (HCC term) '
 cgar_CL35 = 'Claim based cgar35 Other Mental Health Disorders '
 cgar_CL36 = 'Claim based cgar36 Otitis Media '
 cgar_CL37 = 'Claim based cgar37 Parkinsons Disease '
 cgar_CL38 = 'Claim based cgar38 Multiple Sclerosis '
 cgar_CL39 = 'Claim based cgar39 Paralysis '
 cgar_CL40 = 'Claim based cgar40 Seizure Disorders '
 cgar_CL41 = 'Claim based cgar41 Headaches '
 cgar_CL42 = 'Claim based cgar42 Migraine '
 cgar_CL43 = 'Claim based cgar43 Cataract '
 cgar_CL44 = 'Claim based cgar44 Glaucoma '
 cgar_CL45 = 'Claim based cgar45 Eye Disorders? '
 cgar_CL46 = 'Claim based cgar46 Vestibular Disorders '
 cgar_CL47 = 'Claim based cgar47 Other Ear '
 cgar_CL48 = 'Claim based cgar48 Other Disease of the Central Nervous
 System? '
 cgar_CL49 = 'Claim based cgar49 Hypertension '
 cgar_CL50 = 'Claim based cgar50 Undiagnosed Hypertension '
 cgar_CL51 = 'Claim based cgar51 Acute myocardial infarction (AMI) '
 cgar_CL52 = 'Claim based cgar52 Coronary Atherosclerosis and other heart
 disease '
 cgar_CL53 = 'Claim based cgar53 Atrial Fibrillation and flutter '
 cgar_CL54 = 'Claim based cgar54 Other Arrhythmias '
 cgar_CL55 = 'Claim based cgar55 Cardiac Arrest (includes VF) '
 cgar_CL56 = 'Claim based cgar56 Congestive Heart Failure '
 cgar_CL57 = 'Claim based cgar57 Acute Hemorrhagic Stroke '
 cgar_CL58 = 'Claim based cgar58 Ischemic Stroke '
 cgar_CL59 = 'Claim based cgar59 Cerebrovascular Disease '
 cgar_CL60 = 'Claim based cgar60 Peripheral Vascular Disease '

cgar_CL61 = 'Claim based cgar61 Other Cardiovascular Diseases '
cgar_CL62 = 'Claim based cgar62 Other Vascular Diseases '
cgar_CL63 = 'Claim based cgar63 Pulmonary Embolism '
cgar_CL64 = 'Claim based cgar64 DVT '
cgar_CL65 = 'Claim based cgar65 Pneumonia (non-TB, non-STD) '
cgar_CL66 = 'Claim based cgar66 Influenza '
cgar_CL67 = 'Claim based cgar67 Chronic Obstructive Pulmonary Disease (aka
Emphysema) '
cgar_CL68 = 'Claim based cgar68 Asthma '
cgar_CL69 = 'Claim based cgar69 Acute respiratory infection '
cgar_CL70 = 'Claim based cgar70 Respiratory symptoms '
cgar_CL71 = 'Claim based cgar71 Other Respiratory Diseases '
cgar_CL72 = 'Claim based cgar72 Reflux/Ulcer Diseases '
cgar_CL73 = 'Claim based cgar73 Biliary Tract Disease '
cgar_CL74 = 'Claim based cgar74 Liver Disease '
cgar_CL75 = 'Claim based cgar75 Gastrointestinal Bleeding '
cgar_CL76 = 'Claim based cgar76 Other Gastrointestinal Disorders '
cgar_CL77 = 'Claim based cgar77 Acute Renal Failure '
cgar_CL78 = 'Claim based cgar78 Chronic Renal Failure '
cgar_CL79 = 'Claim based cgar79 End Stage Renal Disease (ESRD) '
cgar_CL80 = 'Claim based cgar80 Urinary Tract Infection (UTI) '
cgar_CL81 = 'Claim based cgar81 Urinary Incontinence '
cgar_CL82 = 'Claim based cgar82 Hyperplasia of the Prostate '
cgar_CL83 = 'Claim based cgar83 Other Genitourinary Diseases '
cgar_CL84 = 'Claim based cgar84 Pregnancy and Childbirth '
cgar_CL85 = 'Claim based cgar85 Menopause '
cgar_CL86 = 'Claim based cgar86 Contraception and Procreation '
cgar_CL87 = 'Claim based cgar87 Dermatologic Diseases '
cgar_CL88 = 'Claim based cgar88 Rheumatoid Arthritis '
cgar_CL89 = 'Claim based cgar89 Osteoarthritis '
cgar_CL90 = 'Claim based cgar90 Back Pain '
cgar_CL91 = 'Claim based cgar91 Osteoporosis '
cgar_CL92 = 'Claim based cgar92 Other Rheumatism Diseases '
cgar_CL93 = 'Claim based cgar93 Congenital Disorders '
cgar_CL94 = 'Claim based cgar94 Newborn conditions '
cgar_CL95 = 'Claim based cgar95 Trauma '
cgar_CL96 = 'Claim based cgar96 Hip Fractures '
cgar_CL97 = 'Claim based cgar97 Fractures '
cgar_CL98 = 'Claim based cgar98 Poisoning and Other Injury '
cgar_CL100 = 'Claim based cgar100 Motor Vehicle Accident '
cgar_cl102 = 'Screening: Breast Cancer '
cgar_cl103 = 'Screening: Colon Cancer '
cgar_cl104 = 'Screening: Prostate Cancer '
cgar_cl105 = 'Screening: Cervical Cancer '
cgar_cl122='Claim based cgar122 Generic Illness '
cgar_cl123='Claim based cgar123 Well Care '

