Documentation for SHHS Analytic Database

Contents

Contents

Documentation for SHHS Analytic Database

1.Introduction

The Sleep Heart Health Study (SHHS) is a prospective cohort study of sleep disordered breathing and risk of cardiovascular diseases. The SHHS cohort was enrolled in the mid-1990's from existing epidemiological studies. Six thousand four hundred forty one participants were studied at 11 different investigative sites across the country; the data from these participants are referred to as SHHS 1. Followup 1 data was collected on cardiovascular events for this original cohort approximately 2-3 years after the enrollment visit. The population for the second follow-up examination (SHHS 2) includes surviving members of the original cohort who when contacted by a letter announcing the continuation of the study, indicated interest in undergoing a second home visit AND/OR PSG.

Reference study documents: http://jhuccs1.us/shhs/index.htm

2. SHHS Data Dictionary

All SHHS 1, SHHS 1 ECG Followup, and SHHS 2, SHHS 2 ECG, SHHS 2 PSG, CVD Outcomes Status, and CVD Outcomes Event variables, variable descriptions and table names, can be found in the document entitled "Analytic Data Dictionary" *(Analytic_Data_Dictionary_ddmonyyyy.xls)*.

The dictionary is composed of sheets enumerating dataset specific variables (SHHS 1, SHHS 1 ECG, Followup 1, SHHS 2, SHHS 2 ECG, SHHS 2 PSG, CVD Outcomes – Status, CVD Outcomes – Events). These sheets can be sorted by "variable name" and by "data table" where the variable resides.

The Data Dictionary should be used in conjuction with the Documentation for SHHS Analytic Database to link variable names with their descriptions and codes.

Documentation for SHHS Analytic Database

3. SHHS 1

3.1. Charactistics of SHHS 1 database

Name of dataset: *SHHS1final__ddmonyyyy__6441.sas7bdat*

Number of Records: 6441 participants

Primary Key Field: PPTID - Participant Identifier (alpha - numeric)

File Formats: All files are SAS 9.1 Files.

A-Variables: These are the key risk factors for cardiovascular disease and outcomes collected by Parent Cohorts and listed in the protocol. See attached Appendix 9.1 entitled - SHHS A-variables

Demographic variables: Gender, Race, DOB as submitted by parent cohorts.

Convention

- Most SAS variable names will have the following two digit identifier appended to the variable name which identifies the form the variable is associated with.

  *Varname*AE - Adverse Event Form
  Varname 20 - Blood Pressure Form
  *Varname*03 - Contact Form / Enrolled Contact Form
  *Varname*15 - Health Interview Form (New York (HIN), Framingham (HIF) & ARIC/CHS/Tuscon/Strong Heart (HIA))
  *Varname*10 - Morning Survey Form
  *Varname*1 - Categorized Medications
  *Varname*25 - Quality of Life (Note: No QOL forms exist for 61 = South Dakota, 62 = Oklahoma & 63 = Phoenix)
  *Varname*50 - Sleep Data Retrieval
  *Varname*02 - Sleep Habits / Enrolled SHQ

3.2. Descriptions of Data

- Descriptions of data:
  - Adverse Events (AE)
    - n = 3468
  - A variables 0997_all / enrolled A-variables (AVARS)
    - n=6441
  - Blood Pressure (BP)
    - n=6416
  - Contact Form /Enrolled Contact Form (CF)
    - n=6417
  - Health Interview (HI)
    - n=6380
    - See Appendix 9.2 for characterization of SHHS 1 and SHHS 2 variables for tobacco, caffeine, and alcohol
  - PSG /Derived PSG
    - n=6441
    - Dataset as received by Reading Center
  - Morning Survey (MS)
    - n= 6308
    - See Appendix 9.2 for characterization of SHHS 1 and SHHS 2 variables for tobacco, caffeine, and alcohol
  - Sleep Habits (SH)
    - n= 6149
  - Quality All / Quality Enrolled (QA)
    - n= 6441
  - Sleep Study QC / Enrolled SHQ (QC)
    - n= 6420

- Tuscon CVD (TCvd)
  - n=909
  - Variable names are *MI* and *STROKE*

- Derived variables
  - These were created and embedded within the datasets:
    - Age_s1
    - BMI_s1 - Body Mass Index
    - ESS_s1 - Epworth Sleepiness Score
    - SF36/SF36Sum - Medical Outcome Short Form (36) Health Survey
    - HTNDerv_s1 - Derived Hypertension Status
    - SmokStat_s1 - Smoking Status

- Indicator variables (see Appendix 9.4)
  - Specific variables were created to indicate for each unique PPTID the presence of SHHS 1 forms in the concatenated database (i.e. 0=No 1=Yes)
  - Indicator variables = "1" if the form is available for the participant and = "0" if the form is not available

3.3. Dataset processing specifics

3.3.1. Demographic variables

DEMOGRAPHIC VARIABLES: Race, DOB, Gender, Site

The following demographic variables were collected as reported by the individual parent cohorts:

DOB = Date of Birth
Gender = (M=Male; F=Female)
Race = Race (1=White, 2=Black, 3=American Indian/Alaskan, 4=Pacific Islander, 5=Hispanic/Mexican American)
Site = Field center/site ID (11=Framingham, 21=Hagerstown, 22=Hagerstown, 31=Minneapolis, 40=New York, 41=New York, 51=Pittsburgh, 52=Sacramento, 61=South Dakota, 62=Oklahoma, 63=Phoenix, 64=Tucson)

3.3.2. Derived non-PSG SHHS variables

CALCULATE age: (age_S1)

```
psgdate = DATEPART(StdyDtqa);   /* Make sure StdyDtqa is present (include in psgvars) */
birthdate = DATEPART(DOB);
age = (psgdate - birthdate) / 365.25;
age = int(age);

drop  psgdate birthdate StdyDtqa;
```

CALCULATE BMI (BMI-S1:

A baseline BMI has been derived for the SHHS 1 dataset. NOTE: There were 5 instances where weight collected as part of the SHHS 1 data collection were not available. Weight from the A-variable data was used instead.

```
/* Create BMI variable */

DATA bmidata;
      SET newshhs1data;
      KEEP pptid weight weight20 height;
RUN;

DATA bmi_s1;
SET bmidata;

**   Calculate BMI:

      IF weight20~=. then bmiweight=weight20;  /* weight from BP, or avar if not in BP */
      IF weight20=. and weight~=. THEN bmiweight=weight;
      BMI = bmiweight /((height/100)*(height/100));  /* height from avar only*/

      IF pptid="45503" THEN BMI=30.97;   /* weight20 is 2.2 times avar weight.  weight20 in lbs
            not kg?  Use avar weight */
      IF pptid="4070135" THEN BMI=26.94;  /* weight20 is 2.2 times avar weight. weight20 in lbs
            not kg?  Use avar weight */
      IF pptid="M142181" THEN BMI=18.98;   /* weight20 improbably low! Use avar weight */
```

```
    IF pptid="302214" THEN BMI=30.44;   /* avar weight is 2.2 times weight20, which is quite
          low.  Use avar weight. */
    IF pptid="6500595" THEN BMI=30.44;   /* avar weight is 2.2 times weight20, which is quite
          low.  Use avar weight. */

DROP  weight20 bmiweight;
RUN;
```

CALCULATE ESS: (ESS_S1)

```
incar=incar02-1;     /* transform from 1-4 scale to 0-3 scale */
lydwn=lydwn02-1;
pgrcar=pgrcar02-1;
sitlch=sitlch02-1;
sitpub=sitpub02-1;
sitrd=sitrd02-1;
sittlk=sittlk02-1;
watv=watv02-1;

if incar>3 then incar=.;  /* remove outliers */
if lydwn>3 then lydwn=.;
if pgrcar>3 then pgrcar=.;
if sitlch>3 then sitlch=.;
if sitpub>3 then sitpub=.;
if sitrd>3 then sitrd=.;
if sittlk>3 then sittlk=.;
if watv>3 then watv=.;

if InCar=. or LyDwn=. or PgrCar=. or SitLch=. or SitPub=. or
SitRd=. or SitTlk=. or WaTV=. then do ESS=.; end;
else ESS=SUM(InCar,LyDwn,PgrCar,SitLch,SitPub,SitRd,SitTlk,WaTV);

DROP/* InCar02 LyDwn02 PgrCar02 SitLch02 SitPub02 SitRd02 SitTlk02 WaTV02 */
incar lydwn pgrcar sitlch sitpub sitrd sittlk watv;
```

SF36/SF36Sum:

The following variables for SF36 scales as well as SF36Sum scores were included:

PF - PHYSICAL FUNCTIONING
BP - BODILY PAIN
GH - GENERAL HEALTH
VT - VITALITY
SF - SOCIAL FUNCTIONING
RE - ROLE EMOTIONAL
MH - MENTAL HEALTH INDEX
MCS - STANDARDIZED MENTAL COMPONENT
PCS - STANDARDIZED PHYSICAL COMPONENT

NOTE: No Quality of Life Variables (SF36 composite scores) exists for the following clinics:

61 = South Dakota
62 = Oklahoma
63 = Phoenix

CALCULATE SMOKING STATUS: SMOKSTAT_S1
(0 = never, 1 = current and 2 = former)

```
/* Create SMOKSTAT variable where 0=Never, 1=Current, 2=Former */

DATA smoke_status;
    SET newshhs1data;
    IF evsmok15=0 THEN smokstat=0;
    IF evsmok15=1 AND smknow15=1 THEN smokstat=1;
    IF evsmok15=1 AND smknow15=0 THEN smokstat=0;
    KEEP pptid smokstat;
RUN;
```

CALCULATE HYPERTENSION STATUS: HTNderiv_S1

Utilizing the average of the 2nd and 3rd seated Blood Pressure Readings from the SHHS 1Blood Pressure Weight (BP1) form and the HTNmeds variable from SHHS 1 Categorized Medication database. *(Nieto et al: Association of Sleep Disordered Breathing Sleep Apnea, and Hypertension in a Large Community-Based Study. JAMA, April 12, 2000. Vol 183. No. 14. P 1829-1836)*

SBP2 + SBP3/2 > 140/90 mm HG OR HTNMeds = 1 will be affirmative for hypertension.

```
/* Create HTNDerv variable where 0=No and 1=Yes */

DATA htndata;
      SET newshhs1data;
      KEEP pptid SYST220 DIAS220 SYST320 DIAS320 HTNMED;
RUN;

DATA htn;
      SET htndata;

      /* Calculate average of 2nd and 3rd seated blood pressure readings */
      BY pptid;
      /*IF first.pptid; *duplicates in this form;*/
      LENGTH name $7.;
      name=TRIM(LEFT(UPCASE(pptid)));
      ARRAY bpori{4} $ SYST220 SYST320 DIAS220 DIAS320;
      ARRAY bpnew{4} bps2 bps3 bpd2 bpd3;
      DO i=1 TO 4;
       IF bpori{i} not in ("n", "?", " ", "m") THEN bpnew{i}=bpori{i}+0;
      ELSE bpnew{i}=.;
      END;

      ARRAY bp{4} bps2-bps3 bpd2-bpd3;
      DO i=1 TO 4;
       IF bp{i}=0 THEN bp{i}=.;
      ELSE bp{i}=bp{i};
      END;

bps=((bps2+bps3)/2);
bpd=((bpd2+bpd3)/2);

DROP SYST220 DIAS220 SYST320 DIAS320 i ii bps2 bps3 bpd2 bpd3 name;
```

3.3.2. Derived non-PSG SHHS variables

```
IF ((bps GT 140 AND bpd GT 90) OR HTNMED=1) THEN HTNDerv=1;
ELSE HTNDerv=0;

DROP HTN15 HTNMED bps bpd;
RUN;
```

If further clarification is needed, please contact the SHHS Coordinating Center via email at shhs@jhsph.edu

4. SHHS 1 ECG

4.1. Characteristics of SHHS 1 ECG database

Name of dataset:	*SHHS1ECG_ddmonyyyy.sas7bdat*
Number of Records:	4889 participants
Primary Key Field:	PPTID - Participant Identifier (alpha - numeric)
File Formats:	All files are SAS 9.1 Files.
Demographic variables:	None

4.2. Description of data

Baseline ECGs were performed by the parent cohorts prior to the PSG. All Field Sites performed a standard resting 12-lead ECG with the participant supine. Ten seconds of data were acquired simultaneously from each lead (I, II, III, aVR, aVL, aVF, V1-V6) to a Marquette MAC PC or MAC II system. At each site, a paper copy of the ECG is produced and filed. Minnesota coding of the ECG data is performed on all participants in ARIC, CHS, and SHS. Minnesota coding may be performed for other sites as well. Currently, NYCC ECG data were interpreted by a clinical cardiologist, and FHS ECG data were interpreted by the clinic physician. Minnesota codes provided by the Field Sites were then used to derive the 34 variables in the ECG dataset.

4.3. Dataset processing specifics

4.3.1. ARIC

```
DATA temp;
    SET aric.ecg (RENAME = (psgdate = datepsg));
    site = 'aric';
    LENGTH pptid $ 12;
    pptid = id;
    psgdate = INPUT(datepsg, DATE7.);
    dateb = INPUT(ecgmb06, DATE77.);
    datec = INPUT(ecgmc06, DATE7.);
    dated = INPUT(ecgmd06, DATE7.);
    FORMAT psgdate ecgdate dateb datec dated DATE7.;

    /*****************************************************************
    Find patients with ECG dates within 3 years before date of PSG
    Create variables
    *****************************************************************/
    IF datec - psgdate GE -(365 * 3) THEN DO;
        ecgdate = datec;
        IF ecgmc22 NE . THEN lvh3_1 = (ecgmc22 IN (11 12 14));
        IF ecgmc22 NE . THEN lvh3_3 = (ecgmc22 IN (31 32));

        IF ecgmc12 NE . OR ecgmc13 NE . OR ecgmc14 NE . THEN DO;
            st4_1_3 = (ecgmc12 IN (3 4 11 12) OR ecgmc13 IN (3 4 11 12) OR ecgmc14 IN (3 4 11 12));
            IF st4_1_3 = 0 AND (ecgmc12 = . OR ecgmc13 = . OR ecgmc14 = .) THEN st4_1_3 = .;
        END;

        IF ecgmc15 NE . OR ecgmc16 NE . OR ecgmc17 NE . THEN DO;
            st5_1_3 = (ecgmc15 IN (1 2 3) OR ecgmc16 IN (1 2 3)  OR ecgmc17 IN (1 2 3));
            IF st5_1_3 = 0 AND (ecgmc15 = . OR ecgmc16 = . OR ecgmc17 = .) THEN st5_1_3 = .;
        END;

        IF ecgmc22 NE . OR st4_1_3 NE . OR st5_1_3 NE . THEN DO;
            lvhst = (ecgmc22 IN (11 12 14 31 32) AND (st4_1_3 = 1 OR st5_1_3 = 1));
            IF lvhst = 0 AND (ecgmc22 = . OR st4_1_3 = . OR st5_1_3 = .)  THEN lvhst = .;
        END;

        IF ecgmc23 NE . THEN mob1 = (ecgmc23 = 2);
        IF ecgmc23 NE . THEN part2deg = (ecgmc23 = 2);
        IF ecgmc23 NE . THEN mob2 = (ecgmc23 = 2);
        IF ecgmc23 NE . THEN av3deg = (ecgmc23 = 11);

        IF ecgmc23 NE . THEN av1deg = (ecgmc23 = 3);
        IF ecgmc24 NE . THEN lbbb = (ecgmc24 = 1);
        IF ecgmc24 NE . THEN rbbb = (ecgmc24 = 2);
        IF ecgmc24 NE . THEN ilbbb = (ecgmc24 = 6);
```

```
IF ecgmc24 NE . THEN irbbb = (ecgmc24 = 3);
IF ecgmc24 NE . THEN lah = (ecgmc24 = 7);
IF ecgmc24 NE . THEN iventblk = (ecgmc24 = 4);
IF ecgmc23 NE . THEN wpw = (ecgmc23 = 4);

IF ecgmc11 NE . THEN DO;
    IF ecgmc11 IN (11 12 16 17 21 22 27) THEN antsepmi = 1;
ELSE IF ecgmc11 IN (28 31 32) THEN antsepmi = 2;
ELSE antsepmi = 0;
END;

IF ecgmc10 NE . THEN DO;
    IF ecgmc10 IN (11 12 14 15 21 22 23 24 25) THEN infmi = 1;
ELSE IF ecgmc10 in (26 31 34 35 36) THEN infmi = 2;
ELSE infmi = 0;
END;

IF ecgmc09 NE . THEN DO;
    IF ecgmc09 IN (11 12 13 21 22 23 24 25 2 27 28) THEN antlatmi = 1;
ELSE IF ecgmc09 IN (31 33) THEN antlatmi = 2;
ELSE antlatmi = 0;
END;

IF ecgmc12 NE . OR ecgmc13 NE . OR ecgmc14 NE . THEN DO;
    IF ecgmc12 IN (2 11 12) OR ecgmc13 IN (2 11 12) OR ecgmc14 IN (2 11 12) THEN nonsp_st =
    1;
ELSE IF ecgmc12 = 3 OR ecgmc13 = 3 OR ecgmc14 = 3 THEN nonsp_st = 2;
ELSE nonsp_st = 0;
    IF nonsp_st = 0 AND (ecgmc12 = . OR ecgmc13 = . OR ecgmc14 = .) THEN nonsp_st = .;
END;

IF ecgmc15 NE . OR ecgmc16 NE . OR ecgmc17 NE . THEN DO;
    IF ecgmc15 IN (1 2) OR ecgmc16 IN (1 2) OR ecgmc17 IN (1 2) THEN nonsp_tw = 1;
ELSE IF ecgmc15 = 3 OR ecgmc16 = 3 OR ecgmc17 = 3 THEN nonsp_tw = 2;
ELSE nonsp_tw = 0;
    IF nonsp_tw = 0 AND (ecgmc15 = . OR ecgmc16 = . OR ecgmc17 = .) THEN nonsp_tw = .;
END;

    IF ecgmc26 NE . THEN rtrial = (ecgmc26 = 3);
    IF ecgmc22 NE . THEN rvh = (ecgmc22 = 2);
END;
```

```
/*****************************************************************;
IF NO ECG DATE IN 3 YEARS BEFORE DATE OF PSG THEN
Find patients with ECG dates within 3 months after date of PSG
Create variables
*****************************************************************/;
ELSE IF dated-psgdate le 91.2 THEN DO;
    ecgdate = dated;
    IF ecgmd22 NE . THEN lvh3_1 = (ecgmd22 IN (11 12 14));
    IF ecgmd22 NE . THEN lvh3_3 = (ecgmd22 IN (31 32));
    IF ecgmd12 NE . OR ecgmd13 NE . OR ecgmd14 NE . THEN DO;
        st4_1_3 = (ecgmd12 IN (3 4 11 12) OR ecgmd13 IN (3 4 11 12) OR ecgmd14 IN (3 4 11 12));
        IF st4_1_3 = 0 AND (ecgmd12 = . OR ecgmd13 = . OR ecgmd14 = .) THEN st4_1_3 = .;
    END;

    IF ecgmd15 NE . OR ecgmd16 NE . OR ecgmd17 NE . THEN DO;
        st5_1_3 = (ecgmd15 IN (1 2 3) OR ecgmd16 IN (1 2 3) OR ecgmd17 IN (1 2 3));
        IF st5_1_3 = 0 AND (ecgmd15 = . OR ecgmd16 = . OR ecgmd17 = .) THEN st5_1_3 = .;
    END;

    IF ecgmd22 NE . OR st4_1_3 NE . OR st5_1_3 NE . THEN DO;
        lvhst = (ecgmd22 IN ( 11 12 14 31 32) AND (st4_1_3 = 1 OR st5_1_3 = 1));
        IF lvhst = 0 AND (ecgmd22 = . OR st4_1_3 = . OR st5_1_3 = .) THEN lvhst = .;
    END;

    IF ecgmd23 NE . THEN mob1 = (ecgmd23 = 2);
    IF ecgmd23 NE . THEN part2deg = (ecgmd23 = 2);
    IF ecgmd23 NE . THEN mob2 = (ecgmd23 = 2);
    IF ecgmd23 NE . THEN av3deg = (ecgmd23 = 11);
    IF ecgmd23 NE . THEN av1deg = (ecgmd23 = 3);
    IF ecgmd24 NE . THEN lbbb = (ecgmd24 = 1);
    IF ecgmd24 NE . THEN rbbb = (ecgmd24 = 2);
    IF ecgmd24 NE . THEN ilbbb = (ecgmd24 = 6);
    IF ecgmd24 NE . THEN irbbb = (ecgmd24 = 3);
    IF ecgmd24 NE . THEN lah = (ecgmd24 = 7);
    IF ecgmd24 NE . THEN iventblk = (ecgmd24 = 4);
    IF ecgmd23 NE . THEN wpw = (ecgmd23 = 4);

    IF ecgmd11 NE . THEN DO;
        IF ecgmd11 IN (11 12 16 17 21 22 27) THEN antsepmi = 1;
    ELSE IF ecgmd11 IN (28 31 32) THEN antsepmi = 2;
    ELSE antsepmi = 0;
    END;
```

```
        IF ecgmd10 NE . THEN DO;
            IF ecgmd10 IN (11 12 14 15 21 22 23 24 25) THEN infmi = 1;
        ELSE IF ecgmd10 IN (26 31 34 35 36) THEN infmi = 2;
        ELSE infmi = 0;
        END;

        IF ecgmd09 NE . THEN DO;
            IF ecgmd09 IN (11 12 13 21 22 23 24 25 2 27 28) THEN antlatmi = 1;
        ELSE IF ecgmd09 IN (31 33) THEN antlatmi = 2;
        ELSE antlatmi = 0;
        END;

        IF ecgmd12 NE . OR ecgmd13 NE . OR ecgmd14 NE . THEN DO;
            IF ecgmd12 IN (2 11 12) OR ecgmd13 IN (2 11 12) OR ecgmd14 IN (2 11 12) THEN nonsp_st =
            1;
        ELSE IF ecgmd12 = 3 OR ecgmd13 = 3 OR ecgmd14 = 3 THEN nonsp_st = 2;
        ELSE nonsp_st = 0;
            IF nonsp_st = 0 and (ecgmd12 = . OR ecgmd13 = . OR ecgmd14 = .)  THEN nonsp_st = .;
        END;

        IF ecgmd15 NE . OR ecgmd16 NE . OR ecgmd17 NE . THEN DO;
            IF ecgmd15 IN (1 2) OR ecgmd16 IN (1 2) OR ecgmd17 IN (1 2) THEN nonsp_tw = 1;
        ELSE IF ecgmd15 = 3 OR ecgmd16 = 3 OR ecgmd17 = 3 THEN nonsp_tw = 2;
        ELSE nonsp_tw = 0;
            IF nonsp_tw = 0 and (ecgmd15 = . OR ecgmd16 = . OR ecgmd17 = .) THEN nonsp_tw = .;
        END;

        IF ecgmd26 NE . THEN rtrial = (ecgmd26 = 3);
        IF ecgmd22 NE . THEN rvh = (ecgmd22 = 2);
    END;
RUN;
```