cgar_cl124='Claim based cgar124 Well Care B'
cgar_cl125='Claim based cgar125 Accident Ecodes';

cgar_ccl1 = 'Calibrated claim based cgar01 Tuberculosis '
cgar_ccl2 = 'Calibrated claim based cgar02 STD non-HIV '
cgar_ccl3 = 'Calibrated claim based cgar03 HIV '
cgar_ccl4 = 'Calibrated claim based cgar04 Immunizations and screening for
infectious disease '
cgar_ccl5 = 'Calibrated claim based cgar05 Other Infectious Disease '
cgar_ccl6 = 'Calibrated claim based cgar06 Colon Cancer '
cgar_ccl7 = 'Calibrated claim based cgar07 Lung Cancer '
cgar_ccl8 = 'Calibrated claim based cgar08 Skin Cancer '
cgar_ccl9 = 'Calibrated claim based cgar09 Breast Cancer '
cgar_ccl10 = 'Calibrated claim based cgar10 Cervical Cancer '
cgar_ccl11 = 'Calibrated claim based cgar11 Prostate Cancer '
cgar_ccl12 = 'Calibrated claim based cgar12 Hematologic Cancer '
cgar_ccl13 = 'Calibrated claim based cgar13 Benign Neoplasm '
cgar_ccl14 = 'Calibrated claim based cgar14 Other Cancer '
cgar_ccl15 = 'Calibrated claim based cgar15 Thyroid disorders '
cgar_ccl16 = 'Calibrated claim based cgar16 Diabetes Mellitus '
cgar_ccl17 = 'Calibrated claim based cgar17 Undiagnosed Diabetes Mellitus '
cgar_ccl18 = 'Calibrated claim based cgar18 Hyperlipidemia '
cgar_ccl19 = 'Calibrated claim based cgar19 Undiagnosed Hyperlipidemia '
cgar_ccl20 = 'Calibrated claim based cgar20 Gout and other crystal
arthropathies '
cgar_ccl21 = 'Calibrated claim based cgar21 Other Endocrine Diseases '
cgar_ccl22 = 'Calibrated claim based cgar22 Anemias '
cgar_ccl23 = 'Calibrated claim based cgar23 Other Hematologic Disease '
cgar_ccl24 = 'Calibrated claim based cgar24 ETOH Abuse '
cgar_ccl25 = 'Calibrated claim based cgar25 Illicit Drug Use '
cgar_ccl26 = 'Calibrated claim based cgar26 Tobacco Use '
cgar_ccl27 = 'Calibrated claim based cgar27 Dementia '
cgar_ccl28 = 'Calibrated claim based cgar28 Depression '
cgar_ccl29 = 'Calibrated claim based cgar29 Bipolar Disease '
cgar_ccl30 = 'Calibrated claim based cgar30 Schizophrenia '
cgar_ccl31 = 'Calibrated claim based cgar31 Anxiety '
cgar_ccl32 = 'Calibrated claim based cgar32 Posttraumatic Stress Disorder
(PTSD) '
cgar_ccl33 = 'Calibrated claim based cgar33 Attention Deficit Hyperactivity
Disorder ADD-ADHD '
cgar_ccl34 = 'Calibrated claim based cgar34 Mental Retardation (HCC term)
,
cgar_ccl35 = 'Calibrated claim based cgar35 Other Mental Health Disorders '
cgar_ccl36 = 'Calibrated claim based cgar36 Otitis Media '
cgar_ccl37 = 'Calibrated claim based cgar37 Parkinsons Disease '
cgar_ccl38 = 'Calibrated claim based cgar38 Multiple Sclerosis '