4.3.2. CHS

```
DATA temp;
    SET chs.ecg;
    site = 'chs';
    LENGTH pptid $ 12;
    pptid = PUT (idno,best12.);
    FORMAT psgdate ecgdate DATE7.;

    /*****************************************************************
    Find patients with ECG dates within 3 years before date of PSG
    Create variables
    *****************************************************************/
    IF rdate1c-psgdatec ge -(365*3) THEN DO;
        psgdate = psgdatec;
        ecgdate = rdate1c;
        IF c342_1 NE . THEN lvh3_1 = (c342_1 IN (2 3 4 5));
        IF c342_1 NE . THEN lvh3_3 = (c342_1 IN (7 8));
        IF l442_1 NE . OR f442_1 NE . OR v442_1 NE . THEN DO;
            st4_1_3 = (l442_1 IN (2 3 4 5) OR f442_1 IN (2 3 4 5) OR v442_1 IN (2 3 4 5));
            IF st4_1_3 = 0 and (l442_1 = . OR f442_1 = . OR v442_1 = .) THEN st4_1_3 = .;
        END;

        IF l542_1 NE . OR f542_1 NE . OR v542_1 NE . THEN DO;
            st5_1_3 = (l542_1 IN (2 3 4 5) OR f542_1 IN (2 3 4 5) OR v542_1 IN (2 3 4 5));
            IF st5_1_3 = 0 and (l542_1 = . OR f542_1 = . OR v542_1 = .) THEN st5_1_3 = .;
        END;

        IF ecglvh1 NE . THEN lvhst = (ecglvh1 = 1);
        ventrate = hrate1;
        qrs = qrsaxi1;
        IF c642_1 NE . THEN mob1 = (c642_1 = 5);
        IF c642_1 NE . THEN part2deg = (c642_1 = 4);
        IF c642_1 NE . THEN mob2 = (c642_1 = 3);
        IF c642_1 NE . THEN av3deg = (c642_1 = 2);
        IF ecgafib1 NE . THEN afib = (ecgafib1 = 1);
        IF c642_1 NE . THEN paced = (c642_1 = 10);
        IF c642_1 NE . THEN av1deg = (c642_1 = 6);
        IF c742_1 NE . THEN lbbb = (c742_1 = 2);
        IF c742_1 NE . THEN rbbb = (c742_1 = 3);
        IF c742_1 NE . THEN ilbbb = (c742_1 = 7);
        IF c742_1 NE . THEN irbbb = (c742_1 = 4);
        IF c742_1 NE . THEN lah = (c742_1 = 8);
        IF c742_1 NE . THEN iventblk = (c742_1 = 5);
        IF c642_1 NE . THEN wpw = (c642_1 = 7);
```

```
IF v142_1 NE . THEN DO;
      IF v142_1 IN (2 3 4 9 10 11) THEN antsepmi = 1;
ELSE IF v142_1 IN (16 17 19) THEN antsepmi = 2;
ELSE antsepmi = 0;
END;

IF f142_1 NE . THEN DO;
      IF f142_1 IN (2 3 5 6 9 10 11 12 13) THEN infmi = 1;
ELSE IF f142_1 IN (14 17 20 21 22) THEN infmi = 2;
ELSE infmi = 0;
END;

IF l142_1 NE . THEN DO;
      IF l142_1 IN (2 3 7 8 9 10 15 16) THEN antlatmi = 1;
ELSE IF l142_1 IN (17 18) THEN antlatmi = 2;
ELSE antlatmi = 0;
END;

IF l442_1 NE . OR f442_1 NE . OR v442_1 NE . THEN DO;
      IF l442_1 IN (2 3 4) OR f442_1 IN (2 3 4) OR v442_1 IN (2 3 4) THEN nonsp_st = 1;
ELSE IF l442_1 = 5 OR f442_1 = 5 OR v442_1 = 5 THEN nonsp_st = 2;
ELSE nonsp_st = 0;
      IF nonsp_st = 0 and (l442_1 = . OR f442_1 = . OR v442_1 = .) THEN nonsp_st = .;
END;

IF l542_1 NE . OR f542_1 NE . OR v542_1 NE . THEN DO;
      IF l542_1 IN (2 3) OR f542_1 IN (2 3) OR v542_1 IN (2 3) THEN nonsp_tw = 1;
ELSE IF l542_1 = 4 OR f542_1 = 4 OR v542_1 = 4 THEN nonsp_tw = 2;
ELSE nonsp_tw = 0;
      IF nonsp_tw = 0 and (l542_1 = . OR f542_1 = . OR v542_1 = .) THEN nonsp_tw = .;
END;

IF c9342_1 NE . THEN rtrial = (c9342_1 = 2);
IF c342_1 NE . THEN rvh = (c342_1 = 6);
END;

/*****************************************************************
IF NO ECG DATE IN 3 YEARS BEFORE DATE OF PSG THEN
Find patients with ECG dates within 3 months after date of PSG
Create variables
*****************************************************************/
ELSE IF rdate2c-psgdatec le 91.2 THEN DO;
      psgdate = psgdatec;
      ecgdate = rdate2c;
```

```
IF c342_2 NE . THEN lvh3_1 = (c342_2 IN (2 3 4 5));
IF c342_2 NE . THEN lvh3_3 = (c342_2 IN (7 8));
IF l442_2 NE . OR f442_2 NE . OR v442_2 NE . THEN DO;
      st4_1_3 = (l442_2 IN (2 3 4 5) OR f442_2 IN (2 3 4 5) OR v442_2 IN (2 3 4 5));
      IF st4_1_3 = 0 and (l442_2 = . OR f442_2 = . OR v442_2 = .) THEN st4_1_3 = .;
END;

IF l542_2 NE . OR f542_2 NE . OR v542_2 NE . THEN DO;
      st5_1_3 = (l542_2 IN (2 3 4 5) OR f542_2 IN (2 3 4 5) OR v542_2 IN (2 3 4 5));
      IF st5_1_3 = 0 and (l542_2 = . OR f542_2 = . OR v542_2 = .) THEN st5_1_3 = .;
END;

IF ecglvh2 NE . THEN lvhst = (ecglvh2 = 1);
ventrate = hrate2;
qrs = qrsaxi2;
IF c642_2 NE . THEN mob1 = (c642_2 = 5);
IF c642_2 NE . THEN part2deg = (c642_2 = 4);
IF c642_2 NE . THEN mob2 = (c642_2 = 3);
IF c642_2 NE . THEN av3deg = (c642_2 = 2);
IF ecgafib2 NE . THEN afib = (ecgafib2 = 1);
IF c642_2 NE . THEN paced = (c642_2 = 10);
IF c642_2 NE . THEN av1deg = (c642_2 = 6);
IF c742_2 NE . THEN lbbb = (c742_2 = 2);
IF c742_2 NE . THEN rbbb = (c742_2 = 3);
IF c742_2 NE . THEN ilbbb = (c742_2 = 7);
IF c742_2 NE . THEN irbbb = (c742_2 = 4);
IF c742_2 NE . THEN lah = (c742_2 = 8);
IF c742_2 NE . THEN iventblk = (c742_2 = 5);
IF c642_2 NE . THEN wpw = (c642_1 = 7);

IF v142_2 NE . THEN DO;
      IF v142_2 IN (2 3 4 9 10 11) THEN antsepmi = 1;
ELSE IF v142_2 IN (16 17 19) THEN antsepmi = 2;
ELSE antsepmi = 0;
END;

IF f142_2 NE . THEN DO;
      IF f142_2 IN (2 3 5 6 9 10 11 12 13) THEN infmi = 1;
ELSE IF f142_2 IN (14 17 20 21 22) THEN infmi = 2;
ELSE infmi = 0;
END;

IF l142_2 NE . THEN DO;
      IF l142_2 IN (2 3 7 8 9 10 15 16) THEN antlatmi = 1;
```

```
            ELSE IF l142_2 IN (17 18) THEN antlatmi = 2;
            ELSE antlatmi = 0;
            END;

            IF l442_2 NE . OR f442_2 NE . OR v442_2 NE . THEN DO;
                IF l442_2 IN (2 3 4) OR f442_2 IN (2 3 4) OR v442_2 IN (2 3 4) THEN nonsp_st = 1;
            ELSE IF l442_2 = 5 OR f442_2 = 5 OR v442_2 = 5 THEN nonsp_st = 2;
            ELSE nonsp_st = 0;
                IF nonsp_st = 0 and (l442_2 = . OR f442_2 = . OR v442_2 = .) THEN nonsp_st = .;
            END;

            IF l542_2 NE . OR f542_2 NE . OR v542_2 NE . THEN DO;
                IF l542_2 IN (2 3) OR f542_2 IN (2 3) OR v542_2 IN (2 3) THEN nonsp_tw = 1;
            ELSE IF l542_2 = 4 OR f542_2 = 4 OR v542_2 = 4 THEN nonsp_tw = 2;
            ELSE nonsp_tw = 0;
                IF nonsp_tw = 0 and (l542_2 = . OR f542_2 = . OR v542_2 = .) THEN nonsp_tw = .;
        END;

        IF c9342_2 NE . THEN rtrial = (c9342_2 = 2);
        IF c342_2 NE . THEN rvh = (c342_2 = 6);
    END;
RUN;
```

4.3.3. FHS

```
PROC SORT DATA = fhs.ofsleep5; BY sid; RUN;
PROC SORT DATA = fhs.ofsleep6; BY sid; RUN;

DATA ofsleep;
    MERGE fhs.ofsleep5 fhs.ofsleep6;
    BYsid;
RUN;

DATA temp;
    SET ofsleep fhs.omsleep fhs.cohsleep;
    site = 'fhs';
    id = PUT (sid,12.);
    LENGTH pptid $ 12;
    pptid = ((LEFT(TRIM(sidtype)))||(LEFT(TRIM(id))));
    FORMAT psgdate ecgdate DATE7.;

    /*****************************************************************
    Offspring 5 and Omni:
    Find patients w/ ECG dates w/n 3 years before or 3 months after date of PSG
    Create variables
    *****************************************************************/
    IF sidtype = 's7' THEN date5 = date1;
    IF date5-shhsdate ge -(365*3) OR date5-shhsdate le 91.2 THEN DO;
        psgdate = shhsdate;
        ecgdate = date5;

        IF e607 NE . OR e602 NE . OR e605 NE . THEN DO;
            lvh3_1 = (e607 = 1 OR e602 = 1 OR e605>12);
            IF lvh3_1 = 0 and (e607 = . OR e602 = . OR e605 = .) THEN lvh3_1 = .;
        END;

        IF e611 NE . THEN lvh3_3 = (e611 = 1);
        IF e618 NE . THEN st4_1_3 = (e618 IN (1 2));

        IF lvh3_1 NE . OR lvh3_3 NE . OR st4_1_3 NE . THEN DO;
            lvhst = ((lvh3_1 = 1 OR lvh3_3 = 1) and (st4_1_3 = 1));
            IF lvhst = 0 and (lvh3_1 = . OR lvh3_3 = . OR st4_1_3 = .) THEN lvhst = .;
        END;

        ventrate = e584;
        qrs = e588;
        IF e589 NE . THEN mob1 = (e589 = 3);
        IF e589 NE . THEN mob2 = (e589 = 4);
        IF e589 NE . THEN av3deg = (e589 = 5);
        IF e589 NE . THEN afib = (e589 = 6);
        IF e589 NE . THEN nodal = (e589 = 7);
```

```
IF e589 NE . THEN paced = (e589 = 8);
IF e585 NE . THEN av1deg = (e585 ge 22);

IF e590 NE . OR e591 NE . OR e592 NE . THEN DO;
     lbbb = (e590 = 1 and e591 = 1 and e592 = 1);
     IF lbbb = 0 and (e590 = . OR e591 = . OR e592 = .) THEN lbbb = .;
END;

IF e590 NE . OR e591 NE . OR e592 NE . THEN DO;
     rbbb = (e590 = 1 and e591 = 2 and e592 = 1);
     IF rbbb = 0 and (e590 = . OR e591 = . OR e592 = .) THEN rbbb = .;
END;

IF e590 NE . OR e591 NE . OR e593 NE . THEN DO;
     ilbbb = (e590 = 1 and e591 = 1 and e593 = 1);
     IF ilbbb = 0 and (e590 = . OR e591 = . OR e593 = .) THEN ilbbb = .;
END;

IF e590 NE . OR e591 NE . OR e593 NE . THEN DO;
     irbbb = (e590 = 1 and e591 = 2 and e593 = 1);
     IF irbbb = 0 and (e590 = . OR e591 = . OR e593 = .) THEN irbbb = .;
END;

IF e594 NE . THEN lah = (e594 = 1);
IF e590 NE . OR e591 NE . OR e592 NE . THEN DO;
     iventblk = (e590 = 1 and e591 = 3 and e592 = 1);
     IF iventblk = 0 and (e590 = . OR e591 = . OR e592 = .) THEN iventblk = .;
END;

IF e595 NE . THEN wpw = (e595 = 1);
IF e596 NE . THEN apbs = (e596 = 1);
IF e597 NE . THEN vpbs = (e597 IN (1 2 3 4 5));
antsepmi = e599;
infmi = e600;
truposmi = e601;
nonsp_st = e613;
nonsp_tw = e614;
IF e616 NE . THEN rtrial = (e616 = 2);
IF e617 NE . THEN rvh = (e617 = 1);
END;
```

4.3.3. FHS

```
/*****************************************************************
Offspring 6:
Find patients w/ ECG dates w/n 3 years before or 3 months after date of PSG
Create variables
*****************************************************************/
ELSE IF date6-shhsdate ge -(365*3) OR date6-shhsdate le 91.2 THEN DO;
    psgdate = shhsdate;
    ecgdate = date6;
    IF f602 NE . OR f597 NE . OR f600 NE . THEN DO;
        lvh3_1 = (f602 = 1 OR f597 = 1 OR f600>12);
        IF lvh3_1 = 0 and (f602 = . OR f597 = . OR f600 = .) THEN lvh3_1 = .;
    END;

    IF f605 NE . THEN lvh3_3 = (f605 = 1);
    IF f613 NE . THEN st4_1_3 = (f613 IN (1 2));
    IF lvh3_1 NE . OR lvh3_3 NE . OR st4_1_3 NE . THEN DO;
        lvhst = ((lvh3_1 = 1 OR lvh3_3 = 1) and (st4_1_3 = 1));
        IF lvhst = 0 and (lvh3_1 = . OR lvh3_3 = . OR st4_1_3 = .) THEN lvhst = .;
    END;

    ventrate = f579;
    qrs = f583;
    IF f584 NE . THEN mob1 = (f584 = 3);
    IF f584 NE . THEN mob2 = (f584 = 4);
    IF f584 NE . THEN av3deg = (f584 = 5);
    IF f584 NE . THEN afib = (f584 = 6);
    IF f584 NE . THEN nodal = (f584 = 7);
    IF f584 NE . THEN paced = (f584 = 8);
    IF f580 NE . THEN av1deg = (f580 ge 22);

    IF f585 NE . OR f586 NE . OR f587 NE . THEN DO;
        lbbb = (f585 = 1 and f586 = 1 and f587 = 1);
        IF lbbb = 0 and (f585 = . OR f586 = . OR f587 = .) THEN lbbb = .;
    END;

    IF f585 NE . OR f586 NE . OR f587 NE . THEN DO;
        rbbb = (f585 = 1 and f586 = 2 and f587 = 1);
        IF rbbb = 0 and (f585 = . OR f586 = . OR f587 = .) THEN rbbb = .;
    END;

    IF f585 NE . OR f586 NE . OR f588 NE . THEN DO;
        ilbbb = (f585 = 1 and f586 = 1 and f588 = 1);
        IF ilbbb = 0 and (f585 = . OR f586 = . OR f588 = .) THEN ilbbb = .;
    END;
```

```
    IF f585 NE . OR f586 NE . OR f588 NE . THEN DO;
        irbbb = (f585 = 1 and f586 = 2 and f588 = 1);
        IF irbbb = 0 and (f585 = . OR f586 = . OR f588 = .) THEN irbbb = .;
    END;

    IF f589 NE . THEN lah = (f589 = 1);
    IF f585 NE . OR f586 NE . OR f587 NE . THEN DO;
        iventblk = (f585 = 1 and f586 = 3 and f587 = 1);
        IF iventblk = 0 and (f585 = . OR f586 = . OR f587 = .) THEN iventblk = .;
    END;

    IF f590 NE . THEN wpw = (f590 = 1);
    IF f591 NE . THEN apbs = (f591 = 1);
    IF f592 NE . THEN vpbs = (f592 IN (1 2 3 4 5));
    antsepmi = f594;
    infmi = f595;
    truposmi = f596;
    nonsp_st = f608;
    nonsp_tw = f609;
    IF f611 NE . THEN rtrial = (f611 = 2);
    IF f612 NE . THEN rvh = (f612 = 1);
END;

/*****************************************************************
Cohort:
Find patients w/ ECG dates w/n 3 years before or 3 months after date of PSG
Create variables
*****************************************************************/
ELSE IF date24-shhsdate ge -(365*3) OR date24-shhsdate le 91.2 THEN DO;
    psgdate = shhsdate;
    ecgdate = date24;
    IF fq290 NE . OR fq285 NE . OR fq288 NE . THEN DO;
        lvh3_1 = (fq290 = 1 OR fq285 = 1 OR fq288>12);
        IF lvh3_1 = 0 and (fq290 = . OR fq285 = . OR fq288 = .) THEN lvh3_1 = .;
    END;
    IF fq293 NE . THEN lvh3_3 = (fq293 = 1);
    IF fq301 NE . THEN st4_1_3 = (fq301 IN (1 2));
    IF lvh3_1 NE . OR lvh3_3 NE . OR st4_1_3 NE . THEN DO;
        lvhst = ((lvh3_1 = 1 OR lvh3_3 = 1) and (st4_1_3 = 1));
        IF lvhst = 0 and (lvh3_1 = . OR lvh3_3 = . OR st4_1_3 = .) THEN lvhst = .;
    END;
```

```
        ventrate = fq267;
        qrs = fq271;
        IF fq272 NE . THEN mob1 = (fq272 = 3);
        IF fq272 NE . THEN mob2 = (fq272 = 4);
        IF fq272 NE . THEN av3deg = (fq272 = 5);
        IF fq272 NE . THEN afib = (fq272 = 6);
        IF fq272 NE . THEN nodal = (fq272 = 7);
        IF fq272 NE . THEN paced = (fq272 = 8);
        IF fq268 NE . THEN av1deg = (fq268 ge 22);

        IF fq273 NE . OR fq274 NE . OR fq275 NE . THEN DO;
            lbbb = (fq273 = 1 and fq274 = 1 and fq275 = 1);
            IF lbbb = 0 and (fq273 = . OR fq274 = . OR fq275 = .) THEN lbbb = .;
        END;

        IF fq273 NE . OR fq274 NE . OR fq275 NE . THEN DO;
            rbbb = (fq273 = 1 and fq274 = 2 and fq275 = 1);
            IF rbbb = 0 and (fq273 = . OR fq274 = . OR fq275 = .) THEN rbbb = .;
        END;

        IF fq273 NE . OR fq274 NE . OR fq276 NE . THEN DO;
            ilbbb = (fq273 = 1 and fq274 = 1 and fq276 = 1);
            IF ilbbb = 0 and (fq273 = . OR fq274 = . OR fq276 = .) THEN ilbbb = .;
        END;

        IF fq273 NE . OR fq274 NE . OR fq276 NE . THEN DO;
            irbbb = (fq273 = 1 and fq274 = 2 and fq276 = 1);
            IF irbbb = 0 and (fq273 = . OR fq274 = . OR fq276 = .) THEN irbbb = .;
        END;

        IF fq277 NE . THEN lah = (fq277 = 1);
        IF fq273 NE . OR fq274 NE . OR fq275 NE . THEN DO;
            iventblk = (fq273 = 1 and fq274 = 3 and fq275 = 1);
            IF iventblk = 0 and (fq273 = . OR fq274 = . OR fq275 = .) THEN iventblk = .;
        END;

        IF fq278 NE . THEN wpw = (fq278 = 1);
        IF fq279 NE . THEN apbs = (fq279 = 1);
        IF fq280 NE . THEN vpbs = (fq280 IN (1 2 3 4 5));
        antsepmi = fq282;
        infmi = fq283;
        truposmi = fq284;
        nonsp_st = fq296;
        nonsp_tw = fq297;
        IF fq299 NE . THEN rtrial = (fq299 = 2);
        IF fq300 NE . THEN rvh = (fq300 = 1);
    END;
RUN;
```

4.3.4. NYU

```
** Pull in PSG date from the SHHS 1 DATASET;
DATA s1data;
	SET shhs1.shhs1final_01oct2008_6441;
	RENAME stdydtqa = shhsdate;
	IF clinic NE "41" THEN DELETE;
	KEEP pptid stdydtqa;
RUN;

PROC SORT DATA = s1data; BY pptid; RUN;

** Pull in ECG DATA from NY;
DATA nyecg;
	SET nyu.ekg_ny;
RUN;

PROC SORT DATA = nyecg; BY pptid; RUN;

DATA temp;
	MERGE s1data (IN = A) nyecg (IN = B);
	BY pptid;
RUN;

DATA nyu.ecg_nyu;
	SET temp;
	FORMAT psgdate ecgdate DATE7.;
	psgdate = DATEPARTt(shhsdate);
	ecgdate = ecgdate;

	lvh3_1 = 0;
	lvh3_3 = 0;
	st4_1_3 = 0;
	st5_1_3 = 0;
	lvhst = 0;
	ventrate = 0;
	qrs = 0;
	mob1 = 0;
	part2deg = 0;
	mob2 = 0;
	av3deg = 0;
	afib = 0;
	nodal = 0;
	paced = 0;
	av1deg = 0;
	lbbb = 0;
	rbbb = 0;
	ilbbb = 0;
	irbbb = 0;
```

```
lah = 0;
iventblk = 0;
wpw = 0;
apbs = 0;
vpbs = 0;
antsepmi = 0;
infmi = 0;
truposmi = 0;
antlatmi = 0;
nonsp_st = 0;
nonsp_tw = 0;
rtrial = 0;
rvh = 0;

IF q17 = 1 OR  q18 = 1 OR q22 = 1 THEN lvh3_1 = 1;
IF q25 = 1 THEN lvh3_3 = 1;
IF q28 NE . THEN st4_1_3 = (q28 = 1);
IF q28 NE . THEN st5_1_3 = (q28 IN (1,2,3));

IF lvh3_1 NE . OR lvh3_3 NE . OR st4_1_3 NE . THEN DO;
      lvhst = ((lvh3_1 = 1 OR lvh3_3 = 1) and (st4_1_3 = 1));
      IF lvhst = 0 and (lvh3_1 = . OR lvh3_3 = . OR st4_1_3 = .) THEN lvhst = .;
END;

ventrate = q2;
qrs = q4;
IF q7 NE . THEN mob1 = (q7 = 3);
part2deg = .;
IF q7 NE . THEN mob2 = (q7 = 4);
IF q7 NE . THEN av3deg = (q7 = 5);
IF q7 NE . THEN afib = (q7 = 6);
IF q7 NE . THEN nodal = (q7 = 7);
IF q7 NE . THEN paced = (q7 = 8);
IF q7 NE . THEN av1deg = (q7 IN (0,1));
IF q33 = 9 THEN lbbb = 1;
IF q32 = 9 THEN rbbb = 1;
IF q8 = 1 and q8_a = 1 and q8_c = 1 THEN ilbbb = 1;
IF q8 = 1 and q8_a = 2 and q8_c = 1 THEN irbbb = 1;
IF q9 NE . THEN lah = (q9 = 1);
IF q8 = 1 and q8_a = 3 THEN iventblk = 1;
IF q10 NE . THEN wpw = (q10 = 1);
IF q11 NE . THEN apbs = (q11 IN (1,2));
IF q12 NE . THEN vpbs = (q12 IN (1,2,3,4,5));
IF q14 = 1 THEN antsepmi = 1;
IF q15 = 1 THEN infmi = 1;
IF q16 = 1 THEN truposmi = 1;
antlatmi = .;
```

```
    IF q28 = 1 THEN nonsp_st = 1;
    IF q29 = 1 THEN nonsp_tw = 1;
    IF q31 IN (1,2,3) THEN rtrial = 1;
    IF q32 NE . THEN rvh = (q32 = 1);

    KEEP pptid psgdate ecgdate lvh3_1 lvh3_3 st4_1_3 st5_1_3 lvhst ventrate qrs
    mob1 part2deg mob2 av3deg afib nodal paced av1deg lbbb rbbb ilbbb irbbb lah iventblk
    wpw apbs vpbs antsepmi infmi truposmi antlatmi nonsp_st nonsp_tw rtrial rvh;
RUN;
```

4.3.5. SHS

```
DATA temp;
    SET shs.sleepecg (RENAME = (ecgdate = dateecg));
    site = 'shs';
    LENGTH pptid $ 12;
    pptid = idno;
    month = SUBSTR(dateecg,1,2);
    day = SUBSTR(dateecg,3,2);
    year = SUBSTR(dateecg,5,2);
    ecgdate = MDY(month,day,year);
    psgdate = sleepday;
    FORMAT psgdate ecgdate DATE7.;

    /*****************************************************************
    Find patients with ECG dates within 3 years before date of PSG
    Create variables
    *****************************************************************/
    IF ecgdate-psgdate GE -(365*3) AND ecgdate-psgdate LE 91.2  THEN DO;
        IF m3x NE . THEN lvh3_1 = (m3x = 1);
        IF m3x NE . THEN lvh3_3 = (m3x = 3);
        IF m4xx_1 NE . OR m4xx_2 NE . OR m4xx_3 NE . THEN DO;
            st4_1_3 = (m4xx_1 IN (2 3 4 11 12) OR m4xx_2 IN (2 3 4 11 12) OR m4xx_3 IN (2 3 4 11
            12));
            IF st4_1_3 = 0 and (m4xx_1 = . OR m4xx_2 = . OR m4xx_3 = .) THEN st4_1_3 = .;
        END;

        IF m5x_1 NE . OR m5x_2 NE . OR m5x_3 NE . THEN DO;
            st5_1_3 = (m5x_1 IN (1 2 3 4) OR m5x_2 IN (1 2 3 4) OR m5x_3 IN (1 2 3 4));
            IF st5_1_3 = 0 and (m5x_1 = . OR m5x_2 = . OR m5x_3 = .) THEN st5_1_3 = .;
        END;

        ventrate = heartrat;
        IF m6xx NE . THEN mob1 = (m6xx = 2);
        IF m6xx NE . THEN part2deg = (m6xx = 2);
        IF m6xx NE . THEN mob2 = (m6xx = 2);
        IF m6xx NE . THEN av3deg = (m6xx = 1);
        IF m6xx NE . THEN paced = (m6xx = 8);
        IF m6xx NE . THEN av1deg = (m6xx = 3);
        IF m7xx NE . THEN lbbb = (m7xx = 11);
        IF m7xx NE . THEN rbbb = (m7xx = 21);
        IF m7xx NE . THEN ilbbb = (m7xx = 6);
        IF m7xx NE . THEN irbbb = (m7xx = 3);
        IF m7xx NE . THEN lah = (m7xx = 7);
        IF m7xx NE . THEN iventblk = (m7xx = 4);
        IF m6xx NE . THEN wpw = (m6xx = 4);
```

```
        IF m1xx_3 NE . THEN DO;
            IF m7xx NE 11 and m7xx NE 4 THEN DO;
                IF m1xx_3 IN (14 16 24 25 27) THEN antsepmi = 1;
            ELSE IF m1xx_3 IN (26 28 31 32 34 35 36) THEN antsepmi = 2;
            ELSE antsepmi = 0;
            END;
        END;
        IF m1xx_2 NE . THEN DO;
            IF m7xx NE 11 and m7xx NE 4 THEN DO;
                IF m1xx_2 IN (14 16 24 25 27) THEN infmi = 1;
            ELSE IF m1xx_2 IN (26 28 31 32 34 35 36) THEN infmi = 2;
            ELSE infmi = 0;
            END;
        END;

        IF m1xx_1 NE . THEN DO;
            IF m7xx NE 11 and m7xx NE 4 THEN DO;
                IF m1xx_1 IN (14 16 24 25 27) THEN antlatmi = 1;
            ELSE IF m1xx_1 IN (26 28 31 32 34 35 36) THEN antlatmi = 2;
            ELSE antlatmi = 0;
            END;
        END;

        IF m4xx_1 NE . OR m4xx_2 NE . OR m4xx_3 NE . THEN DO;
            nonsp_st = (m4xx_1 IN (2 3 11 12) OR m4xx_2 IN (2 3 11 12) OR m4xx_3 IN (2 3 11 12));
            IF nonsp_st = 0 and (m4xx_1 = . OR m4xx_2 = . OR m4xx_3 = .) THEN nonsp_st = .;
        END;

        IF m5x_1 NE . OR m5x_2 NE . OR m5x_3 NE . THEN DO;
            nonsp_tw = (m5x_1 IN (1 2 3) OR m5x_2 IN (1 2 3) OR m5x_3 IN (1 2 3));
            IF nonsp_tw = 0 and (m5x_1 = . OR m5x_2 = . OR m5x_3 = .) THEN nonsp_tw = .;
        END;

        IF m3x NE . THEN rvh = (m3x = 2);
    END;
RUN;
```

4.3.6. Tucson

```
DATA tempa;
	SET ids.psgdate6441;
	WHERE site = 64;
	pptid = (LEFT(TRIM(UPCASE(pptid))));
	datepsg = PUT (stdydtqa, DATETIME18.);
	psgdate = INPUT ((SUBSTR((LEFT(TRIM(datepsg))),1,7)),DATE7.);
	FORMAT psgdate DATE7.;
	KEEP pptid psgdate;
RUN;

PROC SORT DATA = tempa; BY pptid; RUN;

DATA tempb;
	SET tuc.ecg (RENAME = (ecgdate = tempedt));
	dateecg = PUT (tempedt, DATETIME18.);
	ecgdate = INPUT((SUBSTR((LEFT(TRIM(dateecg))),1,7)),DATE7.);
	FORMAT ecgdate DATE7.;
	pptid = PUT ((LEFT(TRIM(UPCASE(id)))),12.);
	LENGTH code1 code2 code3 code4 $ 8;
	IF __1 NE '' OR __2 NE '' OR __3 NE '' THEN
		code1 = (LEFT(TRIM(__1)))||'_'||(LEFT(TRIM(__2)))||'_'||(LEFT(TRIM(__3)));

	IF __10 NE '' OR __20 NE '' OR __30 NE '' THEN
		code2 = (LEFT(TRIM(__10)))||'_'||(LEFT(TRIM(__20)))||'_'||(LEFT(TRIM(__30)));

	IF __11 NE '' OR __21 NE '' OR __31 NE '' THEN
		code3 = (LEFT(TRIM(__11)))||'_'||(LEFT(TRIM(__21)))||'_'||(LEFT(TRIM(__31)));

	IF __12 NE '' OR __22 NE '' OR __32 NE '' THEN
		code4 = (LEFT(TRIM(__12)))||'_'||(LEFT(TRIM(__22)))||'_'||(LEFT(TRIM(__32)));

	KEEP pptid ecgdate code1 code2 code3 code4;
RUN;

PROC SORT DATA = tempb; BY ptid; RUN;

DATA temp;
	MERGE tempa (IN = A) tempb (IN = B);
	BY pptid;
	IF B;
	site = 'tuc';
```

```
/*****************************************************************
Find patients w/ ECG dates w/n 3 years before or 3 months after date of PSG
Create variables
*****************************************************************/
IF ecgdate-psgdate GE -(365*3) AND ecgdate-psgdate LE 91.2 THEN DO;
    lvh3_1 = (code1 = '3_1_0' OR code2 = '3_1_0' OR code3 = '3_1_0' OR code4 = '3_1_0');
    st4_1_3 = (code1 IN ('4_1_0' '4_2_0' '4_3_0') OR code2 IN ('4_1_0' '4_2_0' '4_3_0')
        OR code3 IN ('4_1_0' '4_2_0' '4_3_0')  OR code4 IN ('4_1_0' '4_2_0' '4_3_0'));
    st5_1_3 = (code1 IN ('5_1_0' '5_2_0' '5_2_3' '5_3_0') OR code2 IN ('5_1_0' '5_2_0' '5_2_3' '5_3_0')
        OR code3 IN ('5_1_0' '5_2_0' '5_2_3' '5_3_0')  OR code4 IN ('5_1_0' '5_2_0' '5_2_3' '5_3_0'));
    lvhst = (lvh3_1 = 1 and (st4_1_3 = 1 OR st5_1_3 = 1));
    mob1 = (code1 = '6_2_3' OR code2 = '6_2_3'  OR code3 = '6_2_3' OR code4 = '6_2_3');
    part2deg = (code1 = '6_2_2' OR code2 = '6_2_2' OR code3 = '6_2_2' OR code4 = '6_2_2');
    mob2 = (code1 = '6_2_1' OR code2 = '6_2_1' OR code3 = '6_2_1' OR code4 = '6_2_1');
    av3deg = (code1 = '6_1_0' OR code2 = '6_1_0' OR code3 = '6_1_0' OR code4 = '6_1_0');
    afib = (code1 = '8_3_0' OR code2 = '8_3_0' OR code3 = '8_3_0' OR code4 = '8_3_0');
    nodal = (code1 = '8_4_0' OR code2 = '8_4_0' OR code3 = '8_4_0' OR code4 = '8_4_0');
    paced = (code1 = '6_8_0' OR code2 = '6_8_0' OR code3 = '6_8_0' OR code4 = '6_8_0');
    av1deg = (code1 = '6_3_0' OR code2 = '6_3_0' OR code3 = '6_3_0' OR code4 = '6_3_0');
    lbbb = (code1 = '7_1_0' OR code2 = '7_1_0' OR code3 = '7_1_0' OR code4 = '7_1_0');
    rbbb = (code1 = '7_2_0' OR code2 = '7_2_0' OR code3 = '7_2_0' OR code4 = '7_2_0');
    irbbb = (code1 = '7_3_0' OR code2 = '7_3_0' OR code3 = '7_3_0' OR code4 = '7_3_0');
    iventblk = (code1 = '7_4_0' OR code2 = '7_4_0' OR code3 = '7_4_0' OR code4 = '7_4_0');
    wpw = (code1 = '6_4_0' OR code2 = '6_4_0' OR code3 = '6_4_0' OR code4 = '6_4_0');
    apbs = (code1 = '8_1_0' OR code2 = '8_1_0' OR code3 = '8_1_0' OR code4 = '8_1_0');
    vpbs = (code1 = '8_2_0' OR code2 = '8_2_0' OR code3 = '8_2_0' OR code4 = '8_2_0');

    IF code1 IN ('1_1_6' '1_1_7' '1_2_7') OR code2 IN ('1_1_6' '1_1_7' '1_2_7')
        OR code3 IN ('1_1_6' '1_1_7' '1_2_7') OR code4 IN ('1_1_6' '1_1_7' '1_2_7') THEN antsepmi =
        1;
    ELSE IF code1 IN ('1_2_8' '1_3_2') OR code2 IN ('1_2_8' '1_3_2') OR code3 IN ('1_2_8' '1_3_2')
        OR code4 IN ('1_2_8' '1_3_2') THEN antsepmi = 2;
    ELSE antsepmi = 0;
        IF code1 IN ('1_1_4' '1_1_5' '1_2_4' '1_2_5') OR code2 IN ('1_1_4' '1_1_5' '1_2_4' '1_2_5')
            OR code3 IN ('1_1_4' '1_1_5' '1_2_4' '1_2_5')
            OR code4 IN ('1_1_4' '1_1_5' '1_2_4' '1_2_5') THEN infmi = 1;
    ELSE IF code1 IN ('1_2_6' '1_3_2') OR code2 IN ('1_2_6' '1_3_2') OR code3 IN ('1_2_6' '1_3_2')
        OR code4 IN ('1_2_6' '1_3_2') THEN infmi = 2;
    ELSE infmi = 0;
        IF code1 IN ('1_1_1' '1_1_2' '1_1_3' '1_2_1' '1_2_2' '1_2_3')
            OR code2 IN ('1_1_1' '1_1_2' '1_1_3' '1_2_1' '1_2_2' '1_2_3')
            OR code3 IN ('1_1_1' '1_1_2' '1_1_3' '1_2_1' '1_2_2' '1_2_3')
            OR code4 IN ('1_1_1' '1_1_2' '1_1_3' '1_2_1' '1_2_2' '1_2_3') THEN antlatmi = 1;
    ELSE IF code1 IN ('1_3_1' '1_3_3') OR code2 IN ('1_3_1' '1_3_3')
        OR code3 IN ('1_3_1' '1_3_3') OR code4 IN ('1_3_1' '1_3_3') THEN antlatmi = 2;
```

```
        ELSE antlatmi = 0;
            IF code1 IN ('4_1_0' '4_2_0') OR code2 IN ('4_1_0' '4_2_0') OR code3 IN ('4_1_0' '4_2_0')
                OR code4 IN ('4_1_0' '4_2_0') THEN nonsp_st = 1;
        ELSE IF code1 = '4_3_0' OR code2 = '4_3_0' OR code3 = '4_3_0' OR code4 = '4_3_0' THEN
        nonsp_st = 2;
        ELSE nonsp_st = 0;
            IF code1 IN ('5_1_0' '5_2_0' '5_2_3') OR code2 IN ('5_1_0' '5_2_0' '5_2_3')
                OR code3 IN ('5_1_0' '5_2_0' '5_2_3') OR code4 IN ('5_1_0' '5_2_0' '5_2_3') THEN
                nonsp_tw = 1;
        ELSE IF code1 = '5_3_0' OR code2 = '5_3_0' OR code3 = '5_3_0' OR code4 = '5_3_0' THEN
        nonsp_tw = 2;
        ELSE nonsp_tw = 0;
            rvh = (code1 = '3_2_0' OR code2 = '3_2_0' OR code3 = '3_2_0' OR code4 = '3_2_0');
    END;
RUN;
```

5. Followup 1

5.1. Characteristics of Followup 1 database

Name of dataset: *followupfinal_ddmonyyyy_6441.sas7bdat*

Number of records: 6441 participants

Primary Key Field: PPTID - Participant Identifier (alpha - numeric)

File Formats: All files are SAS 9.1 Files

Convention: No naming convention

5.2. Descriptions of Data

- Screener
  - n=197
  - This data was collected via the Contact Form
- Status, BP and Weight Form (1FUPBP&W)
  - n=6102
- Sleep Habits Questionnaire (1FUPSQ2)
  - n=5518
- Indicator Variables (see Appendix 9.5)
  - Specific variables were created to indicate for each unique PPTID the presence of Follow up 1 forms in the concatenated database (i.e. 0=No 1=Yes)
  - Indicator variable = "1" if the form is available for the participant and = "0" if the form is not available

6. SHHS 2

6.1. Characteristics of SHHS 2 database

IMPORTANT NOTE: ALL SHHS 2 DATA FOR NEW YORK (CLINIC=41) HAVE BEEN DELETED.

Name of dataset: SHHS2Final_ddmonyyyy_4586_psg.sas7bdat

Number of records: 4586 participants

Number of Records: The SHHS 2 finalized dataset included n= 4586 records.

Primary Key Field: PPTID - Participant Identifier (alpha - numeric)

File Formats: All files are SAS 9.1 Files.

Convention

- Most SAS variable names or labels are identified by the associated form and form number:

FFrIIa#(s,t,u,v,….)

Where,

FF is the 2 letter form abbreviation
r is the form revision number
II is the item number
a is the subitem letter
is the first or second column
s,t,u,v, … are specify fields associated with respective items

For example, pm225b2 corresponds to the entry found in the Physical Measurements form, revision 2, item 25, subitem b, left leg.

- Datasets received from all sites were combined into one dataset. For items with more than one revision, all items were mapped onto the most current form revision

- Final datasets were created with additional edits to the combined datasets. Descriptions of these edits are provided below.

6.2. Description of data

- Description of data:

  Quality of Life
  - QL (n=3811)

  Sleep Habits
  - SH (n=4889)

  Health Interview
  - HI (n=4382)
    - 183 records deleted
    - For participants with repeated PSG, only the HI associated with the successful PSG was kept
    - For participants without a PSG, the last HI was kept

  Physical Measurements
  - PM (n=4399)
    - 26 records deleted
    - For participants with repeated PSG, only the PM associated with the successful PSG was kept
    - For participants without a PSG, the last PM was kept
    - Data collected outside the allowable time windows were edited out (i.e., converted to missing). See MOP SHHS II data collection and time windows

  Morning Survey
  - MS (n=3341)
    - 153 records deleted
    - For participants with repeated PSG, only the MS associated with the successful PSG was kept

  Medications
  - MD (n=4371)
    - 494 records deleted
    - Records marked as 'deleted' in the Meddb database were excluded
    - Records without detailed information on specific medication use were excluded
    - For duplicate records with the same form date, the record(s) with total number of medications=0 were excluded
    - If more than one MD, the one(s) with the same sequence number as the HI was kept. May have more than one record per participant.
    - MD2_NDC_uniq: dataset with NDC codes. One record per participant

Polysomnography

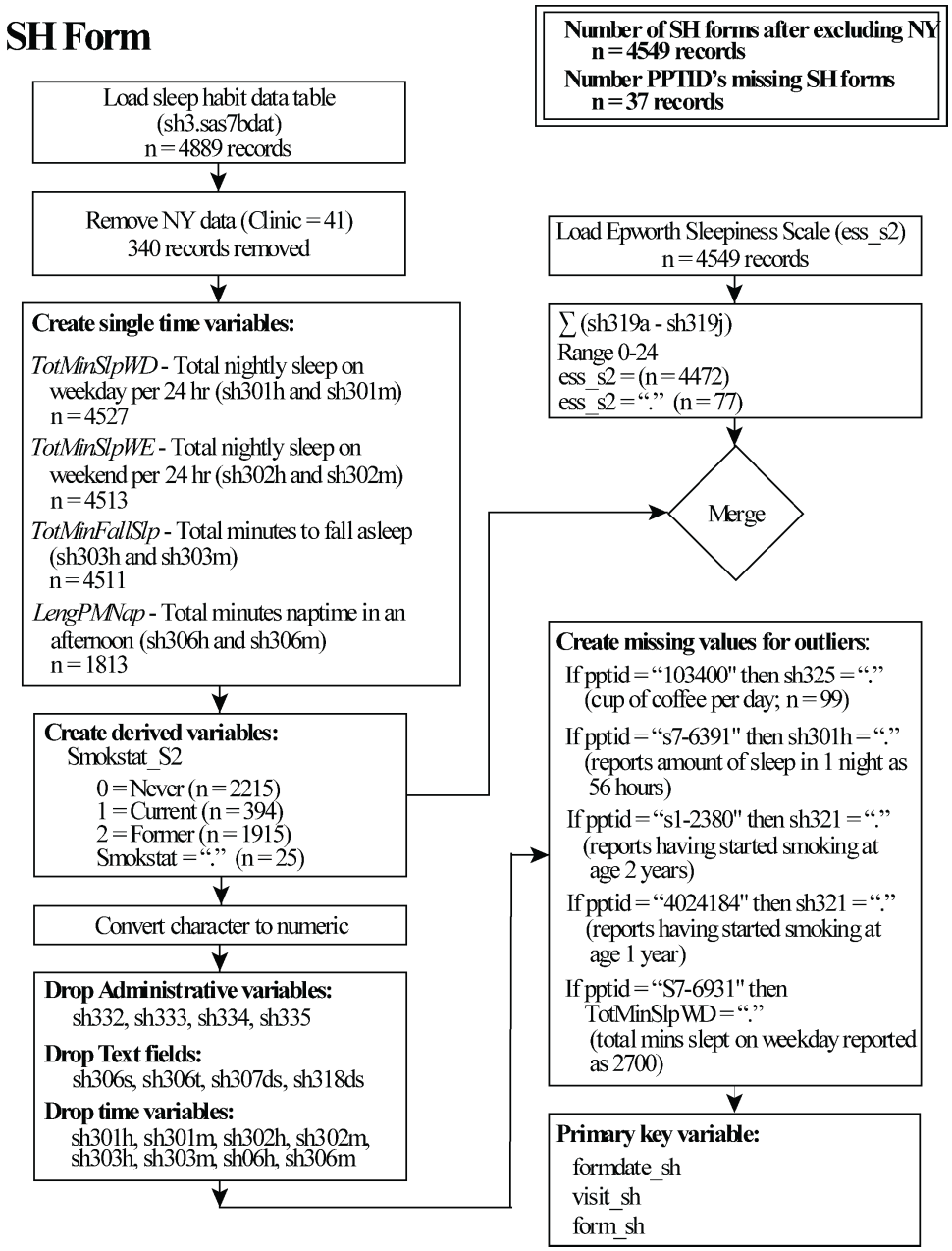
- PSG (n=3078)
  - Dataset as received by Reading Center *(shhs2psg_010606.sas7bdat)*
  - Unique PSG ID assigned

- Derived variables
  These were created and embedded within the datasets:
  - Age_s2
  - BMI_s2 - Body Mass Index
  - ESS_s2 - Epworth Sleepiness Score
  - SF36/SF36Sum - Medical Outcome Short Form (36) Health Survey
  - HTNDerv_s2 - Derived Hypertension Status
  - SmokStat_s2 - Smoking Status

- Single Time Variables
  - TotMinSlpWD - Total nightly sleep on weekday per 24 hr
  - TotMinSlpWE - Total nightly sleep on weekend per 24 hr
  - TotMinFallSlp - Total minutes to fall asleep
  - LengPMNap - Total minutes nap time in an afternoon
  - RptTimSlp - Reported time went to sleep last night
  - RptTimWak - Reported time awakeness today
  - RptActTimSltp - Reported time actually slept
  - RptElapTimSlpWak - Reported elapsed time between sleep and wakening
  - RptTimToSlp - Reported elapsed time between sleep and wakening
  - TimSlp - Time went to sleep last night
  - TotMinSlp - Total minutes slept last night
  - TotMinNap - Total minutes slept during nap

- Indicator variables (see Appendix 9.7)
  - Specific variables were created to indicate for each unique PPTID the presence of SHHS 2 forms in the concatenated database (i.e. 0=No 1=Yes)
  - Indicator variable = "1" if the form is available for the participant and = "0" if the form is not available

6.3. SHHS 2 data set development



HI Form

Load Health Interview data table
(hi2.sas7bdat)
n = 4382 records

↓

Remove NY data (Clinic = 41)
217 records removed

↓

Create single time variables:

TimSlp - Time went to sleep last night
(hi203h and hi203m)
n = 4116

TotMinSlp - Total minutes slept last night
(hi204h and hi204m)
n = 4113

TotMinNap - Total minutes slept during
nap (hi206h and hi206m)
n = 3979

↓

Convert character to numeric

↓

Drop Administrative variables:
hi217, hi218, hi219, hi220

Drop time variables:
hi203h, hi203m, hi204h, hi204m,
hi206h, hi206m

↓

Primary key variable:
formdate_hi
visit_hi
form_hi
formseq: 01 (n = 3946), 02 (n = 167),
03 (n = 3)

Number of HI forms after excluding NY (n = 4116)
Number of PPTID's missing HI forms (n = 35)

MS Form

Load morning survey data table
(ms2.sas7bdat)
n = 3321 records

↓

Remove NY data (Clinic = 41)
217 records removed

↓

Create time variables:

RptTimSlp - Reported time went to sleep last night (ms201h, ms201m, ms201ap)
n = 3124

RptTimWak - Reported time awakening today (ms202h, ms202m, ms202ap)
n = 3124

RptActTimSlpt - Reported time actually slept (ms203h, ms202m)
n = 3082

RptElapTimSlpWak - Reported elapsed time between sleep and wakening (RptTimSlp, RptTimWak)
n = 3097

RptTimToSlp - Reported length of time to fall asleep (ms206h, ms206m)
n = 3075

↓

Convert character to numeric

↓

Drop Administrative variables:
ms213, ms213s, ms214, ms215

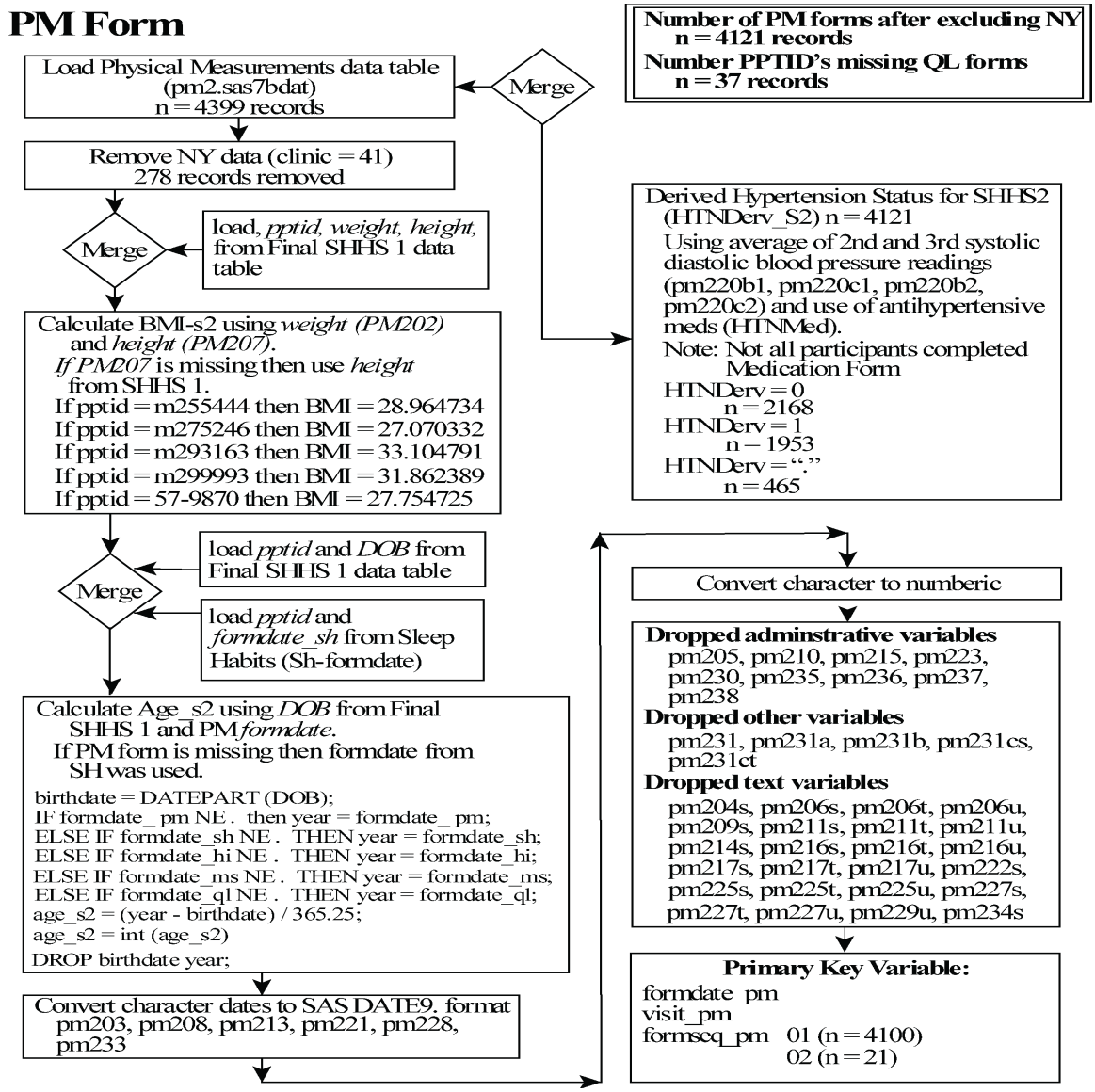
Drop time variables:
ms202h, ms202m, ms202ap, ms203h, ms203m, ms203ap, ms206h, ms206m

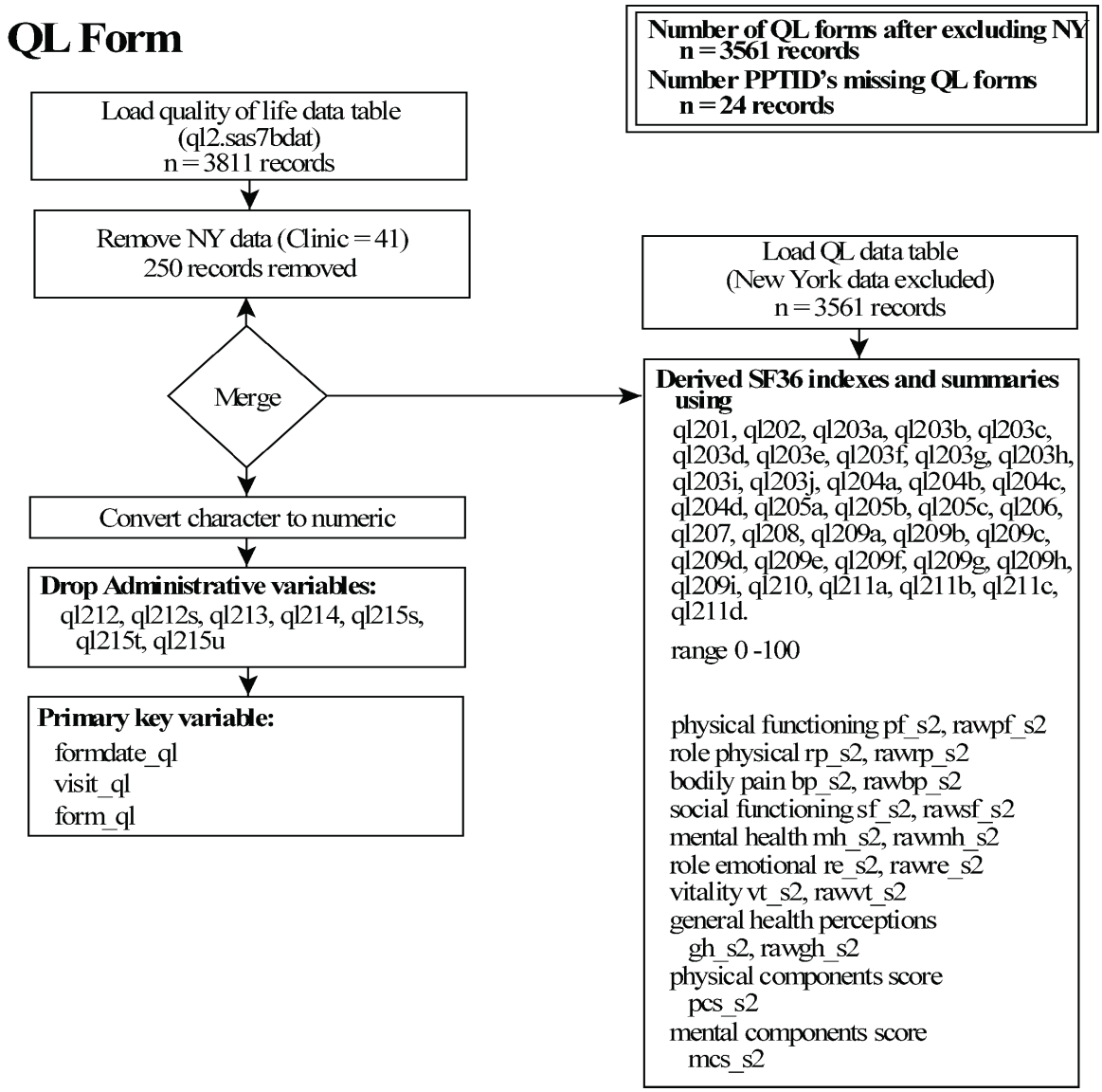
↓

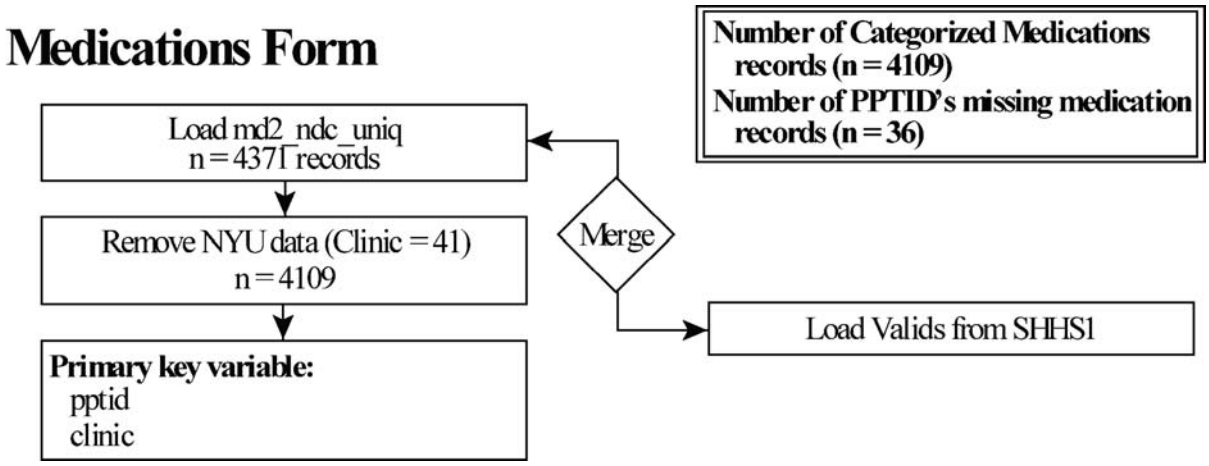
Primary key variable:
formdate_ms
visit_ms
form_ms
formseq

Number of MS forms after excluding NY (n = 3124)
Number of PPTID's missing MS forms (n = 19)

6.3. SHHS 2 data set development




QL Form
Number of QL forms after excluding NY
n = 3561 records
Number PPTID's missing QL forms
n = 24 records
Load quality of life data table
(ql2.sas7bdat)
n = 3811 records
Remove NY data (Clinic = 41)
250 records removed
Merge
Convert character to numeric
Drop Administrative variables:
ql212, ql212s, ql213, ql214, ql215s,
ql215t, ql215u
Primary key variable:
formdate_ql
visit_ql
form_ql
Load QL data table
(New York data excluded)
n = 3561 records
Derived SF36 indexes and summaries using
ql201, ql202, ql203a, ql203b, ql203c,
ql203d, ql203e, ql203f, ql203g, ql203h,
ql203i, ql203j, ql204a, ql204b, ql204c,
ql204d, ql205a, ql205b, ql205c, ql206,
ql207, ql208, ql209a, ql209b, ql209c,
ql209d, ql209e, ql209f, ql209g, ql209h,
ql209i, ql210, ql211a, ql211b, ql211c,
ql211d.
range 0 -100
physical functioning pf_s2, rawpf_s2
role physical rp_s2, rawrp_s2
bodily pain bp_s2, rawbp_s2
social functioning sf_s2, rawsf_s2
mental health mh_s2, rawmh_s2
role emotional re_s2, rawre_s2
vitality vt_s2, rawvt_s2
general health perceptions
gh_s2, rawgh_s2
physical components score
pcs_s2
mental components score
mcs_s2


Medications Form
Load md2_ndc_uniq
n = 4371 records
Remove NYU data (Clinic = 41)
n = 4109
Primary key variable:
pptid
clinic
Merge
Load Valids from SHHS1
Number of Categorized Medications records (n = 4109)
Number of PPTID's missing medication records (n = 36)

6.4. Dataset processing specifics

6.4.1. Derived non-PSG SHHS 2 variables (SAS)

IMPORTANT NOTE: ALL SHHS 2 DATA FOR NEW YORK (CLINIC=41) HAS BEEN DELETED

CALCULATE age (Age_S2):

Date of Birth (DOB) variable is provided from the SHHS 1 dataset. We used the date of the Physical Measurements (PM) form to calculate current age. Please note the format for the DOB variable is SAS DATETIME format *(ddmmyyyy hh:mm:ss)* and SAS DATE7 format *(mm/dd/yyyy)* for FORMDATE.

```
DOB=DATEPART(DOB);
FORMAT DOB DATE7.;
FORMAT formdate DATE7.;

**    Calculate age;
psgdate = formdate;       /* Make sure formdate is present */
birthdate = DOB;*/

FORMAT psgdate DATE7.;
FORMAT birthdate DATE7.;

age = (psgdate - birthdate) / 365.25;
age = int(age);
```

CALCULATE BMI (BMI_S2):

A baseline BMI has been derived for the SHHS 2 dataset. NOTE: Height from the SHHS 1 was substituted for all participants missing a variable for height in the SHHS 2 dataset.

6.4.1. Derived non-PSG SHHS 2 variables (SAS)

```
/* Pull in SHHS 1 data */

DATA shhs1data;
      LENGTH pptid $7;
      SET shhs1.shhs1data25oct05_6441;
      KEEP pptid /* avars */ height /* end avars */;
RUN;

/* Pull in SHHS 2 PM2 form data to calculate BMI */

DATA pmdata_s2;
      SET pmdata_all_s2;
      KEEP pptid clinic pm202 pm207 pm220b1 pm220b2 pm220c1 pm220c2;
RUN;



DATA pmdata;
      MERGE pmdata_s2 (IN=a) shhs1data (IN=b);
      BY pptid;
      IF b AND NOT a THEN DELETE;
RUN;

/* Calculate BMI */

DATA bmi_s2;
      SET pmdata;
      BY pptid;
      IF first.pptid; *duplicates in this form;
      LENGTH name $7.;
      name=TRIM(LEFT(UPCASE(pptid)));
      bmiweight=(pm202+0)/10; *in kg;
      IF pm207 NE . OR pm207 NE "m" THEN bmiheight=(pm207+0)/10; *in cm;
      ELSE bmiheight=(height+0)/10; *in cm;
      bmi=bmiweight/((bmiheight*bmiheight)/10000);

      DROP clinic formdate pm202 pm207 pm220b1 pm220b2 pm220c1 pm220c2 bmiheight
      bmiweight height name;
RUN;
```

CALCULATE ESS (ESS_S2):

```
RENAME namecd=pptid sh319a=sitrd02 sh319b=watv02 sh319c=sitpub02 sh319d=pgrcar02
sh319e=lydwn02 sh319f=sittlk02 sh319g=sitlch02 sh319h=incar02;  /* these variables were renamed
to map to variable names used in the code to calculate ESS */

incar=incar02-1;     /* transform from 1-4 scale to 0-3 scale */
lydwn=lydwn02-1;
pgrcar=pgrcar02-1;
sitlch=sitlch02-1;
sitpub=sitpub02-1;
sitrd=sitrd02-1;
sittlk=sittlk02-1;
watv=watv02-1;

if incar>3 then incar=.;  /* remove outliers */
if lydwn>3 then lydwn=.;
if pgrcar>3 then pgrcar=.;
if sitlch>3 then sitlch=.;
if sitpub>3 then sitpub=.;
if sitrd>3 then sitrd=.;
if sittlk>3 then sittlk=.;
if watv>3 then watv=.;

if InCar=. or LyDwn=. or PgrCar=. or SitLch=. or SitPub=. or
SitRd=. or SitTlk=. or WaTV=. then do ESS=.; end;
else ESS=SUM(InCar,LyDwn,PgrCar,SitLch,SitPub,SitRd,SitTlk,WaTV);

drop /* InCar02 LyDwn02 PgrCar02 SitLch02 SitPub02 SitRd02 SitTlk02 WaTV02 */
incar lydwn pgrcar sitlch sitpub sitrd sittlk watv;
```

SF36/SF36Sum:

The following variables for SF36 scales as well as SF36Sum scores were included:

PF - PHYSICAL FUNCTIONING
BP - BODILY PAIN
GH - GENERAL HEALTH
VT - VITALITY
SF - SOCIAL FUNCTIONING
RE - ROLE EMOTIONAL

MH - MENTAL HEALTH INDEX

MCS - STANDARDIZED MENTAL COMPONENT
PCS - STANDARDIZED PHYSICAL COMPONENT

NOTE: No Quality of Life Variables (SF36 composite scores) exists for the following clinics:

61 = South Dakota
62 = Oklahoma
63 = Phoenix

CALCULATE SMOKING STATUS (SMOKSTAT_S2):

```
/* Pull in the SHHS 2 SH3 form data for SmokeStat variable */

DATA shdata_s2;
      LENGTH namecd $7;
      SET shhs2.sh3;
      namecd=UPCASE(namecd);
      IF clinic="41" THEN DELETE;
      RENAME namecd=pptid;
      FORMAT formdate DATE7.;
RUN;

/* Create SMOKSTAT variable where 0=Never, 1=Current, 2=Former */

DATA smoke_status_s2;
      SET shdata_s2;
      IF sh320=0 THEN smokstat=0;
      IF sh320=1 AND sh323=1 THEN smokstat=1;
      IF sh320=1 AND sh323=0 THEN smokstat=0;
      KEEP pptid smokstat;
RUN;
```

CALCULATE HYPERTENSION STATUS (HTNDerv_S2):

Utilizing the average of the 2nd and 3rd seated Blood Pressure Readings from the SHHS 2 Physical Measurements (PM) form and the HTNmeds variable from SHHS 2 Categorized Medication database. (Definition used by Nieto et al paper)

(SBP2 + SBP3/2 > 140/90 mm HG OR HTNMeds = 1 will be affirmative for hypertension.

```
/* Create HTNDerv variable wher 0=No and 1=Yes */

DATA htndata_pm;
      SET pmdata;
      KEEP pptid pm220b1 pm220b2 pm220c1 pm220c2;
RUN;

DATA htndata_med;
      LENGTH namecd $7;
      SET shhs2.md2_ndc_uniq;
      namecd=UPCASE(namecd);
      RENAME namecd=pptid;
      KEEP namecd htnmed;
RUN;

DATA htndata_merge;
      MERGE htndata_pm (IN=a) htndata_med (IN=b);
      BY pptid;
      IF b AND NOT a THEN DELETE;
RUN;

DATA htndata_s2;
      SET htndata_merge;

      /* Calculate average of 2nd and 3rd seated blood pressure readings */
      BY pptid;
      /*IF first.pptid; *duplicates in this form;*/
      LENGTH name $7.;
      name=TRIM(LEFT(UPCASE(pptid)));
      ARRAY bpori{4} $ pm220b1 pm220c1 pm220b2 pm220c2;
      ARRAY bpnew{4} bps2 bps3 bpd2 bpd3;
      DO i=1 TO 4;
```

```
   IF bpori{i} not in ("n", "?", " ", "m") THEN bpnew{i}=bpori{i}+0;
      ELSE bpnew{i}=.;
      END;

      ARRAY bp{4} bps2-bps3 bpd2-bpd3;
      DO i=1 TO 4;
   IF bp{i}=0 THEN bp{i}=.;
      ELSE bp{i}=bp{i};
      END;

      bps=((bps2+bps3)/2);
      bpd=((bpd2+bpd3)/2);

      DROP pm220b1 pm220c1 pm220b2 pm220c2 i bps2 bps3 bpd2 bpd3 name;

      IF ((bps GT 140 AND bpd GT 90) OR HTNMED=1) THEN HTNDerv=1;
      ELSE HTNDerv=0;

      DROP HTNMED bps bpd;
RUN;
```

CALCULATIONS FOR SINGLE TIME VARIABLES:
See Flow Charts for Health Interview, Morning Survey, and Sleep Habits Forms

If further clarification is needed, please contact the SHHS Coordinating Center via email at shhs@jhsph.edu