cgar_cc139 = 'Calibrated claim based cgar39 Paralysis '
cgar_cc140 = 'Calibrated claim based cgar40 Seizure Disorders '
cgar_cc141 = 'Calibrated claim based cgar41 Headaches '
cgar_cc142 = 'Calibrated claim based cgar42 Migraine '
cgar_cc143 = 'Calibrated claim based cgar43 Cataract '
cgar_cc144 = 'Calibrated claim based cgar44 Glaucoma '
cgar_cc145 = 'Calibrated claim based cgar45 Eye Disorders? '
cgar_cc146 = 'Calibrated claim based cgar46 Vestibular Disorders '
cgar_cc147 = 'Calibrated claim based cgar47 Other Ear '
cgar_cc148 = 'Calibrated claim based cgar48 Other Disease of the Central
Nervous System? '
cgar_cc149 = 'Calibrated claim based cgar49 Hypertension '
cgar_cc150 = 'Calibrated claim based cgar50 Undiagnosed Hypertension '
cgar_cc151 = 'Calibrated claim based cgar51 Acute myocardial infarction
(AMI) '
cgar_cc152 = 'Calibrated claim based cgar52 Coronary Atherosclerosis and
other heart disease '
cgar_cc153 = 'Calibrated claim based cgar53 Atrial Fibrillation and flutter
'
cgar_cc154 = 'Calibrated claim based cgar54 Other Arrhythmias '
cgar_cc155 = 'Calibrated claim based cgar55 Cardiac Arrest (includes VF) '
cgar_cc156 = 'Calibrated claim based cgar56 Congestive Heart Failure '
cgar_cc157 = 'Calibrated claim based cgar57 Acute Hemorrhagic Stroke '
cgar_cc158 = 'Calibrated claim based cgar58 Ischemic Stroke '
cgar_cc159 = 'Calibrated claim based cgar59 Cerebrovascular Disease '
cgar_cc160 = 'Calibrated claim based cgar60 Peripheral Vascular Disease '
cgar_cc161 = 'Calibrated claim based cgar61 Other Cardiovascular Diseases '
cgar_cc162 = 'Calibrated claim based cgar62 Other Vascular Diseases '
cgar_cc163 = 'Calibrated claim based cgar63 Pulmonary Embolism '
cgar_cc164 = 'Calibrated claim based cgar64 DVT '
cgar_cc165 = 'Calibrated claim based cgar65 Pneumonia (non-TB, non-STD) '
cgar_cc166 = 'Calibrated claim based cgar66 Influenza '
cgar_cc167 = 'Calibrated claim based cgar67 Chronic Obstructive Pulmonary
Disease (aka Emphysema) '
cgar_cc168 = 'Calibrated claim based cgar68 Asthma '
cgar_cc169 = 'Calibrated claim based cgar69 Acute respiratory infection '
cgar_cc170 = 'Calibrated claim based cgar70 Respiratory symptoms '
cgar_cc171 = 'Calibrated claim based cgar71 Other Respiratory Diseases '
cgar_cc172 = 'Calibrated claim based cgar72 Reflux/Ulcer Diseases '
cgar_cc173 = 'Calibrated claim based cgar73 Biliary Tract Disease '
cgar_cc174 = 'Calibrated claim based cgar74 Liver Disease '
cgar_cc175 = 'Calibrated claim based cgar75 Gastrointestinal Bleeding '
cgar_cc176 = 'Calibrated claim based cgar76 Other Gastrointestinal
Disorders '
cgar_cc177 = 'Calibrated claim based cgar77 Acute Renal Failure '
cgar_cc178 = 'Calibrated claim based cgar78 Chronic Renal Failure '

```

cgar_ccl179 = 'Calibrated claim based cgar79 End Stage Renal Disease (ESRD)
',
cgar_ccl180 = 'Calibrated claim based cgar80 Urinary Tract Infection (UTI) '
cgar_ccl181 = 'Calibrated claim based cgar81 Urinary Incontinence '
cgar_ccl182 = 'Calibrated claim based cgar82 Hyperplasia of the Prostate '
cgar_ccl183 = 'Calibrated claim based cgar83 Other Genitourinary Diseases '
cgar_ccl184 = 'Calibrated claim based cgar84 Pregnancy and Childbirth '
cgar_ccl185 = 'Calibrated claim based cgar85 Menopause '
cgar_ccl186 = 'Calibrated claim based cgar86 Contraception and Procreation '
cgar_ccl187 = 'Calibrated claim based cgar87 Dermatologic Diseases '
cgar_ccl188 = 'Calibrated claim based cgar88 Rheumatoid Arthritis '
cgar_ccl189 = 'Calibrated claim based cgar89 Osteoarthritis '
cgar_ccl190 = 'Calibrated claim based cgar90 Back Pain '
cgar_ccl191 = 'Calibrated claim based cgar91 Osteoporosis '
cgar_ccl192 = 'Calibrated claim based cgar92 Other Rheumatism Diseases '
cgar_ccl193 = 'Calibrated claim based cgar93 Congenital Disorders '
cgar_ccl194 = 'Calibrated claim based cgar94 Newborn conditions '
cgar_ccl195 = 'Calibrated claim based cgar95 Trauma '
cgar_ccl196 = 'Calibrated claim based cgar96 Hip Fractures '
cgar_ccl197 = 'Calibrated claim based cgar97 Fractures '
cgar_ccl198 = 'Calibrated claim based cgar98 Poisoning and Other Injury '
cgar_ccl100 = 'Calibrated claim based cgar100 Motor Vehicle Accident '
cgar_ccl102 = 'Calibrated Screening: Breast Cancer'
cgar_ccl103 = 'Calibrated Screening: Colon Cancer'
cgar_ccl104 = 'Calibrated Screening: Prostate Cancer'
cgar_ccl105 = 'Calibrated Screening: Cervical Cancer'
cgar_ccl122='Calibrated claim based cgar122 Generic Illness'
cgar_ccl123='Calibrated claim based cgar123 Well Care'
cgar_ccl124='Calibrated claim based cgar124 Well Care B'
cgar_ccl125='Calibrated claim based cgar125 Accident Ecodes';