CALCULATIONS FOR FUNCTIONAL OUTCOMES OF SLEEP QUESTIONNAIRE (FOSQ) SCORE

```
** Transform scales;
** ITEM 1;
item01 = 0;
IF nfq201 = 1 THEN item01 = 4;
IF nfq201 = 2 THEN item01 = 3;
IF nfq201 = 3 THEN item01 = 2;
IF nfq201 = 4 THEN item01 = 1;

** ITEM 2;
item02 = 0;
IF nfq202 = 1 THEN item02 = 4;
IF nfq202 = 2 THEN item02 = 3;
IF nfq202 = 3 THEN item02 = 2;
IF nfq202 = 4 THEN item02 = 1;
```

```
** ITEM 3;
item03 = 0;
IF nfq203 = 1 THEN item03 = 4;
IF nfq203 = 2 THEN item03 = 3;
IF nfq203 = 3 THEN item03 = 2;
IF nfq203 = 4 THEN item03 = 1;

** ITEM 4;
IF nfq204 = 0 THEN item04 = 0;
IF nfq204 = 1 THEN item04 = 4;
IF nfq204 = 2 THEN item04 = 3;
IF nfq204 = 3 THEN item04 = 2;
IF nfq204 = 4 THEN item04 = 1;

** ITEM 5;
IF nfq205 = 0 THEN item05 = 0;
IF nfq205 = 1 THEN item05 = 4;
IF nfq205 = 2 THEN item05 = 3;
IF nfq205 = 3 THEN item05 = 2;
IF nfq205 = 4 THEN item05 = 1;

** ITEM 6;
IF nfq206 = 0 THEN item06 = 0;
IF nfq206 = 1 THEN item06 = 4;
IF nfq206 = 2 THEN item06 = 3;
IF nfq206 = 3 THEN item06 = 2;
IF nfq206 = 4 THEN item06 = 1;

** ITEM 7;
IF nfq207 = 0 THEN item07 = 0;
IF nfq207 = 1 THEN item07 = 4;
IF nfq207 = 2 THEN item07 = 3;
IF nfq207 = 3 THEN item07 = 2;
IF nfq207 = 4 THEN item07 = 1;

** ITEM 8;
IF nfq208 = 0 THEN item08 = 0;
IF nfq208 = 1 THEN item08 = 4;
IF nfq208 = 2 THEN item08 = 3;
IF nfq208 = 3 THEN item08 = 2;
IF nfq208 = 4 THEN item08 = 1;
```

```
** ITEM 9;
IF nfq209 = 0 THEN item09 = 0;
IF nfq209 = 1 THEN item09 = 4;
IF nfq209 = 2 THEN item09 = 3;
IF nfq209 = 3 THEN item09 = 2;
IF nfq209 = 4 THEN item09 = 1;

** ITEM 10;
IF nfq210 = 0 THEN item10 = 0;
IF nfq210 = 1 THEN item10 = 4;
IF nfq210 = 2 THEN item10 = 3;
IF nfq210 = 3 THEN item10 = 2;
IF nfq210 = 4 THEN item10 = 1;

** ITEM 11;
IF nfq211 = 0 THEN item11 = 0;
IF nfq211 = 1 THEN item11 = 4;
IF nfq211 = 2 THEN item11 = 3;
IF nfq211 = 3 THEN item11 = 2;
IF nfq211 = 4 THEN item11 = 1;

** ITEM 12;
IF nfq212 = 0 THEN item12 = 0;
IF nfq212 = 1 THEN item12 = 4;
IF nfq212 = 2 THEN item12 = 3;
IF nfq212 = 3 THEN item12 = 2;
IF nfq212 = 4 THEN item12 = 1;

** ITEM 13;
IF nfq213 = 0 THEN item13 = 0;
IF nfq213 = 1 THEN item13 = 4;
IF nfq213 = 2 THEN item13 = 3;
IF nfq213 = 3 THEN item13 = 2;
IF nfq213 = 4 THEN item13 = 1;

** ITEM 14;
IF nfq214 = 0 THEN item14 = 0;
IF nfq214 = 1 THEN item14 = 4;
IF nfq214 = 2 THEN item14 = 3;
IF nfq214 = 3 THEN item14 = 2;
IF nfq214 = 4 THEN item14 = 1;
```

```
** ITEM 15;
item15 = 0;
IF nfq215 = 1 THEN item15 = 4;
IF nfq215 = 2 THEN item15 = 3;
IF nfq215 = 3 THEN item15 = 2;
IF nfq215 = 4 THEN item15 = 1;

** ITEM 16;
IF nfq216 = 0 THEN item16 = 0;
IF nfq216 = 1 THEN item16 = 4;
IF nfq216 = 2 THEN item16 = 3;
IF nfq216 = 3 THEN item16 = 2;
IF nfq216 = 4 THEN item16 = 1;

** ITEM 17;
IF nfq217 = 0 THEN item17 = 0;
IF nfq217 = 1 THEN item17 = 4;
IF nfq217 = 2 THEN item17 = 3;
IF nfq217 = 3 THEN item17 = 2;
IF nfq217 = 4 THEN item17 = 1;

** ITEM 18;
IF nfq218 = 0 THEN item18 = 0;
IF nfq218 = 1 THEN item18 = 4;
IF nfq218 = 2 THEN item18 = 3;
IF nfq218 = 3 THEN item18 = 2;
IF nfq218 = 4 THEN item18 = 1;

** ITEM 19;
IF nfq219 = 0 THEN item19 = 0;
IF nfq219 = 1 THEN item19 = 4;
IF nfq219 = 2 THEN item19 = 3;
IF nfq219 = 3 THEN item19 = 2;
IF nfq219 = 4 THEN item19 = 1;

** ITEM 20;
IF nfq220 = 0 THEN item20 = 0;
IF nfq220 = 1 THEN item20 = 4;
IF nfq220 = 2 THEN item20 = 3;
IF nfq220 = 3 THEN item20 = 2;
IF nfq220 = 4 THEN item20 = 1;
```

```
** ITEM 21;
IF nfq221 = 0 THEN item21 = 0;
IF nfq221 = 1 THEN item21 = 4;
IF nfq221 = 2 THEN item21 = 3;
IF nfq221 = 3 THEN item21 = 2;
IF nfq221 = 4 THEN item21 = 1;

** ITEM 22;
item22 = 0;
IF nfq222 = 1 THEN item22 = 4;
IF nfq222 = 2 THEN item22 = 3;
IF nfq222 = 3 THEN item22 = 2;
IF nfq222 = 4 THEN item22 = 1;

** ITEM 23;
item23 = 0;
IF nfq223 = 1 THEN item23 = 4;
IF nfq223 = 2 THEN item23 = 3;
IF nfq223 = 3 THEN item23 = 2;
IF nfq223 = 4 THEN item23 = 1;

** ITEM 24;
item24 = 0;
IF nfq224 = 1 THEN item24 = 4;
IF nfq224 = 2 THEN item24 = 3;
IF nfq224 = 3 THEN item24 = 2;
IF nfq224 = 4 THEN item24 = 1;

** ITEM 25;
item25 = 0;
IF nfq225 = 1 THEN item25 = 4;
IF nfq225 = 2 THEN item25 = 3;
IF nfq225 = 3 THEN item25 = 2;
IF nfq225 = 4 THEN item25 = 1;

** Scale okay as is;
item26 = nfq226;

** Calculate GENERAL PRODUCTIVITY score;
sub = item01+item02+item03+item04+item08+item09+item10+item11;
```

```
** Determine missing items;
miss = 0;
IF nfq201 = 0 THEN miss = miss +1;
IF nfq202 = 0 THEN miss = miss +1;
IF nfq203 = 0 THEN miss = miss +1;
IF nfq204 = 0 THEN miss = miss +1;
IF nfq208 = 0 THEN miss = miss +1;
IF nfq209 = 0 THEN miss = miss +1;
IF nfq210 = 0 THEN miss = miss +1;
IF nfq211 = 0 THEN miss = miss +1;

denom = 8-miss;
genprod =  sub/denom;

** Calculate SOCIAL OUTCOME score;
sub = item12+item13;

** Determine missing items;
miss = 0;
IF nfq212 = 0 THEN miss = miss +1;
IF nfq213 = 0 THEN miss = miss +1;
denom = 2-miss;

IF denom = 0 THEN socialout = 0;
IF denom NE 0 THEN socialout = sub/denom;

** Calculate ACTIVITY LEVEL score;
sub = item05+item14+item15+item16+item22+item23+item24+item25+item26;

** Determine missing items;
miss = 0;
IF nfq205 = 0 THEN miss = miss +1;
IF nfq214 = 0 THEN miss = miss +1;
IF nfq215 = 0 THEN miss = miss +1;
IF nfq216 = 0 THEN miss = miss +1;
IF nfq222 = 0 THEN miss = miss +1;
IF nfq223 = 0 THEN miss = miss +1;
IF nfq224 = 0 THEN miss = miss +1;
IF nfq225 = 0 THEN miss = miss +1;
IF nfq226 = 0 THEN miss = miss +1;

denom = 9-miss;
actlevel = sub/denom;

** Calculate VIGILANCE score;
sub = item06+item07+item17+item18+item19+item20+item21;
```

```
** Determine missing items;
miss = 0;
IF nfq206 = 0 THEN miss = miss +1;
IF nfq207 = 0 THEN miss = miss +1;
IF nfq217 = 0 THEN miss = miss +1;
IF nfq218 = 0 THEN miss = miss +1;
IF nfq219 = 0 THEN miss = miss +1;
IF nfq220 = 0 THEN miss = miss +1;
IF nfq221 = 0 THEN miss = miss +1;

denom = 7-miss;
IF denom = 0 THEN vigil = 0;
IF denom NE 0 THEN vigil = sub/denom;

** Calculate TOTAL score;
fosq = genprod+socialout+actlevel+vigil;
```

CALCULATIONS FOR THE SHORT SLEEP APNEA QUALITY OF LIFE INDEX (SAQLI) SCORE

```
** Transform from 0-6 scale to 1-7 scale;
** ITEM 1;
IF nsq201 = 0 THEN item01 = 7;
IF nsq201 = 1 THEN item01 = 6;
IF nsq201 = 2 THEN item01 = 5;
IF nsq201 = 3 THEN item01 = 4;
IF nsq201 = 4 THEN item01 = 3;
IF nsq201 = 5 THEN item01 = 2;
IF nsq201 = 6 THEN item01 = 1;

** ITEM 2;
IF nsq202 = 0 THEN item02 = 7;
IF nsq202 = 1 THEN item02 = 6;
IF nsq202 = 2 THEN item02 = 5;
IF nsq202 = 3 THEN item02 = 4;
IF nsq202 = 4 THEN item02 = 3;
IF nsq202 = 5 THEN item02 = 2;
IF nsq202 = 6 THEN item02 = 1;

** ITEM 3;
IF nsq203 = 0 THEN item03 = 7;
IF nsq203 = 1 THEN item03 = 6;
IF nsq203 = 2 THEN item03 = 5;
IF nsq203 = 3 THEN item03 = 4;
IF nsq203 = 4 THEN item03 = 3;
IF nsq203 = 5 THEN item03 = 2;
IF nsq203 = 6 THEN item03 = 1;
```

```
** ITEM 4;
IF nsq204 = 0 THEN item04 = 7;
IF nsq204 = 1 THEN item04 = 6;
IF nsq204 = 2 THEN item04 = 5;
IF nsq204 = 3 THEN item04 = 4;
IF nsq204 = 4 THEN item04 = 3;
IF nsq204 = 5 THEN item04 = 2;
IF nsq204 = 6 THEN item04 = 1;

** ITEM 5;
IF nsq205 = 0 THEN item05 = 7;
IF nsq205 = 1 THEN item05 = 7;
IF nsq205 = 2 THEN item05 = 6;
IF nsq205 = 3 THEN item05 = 5;
IF nsq205 = 4 THEN item05 = 4;
IF nsq205 = 5 THEN item05 = 3;
IF nsq205 = 6 THEN item05 = 2;
IF nsq205 = 7 THEN item05 = 1;

** ITEM 6;
IF nsq206 = 0 THEN item06 = 7;
IF nsq206 = 1 THEN item06 = 6;
IF nsq206 = 2 THEN item06 = 5;
IF nsq206 = 3 THEN item06 = 4;
IF nsq206 = 4 THEN item06 = 3;
IF nsq206 = 5 THEN item06 = 2;
IF nsq206 = 6 THEN item06 = 1;

** ITEM 7;
IF nsq207 = 0 THEN item07 = 7;
IF nsq207 = 1 THEN item07 = 6;
IF nsq207 = 2 THEN item07 = 5;
IF nsq207 = 3 THEN item07 = 4;
IF nsq207 = 4 THEN item07 = 3;
IF nsq207 = 5 THEN item07 = 2;
IF nsq207 = 6 THEN item07 = 1;

** ITEM 8;
IF nsq208 = 0 THEN item08 = 7;
IF nsq208 = 1 THEN item08 = 6;
IF nsq208 = 2 THEN item08 = 5;
IF nsq208 = 3 THEN item08 = 4;
IF nsq208 = 4 THEN item08 = 3;
IF nsq208 = 5 THEN item08 = 2;
IF nsq208 = 6 THEN item08 = 1;
```

```
** ITEM 9;
IF nsq209 = 0 THEN item09 = 7;
IF nsq209 = 1 THEN item09 = 6;
IF nsq209 = 2 THEN item09 = 5;
IF nsq209 = 3 THEN item09 = 4;
IF nsq209 = 4 THEN item09 = 3;
IF nsq209 = 5 THEN item09 = 2;
IF nsq209 = 6 THEN item09 = 1;

** ITEM 10;
IF nsq210 = 0 THEN item10 = 7;
IF nsq210 = 1 THEN item10 = 6;
IF nsq210 = 2 THEN item10 = 5;
IF nsq210 = 3 THEN item10 = 4;
IF nsq210 = 4 THEN item10 = 3;
IF nsq210 = 5 THEN item10 = 2;
IF nsq210 = 6 THEN item10 = 1;

** ITEM 11;
IF nsq211 = 0 THEN item11 = 7;
IF nsq211 = 1 THEN item11 = 6;
IF nsq211 = 2 THEN item11 = 5;
IF nsq211 = 3 THEN item11 = 4;
IF nsq211 = 4 THEN item11 = 3;
IF nsq211 = 5 THEN item11 = 2;
IF nsq211 = 6 THEN item11 = 1;

** ITEM 12;
IF nsq212 = 0 THEN item12 = 7;
IF nsq212 = 1 THEN item12 = 6;
IF nsq212 = 2 THEN item12 = 5;
IF nsq212 = 3 THEN item12 = 4;
IF nsq212 = 4 THEN item12 = 3;
IF nsq212 = 5 THEN item12 = 2;
IF nsq212 = 6 THEN item12 = 1;

** ITEM 13;
IF nsq213 = 0 THEN item13 = 7;
IF nsq213 = 1 THEN item13 = 6;
IF nsq213 = 2 THEN item13 = 5;
IF nsq213 = 3 THEN item13 = 4;
IF nsq213 = 4 THEN item13 = 3;
IF nsq213 = 5 THEN item13 = 2;
IF nsq213 = 6 THEN item13 = 1;
```

6.4.1. Derived non-PSG SHHS 2 variables (SAS)

```
** ITEM 14;
IF nsq214 = 0 THEN item14 = 7;
IF nsq214 = 1 THEN item14 = 6;
IF nsq214 = 2 THEN item14 = 5;
IF nsq214 = 3 THEN item14 = 4;
IF nsq214 = 4 THEN item14 = 3;
IF nsq214 = 5 THEN item14 = 2;
IF nsq214 = 6 THEN item14 = 1;

** Calculate DOMAIN A - DAILY ACTIVITIES score;
sub =
item01+item02+item03+item04+item05+item06+item07+item08+item09+item10+item11+item12+item13+item
14;
saqli = sub/14;
```

7. SHHS 2 ECG

7.1. Characteristics of SHHS 2 ECG database

Name of dataset:	*SHHS2ECG_03mar2011.sas7bdat*
Number of Records:	3876 participants
Primary Key Field:	PPTID - Participant Identifier (alpha - numeric)
File Formats:	All files are SAS 9.1 Files.
Demographic variables:	None

7.2. Description of data

ECGs were performed as part of the PSG. The modem-transmitted ECGs were interpreted by a centralized ECG Reading Center (ECGRC). The ECGRC was also responsible for assisting in the development of a protocol for ECG performance and reading, developing standards for, and reporting on quality of ECGs so that performance could be determined by Field Site and technician.

7.3. Dataset processing specifics

```
/*********************************************
Finalize Okin's DATASET and save
*********************************************/;
** Encoding the original MNcode into mc indicators that didn't exist;
DATA ecg.finalokin;
    SET ecg.ecg;
    pptid = UPCASE(pptid);
    BY pptid;
    IF (FIRST.pptid AND LAST.pptid) ;          /* Delete all duplicate pptids in original Okin's, N=52*/
    IF pptid = "302432" THEN DELETE    ;       /* No matching pptid in SHHS 2, N = 1; The final
                                               DATASET sample size N=3976 (3992 - 52 - 1); Sub
                                               -tracting original Okin's MNcode */
    ARRAY code{*} $ code1-code10;
    ARRAY tmpmn{*} $tmpmncode1-tmpmncode10;
    ARRAY mn{*} $mncode1-mncode10;
    DO i = 1 to DIM(code);
        code(i)  = SCAN(mncode,i,",");
        tmpmn(i)  = COMPRESS(SCAN(code(i), 1, ".")) || SCAN(code(i), 2, ".") || SCAN(code(i), 3, ".");
        mn(i)  = SUBSTR(tmpmn(i), 1, 3);
    END;

    /* Generating new mc variables(indicators for specific Minnesota Code)*/;
    ARRAY mc{*} mc1_1_1-mc1_1_7 mc1_2_1-mc1_2_8 mc1_3_1-mc1_3_3 mc6_2_1-mc6_2_3;
    ARRAY mccode{21} _temporary_
        (111,112,113,114,115,116,117,121,122,123,124,125,126,127,128,131,132,133,621,622,623);
    DO j = 1 to DIM(mc);
        mc(j) = (mncode1 = mccode(j) | mncode2 = mccode(j) | mncode3 = mccode(j) | mncode4 = mccode(j) |
        mncode5 = mccode(j) | mncode6 = mccode(j) | mncode7 = mccode(j) | mncode8 = mccode(j) |
        mncode9 = mccode(j) | mncode10 = mccode(j));
    END;

    DROP FieldSiteID AlphaCode DateOfForm VisitID FormRevision DateOfECG Narrative MinnesotaA
    MinnesotaB MinnesotaC MinnesotaD MinnesotaE MinnesotaF MinnesotaG MinnesotaH Note Interviewer
    Date formdate subSET_name filepath ID scode1 scode2 handcode  tmpmncode1-tmpmncode10
    mncode1 - mncode10 i j ;
RUN;

/*************************************************
**              Encoding the variables as SHHS1          **
*************************************************/;
DATA shhs2ecg;
    SET ecg.finalokin;
    site = clinic;
    lvh3_1 = mc3_1;
    lvh3_3 = mc3_3;
    st4_1_3 = (mc4_1 | mc4_2 | mc4_3);
```

```
st5_1_3 = (mc5_1 | mc5_2 | mc5_3);
lvhst = (mc3_1 | mc3_3 ) & ( mc4_1 | mc4_2 | mc4_3 | mc5_1 | mc5_2 | mc5_3);
ventrate = vrate;
qrs = qrsax;
mob1 = mc6_2_3;
part2deg = mc6_2_2;
mob2 = mc6_2_1;
av3deg = mc6_1;
afib = (mc8_3_1 OR mc8_3_2);
nodal = mc8_4_1;
paced = mc6_8;
av1deg = mc6_3;
lbbb = mc7_1;
rbbb = mc7_2;
ilbbb = mc7_6;
irbbb = mc7_3;
lah = mc7_7;
iventblk = mc7_4;
wpw = mc6_4; *wpw: MNcode = 6_4_1 but only 6_4 in shhs2;
apbs = mc8_1_1;
vpbs = mc8_1_2;

/* Anteroseptal MI */;
IF (mc1_1_6 OR mc1_1_7 OR mc1_2_7) THEN antsepmi = 1;
ELSE IF (mc1_2_8 OR mc1_3_2) THEN antsepmi = 2;
ELSE antsepmi = 0;

/* Inferior MI */;
IF (mc1_1_4 OR mc1_1_5 OR mc1_2_4 OR mc1_2_5) THEN infmi = 1;
ELSE IF (mc1_2_6 OR mc1_3) THEN infmi = 2;
ELSE infmi = 0;

/* Missing for True posterior MI, no Minnesoata code available;*/

/* Anterolateral MI */;
IF (mc1_1_1 OR mc1_1_2 OR mc1_1_3 OR mc1_2_1 OR mc1_2_2 OR mc1_2_3) THEN antlatmi = 1;
ELSE IF (mc1_3_1 OR mc1_3_3) THEN antlatmi = 2;
ELSE antlatmi = 0;

/* NONSP_ST */;
IF (mc4_1 OR mc4_2 OR mc4_3) THEN nonsp_st = 1;    /* ST depression */
ELSE IF mc4_3 THEN nonsp_st = 2;                   /* ST flattening */
ELSE nonsp_st = 0;
```

```
    /* NONSP_TW */;
    IF (mc5_1 OR mc5_) THEN nonsp_tw = 1;          /* T wave inversion */
    ELSE IF mc5_3 THEN nonsp_tw = 2;               /* T wave flattening */
    ELSE nonsp_tw = 0;

    rtrial = mc9_3;
    rvh = mc3_2;
RUN;

/*****************************************************************;
**   Merge the psgdate and ecgdate from SHHS 2 and EC4 DATASETS   **;
*****************************************************************/;
DATA ecg.shhs2ec4_ecgdate;
    SET ecg.Shhs2_ec4;
    pptid = UPCASE(pptid);
    KEEP pptid formdate;
RUN;

PROC SORT DATA = ecg.shhs2ec4_ecgdate; BY pptid; RUN;
PROC SORT DATA = ecg.shhs2final_psgdate; BY pptid; RUN;
PROC SORT DATA = shhs2ecg; BY pptid; RUN;

DATA ecg.ecg_SHHS2_final_2007_01_25;
    MERGE shhs2ecg (IN = A) ecg.shhs2ec4_ecgdate (IN = B) ecg.shhs2final_psgdate (IN = C);
    BY pptid;
    IF A = 1;

    /* Date variables */ ;
    FORMAT psgdate ecgdate DATE7.;
    psgdate = stdatep;
    ecgdate = formdate;
    LABEL psgdate = "PSG Date";

    KEEP site pptid  psgdate ecgdate lvh3_1 lvh3_3 st4_1_3 st5_1_3 lvhst ventrate qrs mob1 part2deg mob2
    av3deg afib nodal paced av1deg lbbb rbbb ilbbb irbbb lah iventblk wpw apbs vpbs antsepmi infmi antlatmi
    nonsp_st nonsp_tw rtrial rvh;
RUN;
```

8. SHHS CVD Outcomes

8.1. Characteristics of SHHS 2 CVD outcomes

8.1.1 CVD outcomes status dataset

Name of dataset:	*SHHS_Status_03mar2011.sas7bdat*
Number of Records:	5679 participants
Primary Key Field:	PPTID - Participant Identifier (alpha - numeric)
File Formats:	All files are SAS 9.1 Files.
Demographic variables:	None

8.1.2 CVD outcomes events dataset

Name of dataset:	*SHHS_Event_03mar2011.sas7bdat*
Number of Records:	3385 observations
Primary Key Field:	PPTID - Participant Identifier (alpha - numeric)
File Formats:	All files are SAS 9.1 Files.
Demographic variables:	None

8.2. Description of data

SHHS was designed to include subjects participating in multiple existing cohort studies and to take advantage of ongoing mechanisms of CVD outcome ascertainment and adjudication in place in these parent cohorts. Specifically, the ARIC, CHS, FHS, and SHS studies have had mechanisms in place for determining CVD outcomes since the start of SHHS. Outcomes data for SHHS subjects belonging to these cohorts have been provided to SHHS by the parent cohorts. The SHHS subjects recruited in Tucson and New York were members of research cohorts that did not include ongoing assessment of CVD outcomes. In these two sites, SHHS investigators have implemented their own procedures for ascertaining and adjudicating CVD outcomes among SHHS participants. These procedures have been closely modeled on those of CHS.

Key outcomes for SHHS include the following incident or recurrent CVD events or diagnoses occurring subsequent to the first SHHS PSG:

a. Hospitalized acute MI (HAMI)
b. Coronary surgical intervention -- percutaneous transcutaneous angioplasty (PTCA), coronary stent placement, coronary artery bypass grafting (CABG)
c. Congestive Heart Failure (CHF)
d. Coronary heart disease death
e. Any coronary heart disease (CHD) -- summary variable which includes a, b, and d above.
f. Any cardiovascular disease (CVD) -- summary variable which includes a - d above.
g. Angina pectoris (AP) -- at CHS and FHS only

The following recurrent events will be considered endpoints for the SHHS:

a. HAMI
b. Coronary surgical intervention
c. Stroke

8.3. Dataset processing specifics

See Appendix 9.9 for data processing specifics:

9. Appendix

9.1. SHHS A-variables

Variable Name	Variable Description	Variable Codes
Parent Study	Parent Study	1=CHS; 2=ARIC; 3=Strong Heart; 4=Framingham; 5=New York; 6=TUC
Site	Field center/site ID	11=FRA; 21=HAG; 22=HAG; 31=MIN; 41=NY; 51=PIT; 52=SAC; 61=SD; 62=OK; 63=PHX; 64=TUC
Race	Race	1=White; 2=Black; 3=Native Amer/Alaskan; 4=Asian/Pacific Islander; 5=Hispanic/Mex Amer; 6=Other
Gender	Gender	M=Male; F=Female
DOB	Date of Birth	(date)
Educat	Education level; number of years	(GED=12 yrs in tech school counted like years of college)
EducatDT	Education level date	(date)
Mstat	Marital Status	1=Married; 2=Widowed; 3=Divorced/Separated; 4=Never married; 8=Unknown/Refused
MStatDT	Marital Status Date	(date)
SRHype	Self-Reported Hypertension	0=N; 1=yes; 9=Unknown
SRHypeDT	Self-Reported Hypertension Date	(date)
PrtRptDiab	Self-Reported Diabetes	0=No; 1=yes, 9=Unknown
SRDiabDT	Self-Reported Diabetes Date	(date)
CgPkTr	Lifetime cigarette smoking; pack years	(number)
CgPkYrDt	Lifetime cigarette smoking date	(date)
Alcoh	Usual Alcohol Intake: Number of drinks per day	(number)

9.1. SHHS A-variables

Variable Name	Variable Description	Variable Codes
AlcohDT	Usual Alcohol Intake Date	(date)
SystBP	Seated Systolic Blood Pressure	(mm Hg)
DiasBP	Seated Diastolic Blood Pressure	(mm Hg)
SysDiaDT	Systolic/Diastolic Blood Pressure Date	(date)
Height	Height in centimeters	(centimeters)
HeightDT	Height Date	(date)
Weight	Weight in kilograms	(kilograms)
WeightDT	Weight Date	(date)
Waist	Waist measurements in centimeters	(centimeters)
WaistDT	Waist Date	(date)
Hip	Hip measurement in centimeters	(centimeters)
HipDT	Hip date	(date)
Chol	Cholesterol mg/dl	(mg/dl)
HDL	HDL cholesterol (mg/dl)	(mg/dl)
Trig	Triglycerides mg/dl	(mg/dl)
CholDT	Cholesterol/HDL/Triglycerides Date	(date)
FEV1	Forced Expiratory Volume in One Second (Liters)	(liters)
FVC	Forced Vital Capacity (Liters)	(liters)
FEVFVCDT	FEV1 FVC Date	(date)
AAI	Ankle Arm Index	(ratio: ankle to arm SYS BP)
AnkBP	Supine Ankle Blood Pressure	(mm Hg)

9.1. SHHS A-variables

Variable Name	Variable Description	Variable Codes
ArmBP	Supine Arm Blood Pressure	(mm Hg)
AnkArmDT	Supine Ankle/Arm Date	(date)
ECG	ECG Data	0=No; 1=Yes
ECGdt	Date of ECG	(date)

Special Notes:

From the SHHS 2 protocol: SHHS is designed to use existing data collected by the parent studies regarding health history, cardiovascular risk factors, and cardiovascular events. At the study's onset, the Comparability Committee was charged with comparing data collected by the various parent studies to determine the data to be used.

(A) Variables (key risk-factors for cardiovascular disease and outcomes) that are considered critical for the study; if any of the cohorts do not have comparable data in any of these variables, additional data are to be collected.

Note: For each of the A-variables, a maximum acceptable time window between the time of the home PSG and the closest measurement will be specified. That is, data previously collected by the parent study could be used for SHHS as long as they were collected within an acceptable time window. A-variables collected outside the acceptable time window must be re-ascertained for SHHS. For cohort members refusing or ineligible for a second PSG, the reference date will be the date of the home visit. In absence of a home visit, the observation closet in time to the screening interview will be used.

Note: As the SHHS is now in a phase of longitudinal data collection, the A-Variables also need to be considered in a time-dependent fashion. A number of the A-variables might change over time; diabetes status, lipid levels, alcohol intake, and smoking. SHHS will track self-report of diabetes and smoking.

(B) Variables that could be important in specific or subset analysis; an attempt to achieve comparability will be made, but it is not required that all cohorts have comparable information.

(C) Other variables that could be used in cohort-specific analyses, or in ancillary studies, but no specific attempt to achieve comparability will be made.

9.2. SHHS 1 and 2 variables for tobacco, caffeine and alcohol

TABLE 1: SHHS 1 variables related to Tobacco, Caffeine, Alcohol collected by Health Interview (HIN, HIF, and HIA) forms.

SHHS1		**SHHS 1 Variables**		
		Health Information New York (HIN)	Health Information Framingham (HIF)	Health Information ARIC, CHS, Tucson, Strong Heart (HIA)
TOBACCO		EVSMOK15	EVSMOK15	EVSMOK15
		AGSMK15		
		NS1YR15		
		YRSNS15		
		SMKNOW15	SMKNOW15	SMKNOW15
		CIGDAY15	CIGDAY15	CIGDAY15
		YRSTOP15	YRSTOP15	YRSTOP15
		AVESMK15		
CAFFEINE		COFFEE15	COFFEE15	COFFEE15
		TEA15	TEA15	TEA15
		SODA15	SODA15	SODA15
ALCOHOL		WINE15		
		BEER15		
		SHOTS15		

9.2. SHHS 1 and 2 variables for tobacco, caffeine and alcohol

TABLE 2: SHHS 1 and SHHS 2 variables related to Tobacco, Caffeine, and Alcohol collected on Morning Survey (MS) forms.

SHHS 1 & SHHS 2 Morning Survey			**SHHS 1 SHHS 2 Variables**	
			SHHS 1 Morning Survey (MS)	SHHS 2 Morning Survey (MS)
ALCOHOL			WINE10	MS209a
			SHOTS10	MS209b
			BEER10	MS209c
CAFFEINE			COFFEE10	MS210a
			SODA10	MS210b
			TEA10	MS210c
TOBACCO			CGRTTS10	MS211a
			PIPE10	MS211b
			CIGARS10	MS211c

9.2. SHHS 1 and 2 variables for tobacco, caffeine and alcohol

TABLE 3: SHHS 2 variables related to Tobacco, Caffeine, and Alcohol collected on Sleep Habits Questionnaire form. (Variables from SHHS 1 Health interview the align)

SHHS 2 Sleep Habits Questionnaire			SHHS 2 variables / (SHHS 1 Health interview)
D. SMOKING			SH320 (EVSMOK15)
			SH321
			SH322
			SH323 (SMKNOW15)
			SH323a
			SH324
E. BEVERAGES - Caffeine			SH325
			SH326
			SH327a
			SH327b
F. BEVERAGES – Alcohol			SH328 (WINE15)
			SH329 (BEER15)
			SH330 (SHOTS15)

9.3. SHHS 1 medication categories

Variable Name	Description
A2A	Angiotensin Type 2 Antagonists
A2AD	Combinations of Angiotensin II Antagonists Plus Diuretics
ACE	ACE inhibitors without diuretics
ACED	ACE inhibitors with diuretics
ADPI	Inhibitors of ADP-induced platelet aggregation
AGDI	Alpha-glucosidase inhibitors
ALPHA	Alpha-blockers without diuretics
ALPHAD	Alpha-blockers with diuretics
ALZH	Acetylcholine esterase inhibitors for Alzh's
AMLOD	Amlodipine
ANAR1A	Anti-arrhythmics, class 1A
ANAR1B	Anti-arrhythmics, class 1B
ANAR1C	Anti-arrhythmics, class 1C
ANAR3	Anti-arrhythmics, class 3
APSY	Anti-psychotic medications
ASA	Aspirin from 280804 (anti-inflam agents)
BASQ	Bile-acid sequestrants
BENZOD	Benzodiazepines
BETA	Beta-blockers without diuretics
BETAD	Beta-blockers with diuretics
BGND	Biguanides
CCB	Any calcium-channel blocker = CCIR or CCBSR or CCBT
CCBIR	Immediate-release CCBS = NFIR or DIHIR or VERIR or DLTIR
CCBSR	Slow-release CCBS = NIFSR or DIHSR or VERSR or DLTSR or AMLOD
CCBT	T-type calcium-channel blocker
COX2	Cox-2 inhibitors (NSAID agents); separate from NSAID variable
DIG	Digitalis preparations
DIHIR	Immediate-release dihyroprydines other than nifedipine
DIHSR	Slow-release dihydropyridines other than nifedipine or amlodipine
DLTIR	Immediate-release diltiazem
DLTSR	Slow-release diltiazem

9.3. SHHS 1 medication categories

Variable Name	Description
EDD	Erectile dysfunction drugs
ESTRGN	Estrogens, excluding vaginal creams
FIBR	Fibrates
H2B	H2 blockers
HCTZ	Thiazide diuretics without k-sparing agents
HCTZK	Thiazide diuretics with k-sparing agents
HPRNS	Heparins
INSLN	Insulins
ITPRTR (IPRTR)	Anticholinergics + combination with beta2-agonist
ISTRD	Inhaled steroids for asthma
KBLKR	K-channel blockers to enhance insulin secretion for diabetics
KCL	Potassium supplements
KSPR	Potassium-sparing agents alone
LOOP	Loop diuretics
MAOI	MAO inhibitors
MLPD	Miscellaneous lipid-lowering drugs
NIAC	Niacin and nicotinic acid
NIFIR	Immediate-release nifedipine
NIFSR	Slow-release nifedipine
NSAID	Non-steroidal anti-inflammatory agents, excluding aspirin
NTCA	Non-tricyclic antidepressants other than MAOI
NTG	Nitrates
OAIA	Oral anti-inflammatory asthma drugs (leukotriene receptor antagonists and inhibitors of lipo-oxygenase)
OHGA	Oral hypoglycemic agents
OSTRD	Oral steroids
OTCH2B	Over-the-counter H2-blockers
PDEI	Phosphodiesterase inhibitors
PPI	Proton pump inhibitors
PRGSTN	Progestins
PRKNSN	Drugs to treat Parkinson's
PRMRN	Premarin (conjugated estrogens)

9.3. SHHS 1 medication categories

Variable Name	Description
PROB	Probucol
PVDL	Peripheral vasodilators, exclude dipyridamole
SLF1	First generation sulfonylureas
SLF2	Second generation sulfonylureas
STTN	HMG COA reducatace inhibitors (statins)
SYMPTH	Sympathomimetics, oral and inhaled
TCA	Tri-cyclic antidepressants
TCAP	Tri-cyclic antidepressants plus anti-psychotics combinations
THRY	Thyroid agents
THZD	Thiazolidinediones
URCOS	Uricosurics
VASO	Vasodilators, a mixed group
VASOD	Vasodilators with diuretics, a mixed group
VERIR	Immediate-release verapamil
VERSR	Slow-release verapamil
WARF	Oral anticoagulants including warfarin, coumadin, and anisindione
WTLS	Weight loss drugs
XOI	Xanthine oxidase inhibitors
DIUR	Any diuretic
LIPID	Any lipid-lowering medication
HTNMED	Any antihypertensive medication

9.4. SHHS 1 form indicator variables

SHHS 1 Form	Indicator variable
Adverse Events Form	shhs1_ae
Blood Pressure Form	shhs1_bp
Contact Form	shhs1_cf
Health Interview Form	shhs1_hi
Sleep Habits Form	shhs1_sh
Morning Survey Form	shhs1_ms
Quality Assurance Form	shhs1_qc
Sleep Data Retrieval Form	shhs1_sdr
Quality of Life Form	shhs1_ql
Categorized Medications Form	shhs1_meds
PSG variables Form	shhs1_psg
A-Variables Form	shhs1_avars
Tuscon CVD Form	shhs1_tcvd
All SHHS1 Forms present	shhs1_all

Indicator variable = "1" if the form is available for the participant and = "0" if the form is not available

9.5. Followup 1 indicator variables

Follow up 1	Indicator variable
Contact Form	fup_scr
Status, BP and Weight Form	fup_stat
Sleep Habits Questionnaire	fup_slpsym
All FUP forms present	fup_all

Indicator variable = "1" if the form is available for the participant and = "0" if the form is not available

9.6. SHHS 2 medication categories

Variable	Variable Description
A2A2	Angiotensin type 2 antagonists
A2AD2	Combinations of angiotensin II antagonists plus diuretics
ACE2	ACE Inhibitors without diuretics
ACED2	ACE Inhibitors with diuretics
ADPI2	Inhibitors of ADP-induced platelet aggregation
AGDI2	alpha-Glucosidase inhibitors
ALPHA2	Alpha-blockers without diuretics
ALPHAD	Alpha-blockers with diuretics
ALZH2	Acetylcholine esterase inhibitors for Alzh's
AMLOD2	Amlodipine
ANAR1A2	Anti-arrhythmics, class 1A
ANAR1B2	Anti-arrhythmics, class 1B
ANAR22	Anti-arrhythmics, class 2
ANAR32	Anti-arrhythmics, class 3
APSY2	Anti-psychotic medications
ASA2	Aspirin from 280804 (anti-inflam agents)
BASQ2	Bile-acid sequestrants
BENZOD2	Benzodiazepines
BETA2	Beta-blockers without diuretics
BETAD2	Beta-blockers with diuretics
BGND2	Biguanides
CCB2	Any calcium-channel blocker = CCIR or CCBSR or CCBT
CCBIR2	Immediate-release CCBs = NIFIR or DIHIR or VERIR or DLTIR
CCBSR2	Slow-release CCBs = NIFSR or DIHSR or VERSR or DLTSR or AMLOD
CCBT2	t-type calcium-channel blocker

9.6. SHHS 2 medication categories

Variable	Variable Description
COX22	Cox-2 inhibitors (NSAID agents); separate from NSAID variable
DIG2	Digitalis preparations
DIHIR2	Immediate-release dihyropryidines other than nifedipine
DIHSR2	Slow-release dihydropyridines other than nifedipine or amlopidine
DIUR2	Any diuretic
DLTIR2	Immediate-release diltiazem
DLTSR2	Slow-release diltiazem
EDD2	Erectile dysfunction drugs
ESTRGN2	Estrogens, excluding vaginal creams
FIBR2	Fibrates
H2B2	H-2 Blockers
HCTZ2	Thiazide diuretics without K-sparing agents
HCTZK2	Thiazide diuretics with k-sparing agents
HPRNS2	Heparins
HTNMED2	Any anti-hypertensive medication
INSLN2	Insulins
IPRTR2	Anticholinergics + combination with beta2-agonist
ISTRD2	Inhaled steroids for asthma
KBLKR2	K-channel blockers to enhance insulin secretion for diabetics
KCL2	Potassium supplements
KSPR2	Potassium-sparing agents alone
LIPID2	Any lipid-lowering medication
LOOP2	Loop diuretics
MAOI2	MAO Inhibitors
MLPD2	Miscellaneous lipid-lowering drugs

Variable	Variable Description
NIAC2	Niacin and nicotinic acid
NIFIR2 NIFSR2	Immediate-release nifedipine Slow-release nifedipine
NSAID2	Non-steroidal anti-inflammatory agents, excluding aspirin
NTCA2	Non-tricyclic antidepressants other than MAOI
NTG2	Nitrates
OAIA2	Oral Anti-Inflammatory Asthma drugs (leukotriene receptor antagonists and inhibitors of lipo-oxygenase)
OHGA2	Oral hypoglycemic agents
OSTRD2	Oral steroids
OTCH2B2	Over-the-counter H-2 blockers
PDEI2	Phosphodiesterase inhibitors
PPI2	proton pump inhibitors
PRGSTN2	Progestins
PRKNSN2	drugs used to treat Parkinson's
PRMRN2	Premarin (conjugated estrogens)
PROB2	Probucol
PVDL2	Peripheral vasodilators, exclude dipyridamole
SLF12	First generation sulfonylureas
SLF22	Second generation sulfonylureas
STTN2	HMG CoA reducatace inhibitors (statins)
SYMPTH2	Sympathomimetics, oral and inhaled
TCA2	Tricylic anti-depressants
TCAP2	Tri-cyclic anti-depressants plus anti-psychotics combinations
THRY2	Thyroid agents
THZD2	Thiazolidinediones

9.6. SHHS 2 medication categories

Variable	Variable Description
URCOS2	uricosurics
VASO2	Vasodilators, a mixed group
VASOD2	Vasodilators with diuretics, a mixed group
VERIR2	Immediate-release verapamil
VERSR2	Slow-release verapamil
WARF2	Oral anticoagulants including warfarin, coumadin, and anisindione
WTLS2	weight loss drugs
XOI2	xanthine oxidase inhibitors

9.7. SHHS 2 indicator variables

SHHS 2 Form	Indicator variable
Health Interview Form	shhs2_hi
Sleep Habits Form	shhs2sh
Morning Survey Form	shhs2_ms
Physical Measurements Form	shhs2_pm
Quality of Life Form	shhs2_ql
Categorized Medications Form	shhs2_meds
PSG variables Form	shhs2_psg
All SHHS2 Forms present	shhs2_all

Indicator variable = "1" if the form is available for the participant and = "0" if the form is not available

9.8. SHHS 1 / SHHS 2 ECG variables

Variable Name	Variable Description	Variable Codes
site	Field site ID	11=Framingham; 21=Hagerstown; 22=Hagerstown; 31=Minneapolis; 41=NY; 51=Pittsburgh; 52=Sacramento; 61=South Dakota; 62=Oklahoma; 63=Phoenix; 64=Tuscon
pptid	Participant ID	[number]
psgdate	Date of PSG study	[date]
ecgdate	Date of ECG	[date]
lvh3_1	Left ventricular hypertrophy (LVH) Voltage	0=No, 1=Yes
lvh3_3	Left ventricular hypertrophy (LVH) Voltage	0=No, 1=Yes
st4_1_3	ST junction/ST segment depression and T-wave	0=No, 1=Yes
st5_1_3	ST junction/ST segment depression and T-wave	0=No, 1=Yes
lvhst	Left ventricular hypertrophy (LVH) with ST,T-wave abnormality	0=No, 1=Yes
mob1	Mobitz I	0=No, 1=Yes
part2deg	Partial 2nd degree	0=No, 1=Yes
mob2	Mobitz II	0=No, 1=Yes
av3deg	3rd degree AV block	0=No, 1=Yes
av1deg	1st degree AV block	0=No, 1=Yes

9.8. SHHS 1 / SHHS 2 ECG variables

Variable Name	Variable Description	Variable Codes
lbbb	Left bundle branch block	0=No, 1=Yes
rbbb	Right bundle branch block	0=No, 1=Yes
ilbbb	Incomplete left bundle branch block	0=No, 1=Yes
irbbb	Incomplete right bundle branch block	0=No, 1=Yes
lah	Left anterior hemiblock	0=No, 1=Yes
iventblk	Intraventricular block pattern indeterminate	0=No, 1=Yes
wpw	Wolff-Parkinson-White syndrome	0=No, 1=Yes
antsepmi	Anteroseptal MI	0=No; 1=(mc1_1_6 or mc1_1_7 or mc1_2_7); 2=(c1_2_8 or mc1_3_2)
infmi	Inferior MI	0=No; 1=(mc1_1_4 or mc1_1_5 or mc1_2_4 or mc1_2_5); 2=(mc1_2_6 or mc1_3)
antlatmi	Anterolateral MI	0=No; 1=(mc1_1_1 or mc1_1_2 or mc1_1_3 or mc1_2_1 or mc1_2_2 or mc1_2_3); 2=(mc1_3_1 or mc1_3_3)
nonsp_st	Nonspecific ST abnormality	0=No; 1=(mc4_1 or mc4_2 or mc4_3); 2=(mc4_3)
nonsp_tw	Nonspecific T-wave abnormality	0=No; 1=(mc5_1 or mc5_2); 2=(mc5_3)
rtrial	Right trial enlargement	0=No, 1=Yes
rvh	Right ventricular hypertrophy	0=No, 1=Yes
ventrate	Ventricular rate	[number]
qrs	QRS axis	[number]
afib	A-fib/flutter	0=No, 1=Yes

9.8. SHHS 1 / SHHS 2 ECG variables

Variable Name	Variable Description	Variable Codes
paced	Paced	0=No, 1=Yes
nodal	Nodal rhythm	0=No, 1=Yes
apbs	Atrial premature beats	0=No, 1=Yes
vpbs	Ventricular premature beats	0=No, 1=Yes
truposmi	True posterior MI	0=No, 1=Yes

* See Analytic Data Dictionary for more detailed information regarding the ECG variables.

9.9. SHHS CVD outcome variables

9.9.1. Status variables

Variable Name	Variable Description	Variable Codes
pptid	Participant ID	[number]
clinic	Field site ID	11=Framingham; 21=Hagerstown; 22=Hagerstown; 31=Minneapolis; 41=NY; 51=Pittsburgh; 52=Sacramento; 61=South Dakota; 62=Oklahoma; 63=Phoenix; 64=Tuscon
blpsgdate	Baseline PSG date	[date]
vital	Vital status	0=Dead, 1=Alive
censdate	Date of last contact/death	[date]
prev_mi	MI prior to baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes
prev_mip	MI/procedure prior to baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes
prev_stk	Stroke prior to baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes
mi	Number of MI(s) since baseline PSG (as recorded in parent studies datasets)	[number] (0=No events)
mip	Number of MI/procedure event(s) since baseline PSG (as recorded in parent studies datasets)	[number] (0=No events)
mi_fatal	Fatal MI since baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes
mi_date	Date of first MI since baseline PSG (as recorded in parent studies datasets)	[date]
mip_date	Date of first MI/procedure since baseline PSG (as recorded in parent studies datasets)	[date]

9.9.1. Status variables

Variable Name	Variable Description	Variable Codes
stroke	Number of stroke(s) since baseline PSG (as recorded in parent studies datasets)	[number] (0=No events)
stk_fatal	Fatal stroke since baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes
stk_date	Date of first stroke since baseline PSG (as recorded in parent studies datasets)	[date]
stk_type	Type of stroke (as recorded in parent studies datasets)	[text]
chd_death	Coronary Heart Disease (CHD) death (as recorded in parent studies datasets)	0=No, 1=Yes
chd_dthdt	Date of CHD death (as recorded in parent studies datasets)	[date]
cvd_death	Cardiovascular Disease (CVD) death (as recorded in parent studies datasets)	0=No, 1=Yes
cvd_dthdt	Date of CVD death (as recorded in parent studies datasets)	[date]
prev_ang	Angina prior to baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes
angina	Number of angina event(s) since baseline PSG (as recorded in parent studies datasets)	[number] (0=No events)
ang_date	Date of angina since baseline PSG (as recorded in parent studies datasets)	[date]
prev_revpro	Revascularization procedure prior to baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes

9.9.1. Status variables

Variable Name	Variable Description	Variable Codes
revasc_proc	Number of revascularization procedure(s) since baseline PSG (as recorded in parent studies datasets)	[number] (0=No events)
revasc_fatal	Fatal revascularization procedure since baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes
revpro_date	Date of first revascularization procedure since baseline PSG (as recorded in parent studies datasets)	[date]
ptca	Number of PTCA event(s) since baseline PSG (as recorded in parent studies datasets)	[number] (0=No events)
ptca_date	Date of first PTCA since baseline PSG (as recorded in parent studies datasets)	[date]
cabg	Number of CABG event(s) since baseline PSG (as recorded in parent studies datasets)	[number] (0=No events)
cabg_fatal	Fatal CABG since baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes
cabg_date	Date of first CABG since baseline PSG (as recorded in parent studies datasets)	[date]
stent	Stent since baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes
stent_date	Date of first stent since baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes
mi_death	MI death since baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes

9.9.1. Status variables

Variable Name	Variable Description	Variable Codes
prev_chf	Congestive Heart Failure (CHF) prior to baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes
chf	Congestive Heart Failure (CHF) since baseline PSG (as recorded in parent studies datasets)	[number] (0=No events)
chf_date	Date of first CHF since baseline PSG (as recorded in parent studies datasets)	[date]
any_cvd	Any Coronary Heart Disease (CHD) since baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes
any_chd	Any Cardiovascular Disease (CVD) since baseline PSG (as recorded in parent studies datasets)	0=No, 1=Yes
mi15	Self-reported MI (as reported on SHHS 1 baseline health interviews)	0=No, 1=Yes
stroke15	Self-reported stroke (as reported on SHHS 1 baseline health interviews)	0=No, 1=Yes
cabg15	Self-reported CABG (as reported on SHHS 1 baseline health interviews)	0=No, 1=Yes
ca15	Self-reported coronary angioplasty (as reported on SHHS 1 baseline health interviews)	0=No, 1=Yes