```

```

keep baseid _mult_ psu strat wgt
age asthma_emphysema bc_taken bp_taken cgarsr10 cgarsr11 cgarsr16 cgarsr18
cgarsr28 cgarsr37 cgarsr39 cgarsr43
cgarsr49 cgarsr51 cgarsr52 cgarsr56 cgarsr57 cgarsr6 cgarsr7
cgarsr8 cgarsr82 cgarsr88 cgarsr89 cgarsr9 cgarsr91 cgarsr96 comphealth cost
didserv died dif_lift dif_stoop
dif_walk dwel ed5 eversmoke flushot hasjob havecare healthstat
hearing hearingaid height hyst i_days inpatnights inpatstays male mammogram
maritals nbrpeopl nbrrooms pap_smear
pneushot povcat prb_dres prb_eat priv_insur psa1yr pure race
smokenow weightkg
cgar_cl1-cgar_cl98 cgar_ccl1-cgar_ccl98 cgar_cl100 cgar_ccl100 cgar_cl102-
cgar_cl105 cgar_ccl102-cgar_ccl105
cgar_cl122-cgar_cl125 cgar_ccl122-cgar_ccl125;
Run;

```



```
proc means;  
var cgar_cl102-cgar_cl105 cgar_cc1102-cgar_cc1105;  
Run;  
%mend;  
  
%split(cg=57_58);
```

```
/******  
Program: 'save_anal_mcbs09.sas'  
Purpose: Create a table of prevalence estimates of SR in NHANES  
and SR, calibrated and claim-based prevalences in MCBS community population  
Data in: Insert path for input datasets  
Data out: Insert path for output dataset  
*****/  

```

```
Libname g "Insert file path";  
%include "Insert file path\xdelete.sas";  
%include "Insert file path\formats.sas";  
%xdelete(_ALL_);
```

```
proc printto;  
Run;  
%xdelete(_ALL_);
```

```
%macro strings;  
%global me_mcbs;  
%global me_nh;  
%global mcbs_nh;  
%global nhanes_sr;  
%global meps;  
%global mcbs;  
%global me_mcbs_sr;  
%global me_nh_sr;  
%global mcbs_nh_sr;  
%global cat_me_mcbs_sr;  
%global cat_me_nh_sr;  
%global cat_mcbs_nh_sr;  
%global mcbs_sr;  
%global cat_mcbs_sr;  
%global sr_mcbs;  
%global sr_meps;
```

```
%global n_me_mcbs;  
%global n_me_nh;  
%global n_mcbs_nh;  
%global numcalibr;  
%global nsr_mcbs;  
%global claim_me1;  
%global claim_me2;  
%global nclaim_me1;  
%global nclaim_me2;  
%global claim_me3;  
%global nclaim_me3;
```

```

%global cgar_nhanes;
%global cgar_me_mc;
%global nclaim_me;
%global ncgar_nhanes;
%global ncgar_me_mc;
%global nsr_meps;
%global nsr_me_nh;
%global nsr_mcbs_nh;
%global sr_me_nh;
%global sr_mcbs_nh;

%let me_mcbs=;
%let me_nh=;
%let mcbs_nh=;
%let nhanes_sr=;
%let meps=;
%let mcbs=;
%let me_mcbs_sr=;
%let me_nh_sr=;
%let mcbs_nh_sr=;
%let cat_me_mcbs_sr=;
%let cat_me_nh_sr=;
%let cat_mcbs_nh_sr=;
%let numcalibr=;
%let mcbs_sr=;
%let sr_mcbs=;
%let sr_meps=;

%let nsr_mcbs=;
%let nsr_meps=;

%let cat_mcbs_sr=;
%let claim_me1=;
%let claim_me2=;
%let claim_me3=;

%let cgar_nhanes=;
%let cgar_me_mc=;
%let nsr_me_nh=;
%let nsr_mcbs_nh=;
%let sr_me_nh=;
%let sr_mcbs_nh=;
proc contents data=g.MCBS_ni_i09 (drop = type faccost instcost ) out=mcbs;
Run;

```

```
proc contents data=g.meps65_i09 out=meps;  
Run;
```

```
proc contents data=g.nhanes65_i0910 out=nhanes;  
Run;
```

```
Proc sort data=g.claim_meps09 out=meps_claims05;  
by dustersid;  
Run;
```

```
Data meps_claims05;  
Set meps_claims05;  
cgar57_58=max(cgar57,cgar58);  
drop cgar26 cgar28 cgar29 cgar30 cgar35 cgar53 cgar54 cgar81 cgar83 cgar78  
cgar79 cgar100 cgar101 cgar58 ;  
Run;
```

```
proc contents data=meps_claims05 out=claim_me;  
Run;
```

```
Data claim_me2;  
Set claim_me;  
var=lowcase(name);  
If substr(var,1,4)='cgar';  
if var not in('cgar57_58');  
Run;
```

```
Data claim_me1;  
Set claim_me;  
var=lowcase(name);  
If substr(var,1,4)='cgar';  
if var not in('cgar57_58', /*the following 4 CGARS are year  
specific*/'cgar1','cgar3', 'cgar24', 'cgar84','cgar86','cgar94');  
Run;
```

```
Data claim_me3;  
Set claim_me;  
var=lowcase(name);  
If substr(var,1,4)='cgar';  
Run;
```

```
Data meps;  
Set meps;  
var=lowcase(name);  
Run;
```

```

Data mcbs;
  Set mcbs;
  var=lowcase(name);
Run;

```

```

Data nhanes;
  Set nhanes;
  var=lowcase(name);
Run;

```

```

Proc sort data=meps;
by var;
Proc sort data=mcbs;
by var;
Proc sort data=nhanes;
by var;
Data g.comcov;
Merge nhanes(rename=(memname=nha)) meps(rename=(memname=meps))
mcbs(rename=(memname=mcbs));
by var;
keep var nha mcbs meps;
Run;

```

```

Data me_mcbs me_nh mcbs_nh sr_mcbs nhanes_sr meps mcbs me_mcbs_sr me_nh_sr
mcbs_nh_sr mcbs_sr sr_meps sr_me_nh sr_mcbs_nh;
  Set g.comcov;
  If var not in('_mult_', 'baseid', 'psu', 'strat', 'wgt');
  If meps^='' and mcbs^='' then output me_mcbs;
  If meps^='' and nha^='' then output me_nh;
  If nha^='' and mcbs^='' and var not in('cgarsr82', 'cgarsr89', 'cgarsr8')
then output mcbs_nh;
  If nha^='' and substr(var,1,6)='cgarsr' then output nhanes_sr;
  If mcbs^='' and substr(var,1,6)='cgarsr' then output sr_mcbs;
  If meps^='' and substr(var,1,6)='cgarsr' then output sr_meps;

  If meps^='' then output meps;
  If mcbs^='' then output mcbs;
  If mcbs^='' and substr(var,1,6)^='cgarsr' then output mcbs_sr;
  If meps^='' and mcbs^='' and substr(var,1,6)^='cgarsr' then output
me_mcbs_sr;

  If meps^='' and nha^='' and substr(var,1,6)^='cgarsr' then output
me_nh_sr;
  If nha^='' and mcbs^='' and substr(var,1,6)^='cgarsr'
and var not in('asthma_emphysema') then output mcbs_nh_sr;

```

```
    If meps^='' and nha^='' and substr(var,1,6)='cgarsr' then output sr_me_nh;
    If nha^='' and mcbs^='' and substr(var,1,6)='cgarsr' and var not
in('cgarsr82', 'cgarsr89', 'cgarsr8') then output sr_mcbs_nh;
Run;
```

```
data _null_;
  Set me_mcbs;
  call symput('me_mcbs', trim(resolve('&me_mcbs'))||' '||trim(var));
run;
```

```
data _null_;
  Set me_nh;
  call symput('me_nh', trim(resolve('&me_nh'))||' '||trim(var));
run;
```

```
data _null_;
  Set mcbs_nh;
  call symput('mcbs_nh', trim(resolve('&mcbs_nh'))||' '||trim(var));
run;
```

```
data _null_;
  Set nhanes_sr;
  call symput('nhanes_sr', trim(resolve('&nhanes_sr'))||'
' ||trim(var));
run;
```

```
data _null_;
  Set mcbs;
  call symput('mcbs', trim(resolve('&mcbs'))||' '||trim(var));
run;
```

```
data _null_;
  Set meps;
  call symput('meps', trim(resolve('&meps'))||' '||trim(var));
run;
```

```
data _null_;
  Set me_mcbs_sr;
  call symput('me_mcbs_sr', trim(resolve('&me_mcbs_sr'))||' '||trim(var));
run;
```

```
data _null_;
  Set me_nh_sr;
  call symput('me_nh_sr', trim(resolve('&me_nh_sr'))||' '||trim(var));
run;
```

```

data _null_;
  Set sr_me_nh;
  call symput('sr_me_nh', trim(resolve('&sr_me_nh'))||' '||trim(var));
run;

data _null_;
  Set mcbs_nh_sr;
  call symput('mcbs_nh_sr', trim(resolve('&mcbs_nh_sr'))||'
' ||trim(var));
run;

data _null_;
  Set sr_mcbs_nh;
  call symput('sr_mcbs_nh', trim(resolve('&sr_mcbs_nh'))||'
' ||trim(var));
run;

data _null_;
  Set mcbs_sr;
  call symput('mcbs_sr', trim(resolve('&mcbs_sr'))||' '||trim(var));
run;

data _null_;
  Set sr_mcbs;
  call symput('sr_mcbs', trim(resolve('&sr_mcbs'))||' '||trim(var));
run;

data _null_;
Set sr_meps;
  call symput('sr_meps', trim(resolve('&sr_meps'))||' '||trim(var));
run;

Proc sql;
  Select n(var)into:n_me_mcbs from me_mcbs;
  Quit;

Proc sql;
  Select n(var)into:n_me_nh from me_nh;
  Quit;

Proc sql;
  Select n(var)into:n_mcbs_nh from mcbs_nh;
  Quit;

Proc sql;
  Select n(var)into:numcalibr from nhanes_sr;