*See Analytic Data Dictionary for more details

9.9.2. Event variables

Variable Name	Variable Description	Variable Codes
pptid	Participant ID	[number]
clinic	Field site ID	11=Framingham; 21=Hagerstown; 22=Hagerstown; 31=Minneapolis; 41=NY; 51=Pittsburgh; 52=Sacramento; 61=South Dakota; 62=Oklahoma; 63=Phoenix; 64=Tuscon
blpsgdate	Baseline PSG date	[date]
event	Event type (angina, angioplasty, CABG, CHF, MI, MIP, PTCA, revasc procedure, stent, stroke)	[text]
event_dt	Date of MI or stroke	[date]
stk_type	Type of stroke (atherothrombotic infarction, cerebral embolism, DEF_EIB, DEF_TIB, hemorrhagic, intracerebral hemorrhage, isch-cardioembolic, isch-hemorrhagic, isch-lacunar, isch-unknown, ischemic, subarachnoid hemorrhage, TIA, unknown)	[text]
fstk_type	Type of fatal stroke (atherothrombotic infarction, cerebral embolism, DEF_EIB, DEF_TIB, hemorrhagic, intracerebral hemorrhage, isch-cardioembolic, isch-hemorrhagic, isch-lacunar, isch-unknown, ischemic, subarachnoid hemorrhage, TIA, unknown)	[text]
mi_fatal	Fatal MI since baseline PSG	
stk_fatal	Fatal stroke since baseline PSG	
chf_fatal	Fatal Congestive Heart Failure (CHF) since baseline PSG	

9.9.2. Event variables

Variable Name	Variable Description	Variable Codes
revasc_fatal	Fatal revascularization procedure since baseline PSG	
cabg_fatal	Fatal CABG since baseline PSG	

9.9.3. Dataset processing specific for ARIC CVD outcome variables

---

*** NOTE: Where denoted "XX" = 2-digit year and "X" = number between 1 and 10 (eg, in 2006, INC_BYXX = INC_BY04 and CENSDATX = CENSDAT5);*

```
LIBNAME shhs1 "D:\SHHS\Dataset Requests\NHLBI _ Data Release\CD RELEASE_oct2008\SHHS 1";
LIBNAME parent "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\ARIC\...\parent-adjudicated";
LIBNAME shhs "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\ARIC\...\SHHS-adjudicated";
LIBNAME orig "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\ARIC\Aidan_files\final\original";
LIBNAME out  "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS
FUP1\DATA\2006\ARIC\Aidan_files\final\programs\Data";

%INCLUDE "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS
FUP1\DATA\2006\ARIC\...\final\programs\match_death.sas";
%INCLUDE "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS
FUP1\DATA\2006\ARIC\...\final\programs\lastcontact.sas";
%INCLUDE "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\ARIC\...final\programs\aric_mi.sas";
%INCLUDE "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\ARIC\...final\programs\aric_CHF.sas";
%INCLUDE "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\ARIC\...\final\programs\aric_angi.sas";
%INCLUDE "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS
FUP1\DATA\2006\ARIC\...\final\programs\aric_stroke.sas";
%INCLUDE "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\ARIC\...\final\programs\aric_mip.sas";
%INCLUDE "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS
FUP1\DATA\2006\ARIC\...\final\programs\aric_angina.sas";
%INCLUDE "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS
FUP1\DATA\2006\ARIC\...\programs\aric_chd_mi_death.sas";
%INCLUDE "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS
FUP1\DATA\2006\ARIC\...\final\programs\events_hag.sas";

DATA eventsa;
	SET out.mi_events
	out.chf_events
	out.angi_events
	out.stroke_events
	out.mip_events
	out.hag_events(IN=inhag WHEREe=(event NE "Death"));

	DROP death_dt;

	LENGTH type $8;
	IF inhag = 0 THEN type = "Parent";
	ELSE type = "SHHS";
RUN;

PROC SORT DATA=eventsa;
	BY pptid;
RUN;
```

9.9.3. Dataset processing specifics for ARIC CVD outcome variables

```
DATA pptids;
	SET shhs1.shhs1valids_6441;
	IF clinic ^IN (21,31) THEN DELETE;
	KEEP pptid clinic;
RUN;

PROC SORT DATA=pptids;
	BY pptid;
RUN;

DATA out.events;
	MERGE pptids (IN=a) eventsa (IN=b);
	BY pptid;
	IF a AND b;
RUN;

PROC SORT DATA=out.events;
	BY pptid event event_dt;
RUN;

DATA events;
	MERGE eventsa (IN=in1) out.lastcontact(KEEP=pptid blpsgdate censor_dt);
	BY pptid;
	IF in1;
	IF event_dt <= censor_dt;
RUN;

/* Fatal conditions */
PROC MEANS DATA=events noprint;
	VAR mi_fatal revasc_fatal chf_fatal stk_fatal cabg_fatal;
	BY pptid;
	OUTPUT out=fatal(DROP=_:) max= mi_fatal revasc_fatal chf_fatal stk_fatal cabg_fatal;
RUN;

DATA fatal;
	MERGE fatal out.lastcontact out.chd_mi;
	BY pptid;

	DROP inc inlast;

	ARRAY fatal mi_fatal revasc_fatal stk_fatal cabg_fatal chd_death mi_death;
	DO i = 1 TO DIM(fatal); DROP i;
		IF fatal[i] = . THEN fatal[i] = 0;
	END;
RUN;
```

9.9.3. Dataset processing specifics for ARIC CVD outcome variables

```
PROC SORT DATA=events;
      BY pptid event_dt;
RUN;

/* Deal with events -- up to 9 of them */
DATA ev;
      SET events(KEEP=pptid event stk_type event_dt);
      BY pptid;

      LENGTH mi_date mi_date1 8;
      LENGTH mip_date 8;
      LENGTH stk_date stk_date1 8;
      LENGTH chf_date chf_date1 chf_date2 chf_date3 chf_date4 chf_date5 chf_date6 chf_date7 chf_date8 8;
      LENGTH cabg_date cabg_date1 8;
      LENGTH revpro_date revpro_date1 revpro_date2 revpro_date3 8;
      LENGTH ptca_date ptca_date1 ptca_date2 ptca_date3 8;

      ARRAY midate mi_date mi_date1;
      ARRAY mipdate mip_date mipdate1 mipdate2 mipdate3 mipdate4;
      ARRAY stkdate stk_date stk_date1;
      ARRAY chfdate chf_date chf_date1 chf_date2 chf_date3 chf_date4 chf_date5 chf_date6 chf_date7
      chf_date8;
      ARRAY cabgdate cabg_date cabg_date1;
      ARRAY revdate revpro_date revpro_date1 revpro_date2 revpro_date3;
      ARRAY ptcadate ptca_date ptca_date1 ptca_date2 ptca_date3;

      RETAIN mi_date mi_date1;
      RETAIN mip_date;
      RETAIN stk_date stk_date1;
      RETAIN chf_date chf_date1 chf_date2 chf_date3 chf_date4 chf_date5 chf_date6 chf_date7 chf_date8;
      RETAIN cabg_date cabg_date1;
      RETAIN revpro_date revpro_date1 revpro_date2 revpro_date3;
      RETAIN ptca_date ptca_date1 ptca_date2 ptca_date3;

      FORMAT mi_date mi_date1 mip_date stk_date stk_date1 chf_date chf_date1 chf_date2 chf_date3
       chf_date4 chf_date5 chf_date6 chf_date7 chf_date8 cabg_date cabg_date1
       revpro_date revpro_date1 revpro_date2 revpro_date3
       ptca_date ptca_date1 ptca_date2 ptca_date3
       mmddyy10.;

      cutdt=MDY(4,1,2006);
      RETAIN cutdt;
      RETAIN micnt;
      RETAIN mipcnt;
      RETAIN stkcnt;
      RETAIN chfcnt;
      RETAIN cabgcnt;
```

```
RETAIN revcnt;
RETAIN ptcacnt;
RETAIN mi;
RETAIN mip;
RETAIN stroke;
RETAIN chf;
RETAIN angina;
RETAIN revasc_proc ;
RETAIN cabg;
RETAIN ptca;

IF first.pptid THEN DO;
    mi          = 0;
    mip         = 0;
    stroke      = 0;
    chf         = 0;
    angina      = 0;
    revasc_proc = 0;
    cabg        = 0;
    ptca        = 0;

    micnt       = 0;
    mipcnt      = 0;
    stkcnt      = 0;
    chfcnt      = 0;
    cabgcnt     = 0;
    revcnt      = 0;
    ptcacnt     = 0;

    mi_date     = .;
    mi_date1    = .;
    mip_date    = .;
    mip_date1   = .;
    mip_date2   = .;
    mip_date3   = .;
    mip_date4   = .;
    stk_date    = .;
    stk_date1   = .;
    chf_date    = .;
    chf_date1   = .;
    chf_date2   = .;
    chf_date3   = .;
    chf_date4   = .;
    chf_date5   = .;
    chf_date6   = .;
    chf_date7   = .;
    chf_date8   = .;
```

```
        cabg_date        = .;
        cabg_date1       = .;
        revpro_date      = .;
        revpro_date1     = .;
        revpro_date2     = .;
        revpro_date3     = .;
        ptca_date        = .;
        ptca_date1       = .;
        ptca_date2       = .;
        ptca_date3       = .;
    END;

    IF event = "MI" AND event_dt <= cutdt THEN DO;
        mi    = mi + 1;
        micnt = micnt + 1;
        midate[micnt] = event_dt;
    END;
    ELSE IF event = "MIP" AND event_dt <= cutdt THEN DO;
        mip    = mip + 1;
        mipcnt = mipcnt + 1;
        mipdate[mipcnt] = event_dt;
    END;
    ELSE IF event = "STROKE" AND event_dt <= cutdt THEN DO;
        stroke = stroke + 1;
        stkcnt = stkcnt + 1;
        stkdate[stkcnt] = event_dt;
    END;
    ELSE IF event = "CHF" AND event_dt <= cutdt THEN DO;
        chf    = chf + 1;
        chfcnt = chfcnt + 1;
        chfdate[chfcnt] = event_dt;
    END;
    ELSE IF event = "CABG" AND event_dt <= cutdt THEN DO;
        cabg   = cabg + 1;
        cabgcnt = cabgcnt + 1;
        cabgdate[cabgcnt] = event_dt;
    END;
    ELSE IF event = "REVASC PROCEDURE" AND event_dt <= cutdt THEN DO;
        revasc_proc = revasc_proc + 1;
        revcnt = revcnt + 1;
        revdate[revcnt] = event_dt;
    END;
    ELSE IF event = "ANGIOPLASTY" AND event_dt <= cutdt THEN DO;
        ptca = ptca + 1;
        ptcacnt = ptcacnt + 1;
        ptcadate[ptcacnt] = event_dt;
    END;
```

```
    IF LAST.pptid;
        DROP event event_dt;
        IF stroke=1 THEN DO;
        IF pptid='M312385' THEN stk_type='DEF_EIB';
        IF pptid='W157822' THEN stk_type='DEF_EIB';
        IF pptid='M137327' THEN stk_type='DEF_TIB';
        IF pptid='W151734' THEN stk_type='DEF_TIB';
        IF pptid='M109754' THEN stk_type='DEF_TIB';
        IF pptid='M107709' THEN stk_type='DEF_TIB';
        IF pptid='M194130' THEN stk_type='DEF_TIB';
    END;
RUN;

PROC SORT DATA=parent.aric_hom(KEEP=pptid hom10d) OUT=hom NODUPKEY;
    BY pptid;
RUN;

DATA pre;
    MERGE out.mi_pre    (RENAME=(event_dt = prev_mi_date))
        out.prev_chf  (RENAME=(event_dt = prev_chf_date))
        out.stroke_pre(RENAME=(event_dt = prev_stk_date))
        out.mip_pre   (RENAME=(event_dt = prev_mip_date))
        out.previous
        hom;
    BY pptid;

    IF hom10d = "Y" THEN prev_stk = 1;
    ELSE IF hom10d = "N" AND prev_stk = . THEN prev_stk = 0;

    IF prevmi05 = 1 THEN prev_mi = 1;
    ELSE IF prevmi05 = 0 AND prev_mi = .  THEN prev_mi = 0;

    IF prev_mi = 1 AND prev_mip = . THEN prev_mip = 1;

    KEEP pptid prev_mi prev_mip prev_stk prev_chf;
RUN;

DATA out.aric;
    MERGE out.lastcontact pre ev fatal;
    BY pptid;

    RENAME censor_dt=censdate;

    ARRAY ar mi mip stroke chf revasc_proc cabg ptca
      micnt mipcnt stkcnt chfcnt revcnt cabgcnt ptcacnt
      prev_mi prev_chf prev_stk prev_mip
      stk_fatal mi_fatal revasc_fatal chf_fatal;
```

9.9.3. Dataset processing specifics for ARIC CVD outcome variables

```
DO i = 1 TO DIM(ar);
      IF ar[i] = . THEN ar[i] = 0;
END;
DROP i inc inlast chf_fatal;

IF mi = 1 AND mip = 0 THEN DO;
      mip = 1;
      mipcnt = mipcnt +1;
END;

IF mi_fatal = 1 AND mip = 0 THEN DO;
      mip = 1;
      mipcnt = mipcnt +1;
END;

IF mi_fatal = 1 AND micnt = 0 THEN DO;
      micnt    = 1;
      mi       = 1;
      mi_date  = death_dt;
END;

IF death_dt NE . THEN DO;
      IF (stroke = 1 AND (stk_date = death_dt or stk_date1 = death_dt))  THEN stk_fatal=1;
END;

IF cabg = 1 OR mi = 1 OR ptca = 1 OR revasc_proc = 1 OR chd_death = 1 THEN anychd = 1;
ELSE anychd = 0;

IF chd_death = 1 OR stk_fatal = 1 THEN cvd_death = 1;
ELSE cvd_death = 0;

IF anychd = 1 OR cvd_death = 1 OR chf = 1 THEN anycvd = 1;
ELSE anycvd = 0;

event = MAX(mi,stroke);
IF event = 1 THEN event_dt = MIN(mi_date,stk_date);

cutdt=MDY(4,1,2006);
IF death_dt NE . AND death_dt <= cutdt THEN vital=0;
ELSE vital=1;

LABEL pptid              = 'ID';
LABEL clinic             = 'site code';
LABEL blpsgdate          = 'baseline psg date';
LABEL censor_dt          = 'date of last contact/death';
LABEL vital              = 'vital status (0=dead, 1=alive)';
```

9.9.3. Dataset processing specifics for ARIC CVD outcome variables

```
* Only 3 previous indicators ;
LABEL prev_mi          = 'mi prior to bl psg (0=no, 1+=yes)';
LABEL prev_mip         = 'mi/proc prior to bl psg (0=no, 1+=yes)';
LABEL prev_stk         = 'stroke prior to bl psg (0=no, 1+=yes)';

LABEL mi               = 'mi since bl psg (0=no, 1+=yes)';
LABEL mi_fatal         = 'fatal mi (0=no, 1=yes)';
LABEL mi_date          = 'date of first mi since bl psg';

LABEL stroke           = 'stroke since bl psg (0=no, 1+=yes)';
LABEL stk_fatal        = 'fatal stroke (0=no, 1=yes)';
LABEL stk_date         = 'date of first stroke since bl psg';
LABEL stk_type         = 'type of stroke';

LABEL event            = 'event';
LABEL event_dt         = 'date of event';

LABEL chf              = 'CHF (0=no, 1=yes)';
LABEL prev_chf         = 'CHF prior to bl psg (0=no, 1+=yes)';

LABEL cvd_death        = 'CVD death (0=no, 1=yes)';
LABEL chf_date         = 'date of CHF prior to bl psg';
LABEL chd_death        = 'CHD death (0=no, 1=yes)';

LABEL mip              = 'mi/proc since bl psg (0=no, 1+=yes)';
LABEL mip_date         = 'date of first mi/proc since bl psg';

LABEL angina           = 'Angina pectoris (0=no, 1+=yes)';

LABEL revasc_proc      = 'Revascularization procedures (0=no, 1+=yes)';
LABEL revpro_date      = 'date of first revasculariztion procedure since bl psg';

LABEL ptca             = 'PTCA (0=no, 1+=yes)';
LABEL ptca_date        = 'date of first PTCA since bl psg';

LABEL cabg             = 'CABG (0=no, 1+=yes)';
LABEL cabg_date        = 'date of first CABG since bl psg';

LABEL cabgcnt          = "Number of CABG events";
LABEL chfcnt           = "Number of CHF events";
LABEL micnt                  = "Number of MI events";
LABEL mipcnt           = "Number of MIP events";
LABEL ptcacnt          = "Number of PTCA events";
LABEL revcnt           = "Number of Revasularization events";
LABEL stkcnt           = "Number of stroke events";
LABEL chfcnt           = "Number of CHF events";
```

9.9.3. Dataset processing specifics for ARIC CVD outcome variables

```
PROC CONTENTS DATA=out.aric;
RUN;

PROC FREQ DATA=out.aric;
	TABLES vital mi prev_mi mip prev_mip stroke prev_stk chf chd_death cvd_death mi_fatal revasc_proc
cabg
	ptca anychd anycvd;
RUN;

DATA temp;
	SET out.aric;
	IF pptid="M100340" THEN OUTPUT;
	IF pptid="M137411" THEN OUTPUT;
	IF pptid="M194354" THEN OUTPUT;
	IF pptid="W121963" THEN OUTPUT;
	IF pptid="W147396" THEN OUTPUT;
	IF pptid="W152545" THEN OUTPUT;
	IF pptid="W165259" THEN OUTPUT;
	IF pptid="W170996" THEN OUTPUT;
	IF pptid="W179800" THEN OUTPUT;
	IF pptid="W215893" THEN OUTPUT;
	IF pptid="W275533" THEN OUTPUT;
	IF pptid="W296977" THEN OUTPUT;
	KEEP pptid mi mip revasc_proc;
RUN;

PROC PRINT;
RUN;

DATA temp (KEEP=pptid revasc_proc cabg ptca);
	SET out.aric;
	IF pptid='W197421' THEN OUTPUT;
	IF pptid='M128377' THEN OUTPUT;
	IF pptid='M171094' THEN OUTPUT;
	IF pptid='M184106' THEN OUTPUT;
	IF pptid='M210305' THEN OUTPUT;
	IF pptid='M307178' THEN OUTPUT;
	IF pptid='W159197' THEN OUTPUT;
	IF pptid='W170935' THEN OUTPUT;
	IF pptid='W150821' THEN OUTPUT;
RUN;
```

```
DATA tempb (KEEP=pptid mi mip mipcnt);
	SET out.aric;
	IF pptid='W152545' THEN OUTPUT tempb;
	IF pptid='W165259' THEN OUTPUT tempb;
	IF pptid='W170996' THEN OUTPUT tempb;
	IF pptid='W215893' THEN OUTPUT tempb;
RUN;
```

9.9.3.1 MATCH_DEATH – SAS Code

```
** Cohort list;
DATA s1valids;
	SET parent.shhs1valids_6441;
	WHERE clinic IN(21,31);
	KEEP pptid clinic;
	PROC SORT DATA=s1valids;
		BY pptid;
RUN;

PROC SORT DATA=parent.aric_inc_byXX OUT=dead(KEEP=pptid deadXX censdatX fatchdXX);
	BY pptid;
RUN;

/* Check for duplicates -- none */
PROC MEANS DATA=dead noprint;
	VAR deadXX;
	BY pptid;
	OUTPUT OUT=N(DROP=_:) N=n;

DATA dead;
	MERGE dead(IN=in1) s1valids(IN=in2);
	BY pptid;
	IF in1 AND in2;
RUN;

/* One duplicate M145211 */
PROC SORT DATA=parent.aric_cXXdtha1 OUT=d NODUPKEY;
	BY pptid;
RUN;

DATA d;
	MERGE d(IN=in1) s1valids(IN=in2);
	BY pptid;
	IF in1 AND in2;
RUN;
```

9.9.3. Dataset processing specifics for ARIC CVD outcome variables

```
DATA cohort;
      MERGE dead(IN=_indead) d(IN=_ind);
      BY pptid;
      indead = _indead;
      ind    = _ind;
RUN;

** Add the left censoring date;
DATA schedule;
      SET parent.psgdate6441;
      WHERE site IN (21,31);

      LENGTH nid $7;
      nid = PUT(pptid,$7.);
      RENAME nid = pptid;

      blpsgdate=DATEPART(stdydtqa);
      FORMAT blpsgdate mmddyy8.;

      KEEP nid blpsgdate;
RUN;

PROC SORT DATA=schedule NODUPKEY;
      BY pptid;
RUN;

** Exact match 1920 observations in both;
DATA out.cohort;
      MERGE cohort(IN=in1) schedule(IN=in2);
      BY pptid;

      IF in1 AND in2;
      DROP ind indead celb02;
RUN;
```

9.9.3.2 LASTCONTACT – SAS code

```
DATA lastcontact(RENAME=(npptid=pptid));
	SET shhs.hag_sa_data;
	censor_dt = MAX(lastcontact,datedeath,admitdate,q6a);
	censor_dt = MIN(censor_dt,datedeath);
	cutdt=MDY(4,1,2006);
	IF censor_dt > cutdt THEN censor_dt=cutdt;
	FORMAT censor_dt mmddyy10.;
	KEEP npptid lastcontact datedeath q6a admitdate censor_dt;

	LENGTH npptid $7;
	npptid = TRIM(LEFT(pptid));

	** Duplicates ;
	PROC SORT DATA=lastcontact;
		BY pptid;
	PROC MEANS DATA=lastcontact noprint;
		VAR lastcontact datedeath censor_dt;
		BY pptid;
		OUTPUT OUT=lastcontact(DROP=_:) MAX(lastcontact)=lastcontact MIN(datedeath
		censor_dt)=datedeath censor_dt;
RUN;

DATA out.lastcontact;
	MERGE out.cohort(IN=_inc) lastcontact(IN=_inlast);
	BY pptid;

	inc    = _inc;
	inlast = _inlast;

	FORMAT death_dt mmddyy10.;
	DROP fatchdXX;
	censor_dt = MAX(censor_dt,censdatX);
	death_dt  = MIN(dtha09,datedeath);


	KEEP pptid clinic blpsgdate censor_dt death_dt inc inlast;
RUN;
```

9.9.3.3 ARIC_MI – SAS code

```
PROC SORT DATA=parent.aric_cXXevt1 OUT=cXXevt1;
      BY pptid cmidate;
RUN;

/* Clinics match */
DATA cXXevt1;
      MERGE cXXevt1(IN=in1 DROP=clinic) out.cohort(IN=in2);
      BY pptid;
      IF in1 AND in2;

      KEEP pptid cmidate cfataldx cmidx dtha09 deadXX censdatX blpsgdate;
RUN;

DATA      events(KEEP=pptid event event_dt mi_fatal death_dt)
      pre(KEEP=pptid event_dt prev_mi);
    SET cXXevt1;
    BY pptid;

    LENGTH event $16;
    FORMAT event_dt mmddyy10.;
    FORMAT death_dt mmddyy10.;

    mi_fatal = 0;
    IF cmidx = 'DEFMI' THEN DO;
      event = "MI";
      event_dt=cmidate;
      IF cfataldx = "DEFFATMI" THEN DO;
            mi_fatal=1;
      death_dt = dtha09;
      END;
      prev_mi = 1;
      IF (cmidate <= blpsgdate) THEN OUTPUT pre;
      IF (cmidate >  blpsgdate) THEN OUTPUT events;
    END;

    IF midx = 'DEFMI' THEN DO;
      event = "MIP";
      event_dt=cmidate;
      prev_mip = 1;
      IF (cmidate <= blpsgdate) THEN OUTPUT pre;
      IF (cmidate >  blpsgdate) THEN OUTPUT events;
    END;
RUN;
```

```
PROC SORT DATA=events NODUPKEY out=out.mi_events;
   BY pptid event_dt event death_dt;
RUN;

PROC SORT DATA=pre;
   BY pptid event_dt;
RUN;

** Take the first date ;
DATA pre;
   SET pre;
   BY pptid;
   IF FIRST.pptid;
   PROC SORT DATA=pre NODUPKEY out=out.mi_pre;
     BY pptid event_dt;
RUN;
```