```

```

Quit;

Proc sql;
  Select n(var)into:nsr_mcbs from sr_mcbs;
Quit;

Proc sql;
  Select n(var)into:nsr_meps from sr_meps;
Quit;

Proc sql;
  Select n(var)into:nsr_mcbs_nh from sr_mcbs_nh;
Quit;

Proc sql;
  Select n(var)into:nsr_me_nh from sr_me_nh;
Quit;

Proc freq data=g.nhanes65_i0910 nlevels;
table &mcbs_nh_sr;
ods output nlevels=mcbs_nh_sr;
Run;

Data mcbs_nh_sr1;
  Set mcbs_nh_sr;
  catvar=tablevar;
  If nlevels<=4;
Run;

data _null_;
  Set mcbs_nh_sr1;
  call symput('cat_mcbs_nh_sr', trim(resolve('&cat_mcbs_nh_sr'))||'|'
' ||trim(catvar));
run;

Proc freq data=g.nhanes65_i0910 nlevels;
table &me_nh_sr;
ods output nlevels=me_nh_sr;
Run;

Data me_nh_sr1;
  Set me_nh_sr;
  catvar=tablevar;
  If nlevels<=4;
Run;

```



```

data _null_;
  Set me_nh_sr1;
  call symput('cat_me_nh_sr', trim(resolve('&cat_me_nh_sr'))||'
'||trim(catvar));
run;

Proc freq data=g.MCBS_ni_i09 nlevels;
table &me_mcbs_sr;
ods output nlevels=me_mcbs_sr;
Run;

Data me_mcbs_sr1;
  Set me_mcbs_sr;
  catvar=tablevar;
  If nlevels<=4;
  Run;

data _null_;
  Set me_mcbs_sr1;
  call symput('cat_me_mcbs_sr', trim(resolve('&cat_me_mcbs_sr'))||'
'||trim(catvar));
run;

Proc freq data=g.MCBS_ni_i09 nlevels;
table &mcbs_sr;
ods output nlevels=mcbs_sr;
Run;

Data mcbs_sr1;
  Set mcbs_sr;
  catvar=tablevar;
  If nlevels<=4;
  Run;

data _null_;
  Set mcbs_sr1;
  call symput('cat_mcbs_sr', trim(resolve('&cat_mcbs_sr'))||'
'||trim(catvar));
run;

*****Creates a list of diseases to be calibrated to
NHANES*****;
data _null_;
  Set claim_me2;
  call symput('claim_me2', trim(resolve('&claim_me2'))||' '||trim(var));
run;

```

```

Proc sql;
  Select n(var)into:nclaim_me2 from claim_me2;
  Quit;

data _null_;
  Set claim_me3;
  call symput('claim_me3', trim(resolve('&claim_me3'))||' '||trim(var));
run;

Proc sql;
  Select n(var)into:nclaim_me3 from claim_me3;
  Quit;

data _null_;
  Set claim_me1;
  call symput('claim_me1', trim(resolve('&claim_me1'))||' '||trim(var));
run;

Proc sql;
  Select n(var)into:nclaim_me1 from claim_me1;
  Quit;

Data cgar_nhanes;
  Set nhanes_sr;
  var='cgar' || substr(var,7);
run;

data _null_;
  Set cgar_nhanes;
  call symput('cgar_nhanes', trim(resolve('&cgar_nhanes'))||'
' ||trim(var));
run;

Proc sql;
  Select n(var)into:ncgar_nhanes from cgar_nhanes;
  Quit;

*****Creates a list of MEPS diseases to be calibrated to
MCBS*****;
proc sort data=claim_me2;
by var;
proc sort data=cgar_nhanes;
by var;
Data cgar_me_mc;
Merge claim_me2(in=a) cgar_nhanes(in=b);

```

```
by var;  
If a and not b;  
If var^='cgar57_58';  
Run;
```

```
data _null_;  
    Set cgar_me_mc;  
    call symput('cgar_me_mc', trim(resolve('&cgar_me_mc'))||' '||trim(var));  
run;
```

```
Proc sql;  
    Select n(var)into:ncgar_me_mc from cgar_me_mc;  
Quit;
```

```
%put me_mcbs=&me_mcbs;  
%put n_me_mcbs=&n_me_mcbs;
```

```
%put me_nh= &me_nh;  
%put n_me_nh= &n_me_nh;
```

```
%put mcbs_nh=&mcbs_nh;  
%put n_mcbs_nh=&n_mcbs_nh;
```

```
%put nhanes_sr= &numcalibr &nhanes_sr;  
%put mcbs_sr=&mcbs_sr;  
%put cat_mcbs_sr=&cat_mcbs_sr;
```

```
%put meps=&meps;  
%put mcbs=&mcbs;  
%put me_mcbs_sr=&me_mcbs_sr;  
%put me_nh_sr=&me_nh_sr;  
%put mcbs_nh_sr=&mcbs_nh_sr;  
%put cat_me_mcbs_sr=&cat_me_mcbs_sr;  
%put cat_me_nh_sr=&cat_me_nh_sr;  
%put cat_mcbs_nh_sr=&cat_mcbs_nh_sr;  
%put sr_mcbs=&sr_mcbs;  
%put claim_me1=&claim_me1;  
%put claim_me2=&claim_me2;  
%put claim_me3=&claim_me3;
```

```
%put cgar_nhanes=&cgar_nhanes;  
%put ncgar_me_mc=&ncgar_me_mc;
```

```
%put cgar_me_mc=&cgar_me_mc;  
%put sr_meps=&sr_meps;  
%put sr_me_nh=&sr_me_nh;
```

```
%put sr_mcbs_nh=&sr_mcbs_nh;  
%mend;
```

```
%strings;
```

```
%macro answer;
```

```
Proc sort data=g.calibr3_mcbs09 out=all;  
by _mult_;  
Run;
```

```
Data all;  
Set all;  
cgarsr102=mammogram;  
cgarsr105=pap_smear;  
cgarsr104=psa1yr;  
If male=0 then do;  
cgarsr11=. ;  
cgarsr82=. ;  
cgarsr104=.;  
cgar_cl11=. ;  
Cgar_cc111=. ;  
cgar_cl82=. ;  
Cgar_cc182=. ;  
cgar_cl104=. ;  
Cgar_cc104=. ;  
end;  
If male=1 then do;  
cgarsr102=. ;  
cgarsr105=. ;  
cgar_cl9=. ;  
Cgar_cc19=. ;  
cgar_cl10=. ;  
Cgar_cc110=. ;  
cgar_cl102=. ;  
Cgar_cc1102=. ;  
cgar_cl105=. ;  
Cgar_cc1105=. ;  
end;  
Run;
```

```
proc surveyfreq data=all;  
by _mult_;  
table cgar_cc11-cgar_cc198 cgar_cc1100 cgar_cc1102-cgar_cc1105  
cgar_cc1122-cgar_cc1125;  
ods output OneWay=mcbs_ccl1;
```

```

        strata STRat;
        cluster psu;
        weight WGT;
Run;

Data mcbs_ccl2(keep=clcgar _mult_ percent stderr);
  Set mcbs_ccl;;
  length clcgar $7;
  clcgar=substr(table,15);
      %do h=1 %to 98;
        If cgar_ccl&h=1 then output;
      %end;
        If cgar_ccl100=1 then output;
      %do h=102 %to 105;
        If cgar_ccl&h=1 then output;
      %end;
      %do h=122 %to 125;
        If cgar_ccl&h=1 then output;
      %end;
Run;

Proc sql;
  Create table parest as
  select mean(percent) as percent, var(percent) as bvar, mean(stderr**2) as
wvar,clcgar
  from mcbs_ccl2
  group by clcgar ;
Quit;

Data parest1(keep=clcgar CCl_MCBS);
  Set parest;
  mivar= wvar+bvar*6/5;
  mistderr=sqrt(mivar);
  CCl_MCBS=trim(left(put(percent,5.2)))||'
('||trim(left(put(mistderr,4.2)))||')');
  Run;

*****Observed SR in
MCBS*****;
  proc surveyfreq data=all ;
  by _mult_;
  table &sr_mcbs cgarsr102 cgarsr104 cgarsr105;
  ods output OneWay=mcbs_sr;
  strata STRat;
  cluster psu;
  weight WGT;

```

```

Run;
Data mcbs_sr0(keep=clcgar percent stderr _mult_);
  Set mcbs_sr;;
  length clcgar $7;

  clcgar=substr(table,13);
%do n=1 %to &nsr_mcbs;
%let cgsr=%substr(%qscan(&sr_mcbs, &n),7);
  If cgarsr&cgsr=1 then output;
%end;
  If cgarsr102=1 then output;
  If cgarsr104=1 then output;
  If cgarsr105=1 then output;

Run;

Proc sql;
  Create table mcbs_sr1 as
  select mean(percent) as percent, var(percent) as bvar, mean(stderr**2) as
wvar,clcgar
  from mcbs_sr0
  group by clcgar ;
Quit;

Data mcbs_sr2(keep=clcgar SR_MCBS);
  Set mcbs_sr1;
  mivar= wvar+bvar*6/5;
  mistderr=sqrt(mivar);
  SR_MCBS=trim(left(put(percent,5.2)))||'
('||trim(left(put(mistderr,4.2)))||')');
  Run;

*****Observed SR in NHANES*****;
  proc sort data=g.nhanes65_i0910 out=nhanes65_i0910;
  by _mult_;
  Run;
  Data nhanes65_i0910;
    Set nhanes65_i0910;
cgarsr102=mammogram;
cgarsr105=pap_smear;
cgarsr104=psa1yr;
If male=0 then do;
cgarsr11=. ;
end;
If male=1 then do;

```

```

cgarsr10=.;
cgarsr9=.;
end;
Run;
proc surveyfreq data=nhanes65_i0910;
by _mult_;
table &nhanes_sr;
ods output OneWay=nha_sr;
strata STRat;
cluster psu;
weight WGT;
Run;

Data SR_nha0(keep=clcgar percent stderr _mult_);
Set nha_sr;;
length clcgar $7;

clcgar=substr(table,13);
%do n=1 %to &numcalibr;
%let cgsr=%substr(%qscan(&nhanes_sr, &n),7);
If cgarsr&cgsr=1 then output;
%end;
Run;

Proc sql;
Create table sr_nha1 as
select mean(percent) as percent, var(percent) as bvar, mean(stderr**2) as
wvar,clcgar
from sr_nha0
group by clcgar ;
Quit;

Data sr_nha2(keep=clcgar SR_NHANES);
Set sr_nha1;
mivar= wvar+bvar*6/5;
misderr=sqrt(mivar);
SR_NHANES=trim(left(put(percent,5.2)))||'
('||trim(left(put(misderr,4.2)))||')');
Run;

*****Claims rate in MCBS*****;
proc surveyfreq data=all;
where _mult_=1;
table cgar_cl1-cgar_cl98

```

```

                cgar_cl100 cgar_cl102-cgar_cl105 cgar_cl122-cgar_cl125;
ods output OneWay=mcbs_cl;
    strata STRat;
    cluster psu;
    weight WGT;
Run;

```

```

Data obs_mcbs(keep=clcgar CL_MCBS);
    Set mcbs_cl;;
    length clcgar $7;
    clcgar=substr(table,14);
    CL_MCBS=trim(left(put(percent,5.2)))||'
('||trim(left(put(stderr,4.2)))||')');
        %do m=1 %to 98;
            If cgar_cl&m=1 then output;
        %end;
            If cgar_cl100=1 then output;
        %do m=102 %to 105;
            If cgar_cl&m=1 then output;
        %end;
        %do m=122 %to 125;
            If cgar_cl&m=1 then output;
        %end;
Run;

```

```

proc sort data= SR_nha2;
by clcgar;
proc sort data= mcbs_sr2;
by clcgar;
proc sort data=obs_mcbs;
by clcgar;
proc sort data= parest1;
by clcgar;
proc sort data= g.names out=names;
by clcgar;
run;
Data comp;
    Merge names SR_nha2 mcbs_sr2 obs_mcbs parest1;
    by clcgar;
    cgar=clcgar+0;
    drop clcgar;
Run;

```

```

proc sort;
by cgar;

```



```

Run;

ods rtf file='Insert file path\MCBS-NHANES prev09.rtf';
Proc print label noobs;
Title "MCBS09 to NHANES calibration: Comparison of Prevalence rates";
var cgar name SR_MCBS SR_NHANES CL_MCBS CCl_MCBS;
Run;

ods rtf close;

%mend;
%answer;;

Data g.mcbs09_cl1 g.mcbs09_cl2 g.mcbs09_cl3 g.mcbs09_cl4 g.mcbs09_cl5;
Set g.calibr3_mcbs09;
If everSmoke=0 then smokenow=.;
If male=0 then do;
cgarsr11=. ;
cgarsr82=. ;
psa1yr=. ;
cgar_cl11=. ;
Cgar_cc11=. ;
cgar_cl82=. ;
Cgar_cc182=. ;
cgar_cl104=. ;
Cgar_cc104=. ;
end;
If male=1 then do;
mammogram=. ;
pap_smear=. ;
cgar_cl19=. ;
Cgar_cc19=. ;
cgar_cl10=. ;
Cgar_cc10=. ;
cgar_cl102=. ;
Cgar_cc102=. ;
cgar_cl105=. ;
Cgar_cc105=. ;
end;
If _mult_=1 then output g.mcbs09_cl1;
If _mult_=2 then output g.mcbs09_cl2;
If _mult_=3 then output g.mcbs09_cl3;
If _mult_=4 then output g.mcbs09_cl4;
If _mult_=5 then output g.mcbs09_cl5;
Run;

```

```
ods rtf file='Insert file path\MCBS09 contents.rtf';
```

```
proc contents data=g.calibr3_mcbs09;
```

```
Run;
```

```
ods rtf close;
```

```
proc means data=g.mcbs09_c11;
```

```
Run;
```

mc_nh09.set

```
datain back;
  dataout back2;
  default categorical;
  mixed lcost;
  continuous height weightkg psnhanes ;
  transfer baseid _mult_ imp wgt psu strat mcgar1-mcgar125 cost
  _type_ _freq_ donor cgarsr10 cgarsr11 cgarsr12 cgarsr16 cgarsr17
  cgarsr18 cgarsr19 cgarsr28 cgarsr49 cgarsr50 cgarsr51 cgarsr52
  cgarsr56 cgarsr57 cgarsr6 cgarsr67 cgarsr68 cgarsr7 cgarsr8 cgarsr88
  cgarsr89 cgarsr9 cgarsr91 cgarsr96 age_cont inpatnights mammogram
  flushot cgarsr37 cgarsr39 pap_smear cgar58 i_days ;
  restrict cgar84(male=0) cgar85(male=0) cgar102(male=0)
  cgar104(male=1) cgar105(male=0);
  iterations 2;
  mcnrsqd 0.25;
  multiples 1;
seed 2015 ;
Run;
```

mc_mc09.set

```
title Multiple imputation;
  datain ps;
  dataout f;
  default categorical;
continuous claim p1;
bounds cal claim(>=claim);
interact p1*claim;
transfer _mult_ strat psu wgt baseid condition rank1 cost mark;
  iterations 10;
  multiples 1;
  seed 2015;
run;
```