9.9.3.4 ARIC_CHF – SAS CODE

```
PROC SORT DATA=parent.aric_cXXcelb1 out=revasc;
   BY pptid celb04;
RUN;

DATA revasc;
   MERGE revasc(IN=in1) out.cohort(IN=in2 KEEP=pptid blpsgdate dtha09);
   BY pptid;

   IF in1 AND in2;
RUN;

DATA     events(KEEP=pptid event event_dt revasc_fatal chf_fatal death_dt)
       pre(KEEP=pptid prev_chf pre_revasc_proc event_dt);
   SET revasc;

   LENGTH event $16;
   FORMAT event_dt mmddyy10.;
   FORMAT death_dt mmddyy10.;


   ARRAY icd
     celb10a celb10b celb10c celb10d celb10e celb10f celb10g celb10h celb10i
     celb10j celb10k celb10l celb10m celb10n celb10o celb10p celb10q celb10r celb10s celb10t celb10u
     celb10v celb10w celb10x celb10y celb10z;

   LENGTH icd3 $4;
```

9.9.3. Dataset processing specifics for ARIC CVD outcome variables

```
DO i = 1 TO DIM(icd);
   icd3 = SUBSTR(LEFT(icd[i]),1,4);
   IF icd3 IN ("36.0","36.1","36.2") THEN revasc_proc=1;
   icd3 = SUBSTR(LEFT(icd[i]),1,3);
   IF icd3 = "428" THEN chf=1;
END;

pre_revasc_proc = 0;
prev_chf = 0;
revasc_fatal = 0;
chf_fatal    = 0;

death_dt = dtha09;

IF dtha09 ~= . THEN fatal = (dtha09 - celb04) + 1;
ELSE fatal = 0;
IF fatal > 7 THEN fatal = 0;

IF revasc_proc = 1 AND blpsgdate NE . THEN DO;
          event = "REVASC PROCEDURE";
          event_dt = celb04;
          pre_revasc_proc = 1;
          revasc_fatal = fatal;
          IF celb04 > blpsgdate THEN OUTPUT events;
          ELSE OUTPUT pre;
END;
pre_revasc_proc = 0;

IF revasc_proc = 1 AND blpsgdate NE . THEN DO;
   event = "MIP";
   event_dt = celb04;
   prev_mip = 1;
   IF celb04 > blpsgdate THEN OUTPUT events;
   ELSE OUTPUT pre;
END;
prev_mip = 0;
```

```
    IF chf = 1 AND blpsgdate ne . THEN DO;
      event = "CHF";
      event_dt = celb04;
      prev_chf  = 1;
      chf_fatal = fatal;
      IFcelb04 > blpsgdate THEN OUTPUT events;
      ELSE OUTPUT pre;
    END;
    prev_chf = 0;
RUN;

PROC SORT DATA=events NODUPKEY OUT=out.chf_events;
    BY pptid event_dt event revasc_fatal chf_fatal;
RUN;

** Take the first date -- just do this one;
PROC SORT DATA=pre(WHERE=(prev_chf = 1)) OUT=prev_chf(KEEP=pptid event_dt prev_chf);
    BY pptid event_dt;
RUN;

DATA out.prev_chf;
    SET prev_chf;
    BY pptid;
    IF FIRST.pptid;
RUN;
```

9.9.3.5 ARIC_ANGI – SAS Code

```
DATA f;
    SET parent.aric_cXXhrma1;
    KEEP pptid hraa14 hraa29a hraa29b hraa29c hraa29d hraa29e hraa29f hraa29g hraa29h hraa29i hraa29j
    hraa29k hraa29l hraa11a;
RUN;
```

9.9.3. Dataset processing specifics for ARIC CVD outcome variables

```
DATA schedule;
    SET parent.psgdate6441;
    WHERE site IN (21,31);

    LENGTH nid $7;
    nid = PUT(pptid,$7.);
    RENAME nid = pptid;

    blpsgdate=DATEPART(stdydtqa);
    FORMAT blpsgdate mmddyy8.;

    KEEP nid blpsgdate;
RUN;

PROC SORT DATA=schedule NODUPKEY;
    BY pptid;
RUN;

PROC SORT DATA=f;
    BY pptid hraa11a;
RUN;

DATA events(KEEP=pptid event event_dt)
        pre(KEEP=pptid pre_cabg pre_ang event_dt);

    MERGE f(IN=in1) schedule;
    BY pptid;
    IF in1;

    LENGTH event $16;
    FORMAT event_dt mmddyy10.;

    IF blpsgdate ne .;
      pre_cabg  = 0;
      pre_ang   = 0;

      IF UPCASE(hraa29f) = "Y" THEN DO;
            event = "CABG";
            event_dt = hraa11a /* arrival */ ;
            pre_cabg = 1;
            IF hraa11a > blpsgdate THEN OUTPUT events;
            ELSE OUTPUT pre;
      END;
```

```
      pre_cabg = 0;
      IF UPCASE(hraa29c) = "Y" THEN DO;
      event = "ANGIOPLASTY";
      event_dt = hraa11a /* arrival */ ;
      pre_ang = 1;
      IF hraa11a > blpsgdate THEN OUTPUT events;
      ELSE OUTPUT pre;
    END;

    pre_ang = 0;
    IF UPCASE(hraa29c) = "Y" THEN DO;
      event = "STENT";
      event_dt = hraa11a /* arrival */ ;
      pre_stent = 1;
      IF hraa11a > blpsgdate THEN OUTPUT events;
      ELSE OUTPUT pre;
    END;

    pre_stent = 0;
RUN;

PROC SORT DATA=events NODUPKEY OUT=out.angi_events;
    BY pptid event event_dt;
RUN;

PROC SORT DATA=pre NODUPKEY OUT=out.angi_pre;
    BY pptid event_dt  pre_cabg pre_ang;
RUN;
```

9.9.3.6 ARIC_STROKE – SAS Code

```
** Unique ID;
PROC SORT DATA=orig.cXXcelb1 OUT=celb NODUPKEY;
    BY id;
RUN;

** Not unique;
PROC SORT DATA=orig.cXXsdxa1 OUT=sdx  NODUPKEY;
    BY id;
RUN;

** Go back to the original and take only the ids in the new;
PROC SORT DATA=orig.cXXsdxa1 OUT=c04;
    BY celb02;
RUN;
```

9.9.3. Dataset processing specifics for ARIC CVD outcome variables

```
PROC SORT DATA=parent.aric_cXXsdxa1 OUT=new(keep=celb02 pptid) NODUPKEY;
   BY celb02;
RUN;

DATA c04;
   MERGE c04(IN=in1) new(IN=in2);
   BY celb02;
   IF in1 AND in2;

   KEEP pptid id sdxa5 sdxa1c;
RUN;

PROC SORT DATA=c04;
   BY id;
RUN;

** Original too;
PROC SORT DATA=orig.cXXcelb1 OUT=dates(KEEP=id celb04  /* discharge date */);
   BY id;
RUN;

DATA c04;
   MERGE c04(IN=in1) dates(IN=in2);
   BY id;
   IF in1;
RUN;

PROC SORT DATA=c04;
   BY pptid celb04;
RUN;

** Add the blpsgdate from schedule;
DATA schedule;
   SET parent.psgdate6441;
   WHERE site IN (21,31);

   LENGTH nid $7;
   nid = PUT(pptid,$7.);
   RENAME nid = pptid;

   blpsgdate=DATEPART(stdydtqa);
   FORMAT blpsgdate mmddyy8.;

   KEEP nid blpsgdate;
RUN;
```

9.9.3. Dataset processing specifics for ARIC CVD outcome variables

```
PROC SORT DATA=schedule NODUPKEY;
   BY pptid;
RUN;

DATA c04;
   MERGE c04(IN=in1) schedule(IN=in2);
   BY pptid;
   IF in1;
RUN;


DATA events(KEEP=pptid event event_dt stk_type)
      pre(KEEP=pptid prev_stk event_dt);
   SET c04;

   IF blpsgdate NE .;

   LENGTH event stk_type $27;
   FORMAT event_dt mmddyy10.;

   IF UPCASE(sdxa5) IN ("A","B","C","D") THEN DO;
     stroke = 1;
     event = "STROKE";
     event_dt = celb04;

     IF sdxa5='A' THEN stk_type='SUBARACHNOID HEMORRHAGE';
     IF sdxa5='B' THEN stk_type='INTRACEREBRAL HEMORRHAGE';
     IF sdxa5='C' THEN stk_type='DEF_TIB';
     IF sdxa5='D' THEN stk_type='DEF_EIB';

     ** Not possible by above condition;
     IF sdxa5='E' THEN stk_type='SUBARACHNOID HEMORRHAGE';
     IF sdxa5='F' THEN stk_type='INTRACEREBRAL HEMORRHAGE';
     IF sdxa5='G' THEN stk_type='PROB_TIB';
     IF sdxa5='H' THEN stk_type='PROB_EIB';
     IF sdxa5='I' THEN stk_type='UNKNOWN';
     IF sdxa5='J' THEN stk_type='UNKNOWN';

     prev_stk = 1;
     IF celb04 >  blpsgdate THEN OUTPUT events;
     ELSE OUTPUT pre;
   END;
RUN;

PROC SORT DATA=events NODUPKEY OUT=out.stroke_events;
   BY pptid event event_dt stk_type;
RUN;
```

```
** Take the first;
 PROC SORT DATA=pre;
    BY pptid event_dt;
RUN;

DATA out.stroke_pre;
    SET pre;
    BY pptid;
    IF FIRST.pptid;
RUN;
```

9.9.3.7 ARIC_MIP – SAS Code

```
DATA c;
    SET parent.aric_inc_byXX;
RUN;

** Unique ;
PROC SORT DATA=c;
    BY pptid;
RUN;

DATA c;
    MERGE c(IN=inc) out.cohort;
    BY pptid;
    IF inc;
RUN;

PROC SORT DATA=parent.aric_cXXevt1 OUT=cXXevt1(KEEP=pptid cfataldx cmidate dthdate);
    WHERE UPCASE(cfataldx) = "DEFFATMI";
    BY pptid;
RUN;

DATA events(KEEP=pptid event event_dt mi_fatal death_dt)
       pre(KEEP=pptid prev_mip event_dt);
    MERGE c cXXevt1;
    BY pptid;

    LENGTH event $16;
    FORMAT event_dt mmddyy10.;
    FORMAT death_dt mmddyy10.;
```

```
    IF in_byXXp = 1 or UPCASE(cfataldx) = "DEFFATMI" THEN DO;
      event = "MIP";
      event_dt = enddatep;
      prev_mip = 1;

      death_dt = dthdate;
      IF UPCASE(cfataldx) = "DEFFATMI" AND (dthdate=enddatep or cmidate=enddatep) THEN mi_fatal=1;
      ELSE mi_fatal = 0;

      IF blpsgdate < enddatep THEN OUTPUT events;
      ELSE OUTPUT pre;
    END;
RUN;

PROC SORT DATA=events NODUPKEY OUT=out.mip_events;
    BY pptid event event_dt;
RUN;

** Take the first ;
PROC SORT DATA=pre;
    BY pptid event_dt;
RUN;

DAT out.mip_pre;
    SET pre;
    BYpptid;
    IF FIRST.pptid;
RUN;
```

9.9.3.8 ARIC_ANGINA – SAS Code

```
** Unique IDs;
PROC SORT DATA=parent.aric_dervie12(KEEP=pptid prevmi05 prevhf01 prvchd05) OUT=out.previous;
    BY pptid;
RUN;
```

9.9.3.9 ARIC_CHD_MI_DEATH – SAS Code

```
PROC SORT DATA=parent.aric_cXXevt1 OUT=cXXevt1;
    BY pptid cmidate;
RUN;
```

```
** Clinics match;
DATA cXXevt1;
    MERGE cXXevt1(IN=in1 DROP=clinic) out.cohort(IN=in2);
    BY pptid;
    IF in1 AND in2;

    KEEP pptid cmidate cfataldx cmidx dtha09 deadXX censdatX blpsgdate;
RUN;

DATA chd_mi;
    SET cXXevt1;
    BY pptid;
    IF cfataldx = "DEFFATCHD" OR cfataldx = "DEFFATMI" THEN chd_death = 1;
RUN;

PROC SORT DATA=chd_mi;
    BY pptid;
RUN;

PROC MEANS DATA=chd_mi noprint;
    VAR chd_death;
    BY pptid;
    OUTPUT OUT=out.chd_mi(DROP=_:) MAX=chd_death;
RUN;
```

9.9.3.10 EVENTS_HAG – SAS Code

```
DATA schedule;
      SET parent.psgdate6441;
      WHERE site IN (21,31);

      LENGTH nid $7;
      nid = PUT(pptid,$7.);
      RENAME nid = pptid;

      blpsgdate=DATEPART(stdydtqa);
      FORMAT blpsgdate mmddyy8.;

      KEEP nid blpsgdate;
RUN;

PROC SORT DATA=schedule NODUPKEY;
      BY pptid;
RUN;
```

9.9.3. Dataset processing specifics for ARIC CVD outcome variables

```
PROC SORT DATA=shhs.hag_sa_data OUT=hag;
	BY pptid;
RUN;

DATA hag;
	MERGE hag(IN=in1) schedule;
	BY pptid;
	IF in1;
RUN;

DATA events(KEEP=pptid event event_dt stk_type mi_fatal stk_fatal cabg_fatal chf_fatal death_dt);
	SET hag;

	LENGTH event $20;
	FORMAT event_dt mmddyy10.;
	FORMAT death_dt mmddyy10.;

	mi_fatal   = 0;
	stk_fatal  = 0;
	cabg_fatal = 0;
	chf_fatal  = 0;

	IF q5a = 1 AND admitdate > blpsgdate THEN DO;
		event = "MI";
		event_dt = admitdate;
		IF q5a1 = 1 THEN DO;
			mi_fatal = 1;
			death_dt = q5a2;
		END;
		OUTPUT;
	END;

	mi_fatal = 0;

	LENGTH stk_type $20;
	IF q5j=1 AND admitdate > blpsgdate THEN DO;
		event = "STROKE";
		event_dt = admitdate;

		IF q5k = 1 THEN stk_type = "HEMORRHAGIC";
		ELSE IF q5k = 2 THEN stk_type = "ISCHEMIC";
		ELSE IF q5k = 3 THEN stk_type = "UNKNOWN";
		ELSE stk_type = " ";
```

```
        IF q5j1 = 1 THEN DO;
            stk_fatal = 1;
            death_dt = q5j2;
        END;
        OUTPUT;
    END;
    stk_fatal = 0;

    IF ( q5d = 1 OR q5e = 1 OR q5f = 1) AND admitdate > blpsgdate THEN DO;
        event = "REVASC PROCEDURE";
        event_dt = admitdate;
         OUTPUT;
    END;

    IF ( q5f = 1 AND admitdate > blpsgdate ) THEN DO ;
        event = "CABG";
        event_dt = admitdate;
        IF q5f1 = 1 THEN DO;
            cabg_fatal = 1;
            death_dt = q5f2;
        END;
        OUTPUT;
    END;
    cabg_fatal = 0;

    IF ( q5d = 1 AND admitdate > blpsgdate ) THEN DO;
        event = "ANGIOPLASTY";
        event_dt = admitdate;
        OUTPUT;
    END;

    IF ( q5e = 1 AND admitdate > blpsgdate ) THEN DO;
        event = "STENT";
        event_dt = admitdate;
         OUTPUT;
    END;

    IF ( q5b = 1 AND admitdate > blpsgdate ) THEN DO;
        event = "CHF";
        event_dt = admitdate;
        IF q5b1 = 1 THEN DO;
            chf_fatal = 1;
            death_dt = q5b2;
        END;
        OUTPUT;
    END;
    chf_fatal = 0;
```

9.9.3. Dataset processing specifics for ARIC CVD outcome variables

```
    IF q6a ~= . OR datedeath ~= . THEN DO;
        event = "Death";
        event_dt = MIN(q6a,datedeath);
        death_dt = MIN(q6a,datedeath);
        OUTPUT;
    END;
RUN;

PROC SORT DATA=events NODUPKEY OUT=out.hag_events;
    BY pptid event_dt event;
RUN;
```

9.9.4. Dataset processing specifics for CHS CVD outcome variables

```
LIBNAME data     "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\all";
LIBNAME dirs       "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA";
LIBNAME mmdat "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\M&M
requests\mm_dsr05_21aug07";
LIBNAME chs      "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\CHS";
LIBNAME chs02   "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2002\CHS";
LIBNAME library  "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\CHS";

DATA schedule;
	LENGTH namecd $ 12;
	ATTRIB blpsgdate FORMAT=mmddyy8.;
	SET dirs.psgdate6441;
	blpsgdate=DATEPART(stdydtqa);
	namecd=TRIM(LEFT(UPCASE(pptid)));
	clinic=site;
	RENAME namecd=pptid;
	shhsid=namecd;
	KEEP clinic namecd blpsgdate shhsid;
RUN;

PROC SORT;
	BY pptid;
RUN;

DATA chs;
	SET chs.chsdata_new;
RUN;

PROC SORT;
	BY pptid;
RUN;

/*prevalent disease status*/
DATA bl;
	SET chs02.tibbsreq;
	LENGTH namecd $ 12;
	namecd=TRIM(LEFT(UPCASE(IDNO)));
	RENAME namecd=pptid;
	KEEP namecd miblmod stblmod;
RUN;

PROC SORT;
	BY pptid;
RUN;
```

9.9.4. Dataset processing specifics for CHS CVD outcome variables

```
DATA chsmergea;
      MERGE schedule (IN=a) chs (IN=b);
      BY pptid;
RUN;

DATA chsmergeb;
      MERGE chsmergea (IN=a) bl (IN=b);
      BY pptid;
RUN;

/* Create dataset of N=27 not included in CHS dataset */
DATA chs.chs_noevents;
      SET chsmergeb;
      IF pptid='4025946' THEN OUTPUT chs.chs_noevents;
      IF pptid='4027531' THEN OUTPUT chs.chs_noevents;
      IF pptid='4027930' THEN OUTPUT chs.chs_noevents;
      IF pptid='4071298' THEN OUTPUT chs.chs_noevents;
      IF pptid='4071921' THEN OUTPUT chs.chs_noevents;
      IF pptid='4072162' THEN OUTPUT chs.chs_noevents;
      IF pptid='4500431' THEN OUTPUT chs.chs_noevents;
      IF pptid='5001226' THEN OUTPUT chs.chs_noevents;
      IF pptid='5021308' THEN OUTPUT chs.chs_noevents;
      IF pptid='5025001' THEN OUTPUT chs.chs_noevents;
      IF pptid='5027632' THEN OUTPUT chs.chs_noevents;
      IF pptid='5041651' THEN OUTPUT chs.chs_noevents;
      IF pptid='5042488' THEN OUTPUT chs.chs_noevents;
      IF pptid='5500842' THEN OUTPUT chs.chs_noevents;
      IF pptid='5506190' THEN OUTPUT chs.chs_noevents;
      IF pptid='5506700' THEN OUTPUT chs.chs_noevents;
      IF pptid='6025528' THEN OUTPUT chs.chs_noevents;
      IF pptid='6035159' THEN OUTPUT chs.chs_noevents;
      IF pptid='6041728' THEN OUTPUT chs.chs_noevents;
      IF pptid='6048072' THEN OUTPUT chs.chs_noevents;
      IF pptid='6070140' THEN OUTPUT chs.chs_noevents;
      IF pptid='6070663' THEN OUTPUT chs.chs_noevents;
      IF pptid='6070744' THEN OUTPUT chs.chs_noevents;
      IF pptid='6070973' THEN OUTPUT chs.chs_noevents;
      IF pptid='6071929' THEN OUTPUT chs.chs_noevents;
      IF pptid='6501680' THEN OUTPUT chs.chs_noevents;
      IF pptid='6502571' THEN OUTPUT chs.chs_noevents;
RUN;

DATA all;
      SET chsmergeb;
      IF clinic ^IN (22 51 52) THEN DELETE;   /*if death=. then delete;*/
      DROP shhsid;
RUN;
```

9.9.4. Dataset processing specifics for CHS CVD outcome variables

```
PROC FREQ DATA=all;
    TABLE cause80 chd80 death evtype fatal stktype /MISSING;
RUN;

DATA status (KEEP=clinic pptid blpsgdate vital censdate mip prev_mip prev_mi prev_stk mi mi_fatal mi_date
    stroke stk_fatal stk_date mi_death chd_death chd_dthdt cvd_death cvd_dthdt angina ang_date any_cvd
    any_chd revasc_proc revpro_date ptca ptca_date cabg cabg_date stent_date chf prev_chf chf_date
    prev_revpro prev_ang)

    event (KEEP=clinic pptid blpsgdate event event_dt stk_type fstk_type);
    SET all;
    BY pptid evdstr;

    LENGTH vital censdate mip prev_mip prev_mi prev_stk mi mi_fatal mi_date stroke stk_fatal stk_date
   mi_death chd_death cvd_death angina revasc_proc chf prev_chf any_chd any_cvd prev_ang prev_revpro 8.
   event $16. event_dt 8. stk_type $40.;

    ATTRIB censdate mi_date stk_date event_dt ang_date revpro_date ptca_date cabg_date stent_date
    chf_date chd_dthdt cvd_dthdt format=mmddyy8.;

    IF FIRST.pptid THEN DO;
        vital=.;
        censdate=.;
        mip=0;
        prev_mip=0;
        prev_mi=0;
        prev_stk=0;
        mi=0;
        mi_fatal=0;
        mi_date=.;
        stroke=0;
        stk_fatal=0;
        stk_date=.;
        mi_death=0;
        chd_death=0;
        chd_dthdt=.;
        cvd_death=0;
        cvd_dthdt=.;
        angina=0;
        ang_date=.;
        revasc_proc=0;
        revpro_date=.;
        ptca=0;
        ptca_date=.;
        cabg=0;
        cabg_date=.;
```

```
        stent=0;
        stent_date=.;
        chf=0;
        prev_chf=0;
        chf_date=.;
        any_cvd=0;
        any_chd=0;
        prev_ang=0;
        prev_revpro=0;
END;

ATTRIB cutdt FORMAT=mmddyy8.;
cutdt=MDY(4,1,2006);

RETAIN vital censdate mip prev_mip prev_mi prev_stk mi mi_fatal mi_date stroke stk_fatal stk_date
mi_death chd_death chd_dthdt cvd_death cvd_dthdt angina ang_date revasc_proc revpro_date ptca
ptca_date cabg cabg_date stent stent_date chf prev_chf chf_date any_cvd any_chd cutdt prev_ang
prev_revpro;

evdt=evdstr;
IF miblmod=1 OR (evtype=1 AND evdt <= blpsgdate) THEN prev_mip=prev_mip+1;
IF (evtype=7 AND evdt <= blpsgdate) THEN prev_mip=prev_mip+1;
IF (evtype=8 AND evdt <= blpsgdate) THEN prev_mip=prev_mip+1;

IF (evtype=1 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN mip=mip+1;
IF (evtype=7 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN mip=mip+1;
IF (evtype=8 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN mip=mip+1;

IF miblmod=1 OR (evtype=1 AND evdt <= blpsgdate) THEN prev_mi=prev_mi+1;
IF (evtype=1 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN DO;
        mi=mi+1;
        mi_fatal=(fatal=1);
        IF chd80 IN (1) THEN mi_fatal=1;
        IF mi_date=. OR mi_date > evdt THEN DO;
                mi_date=evdt;
        END;
        event='MI';
        event_dt=evdt;
        OUTPUT event;
END;

IF (evtype=1 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN any_chd=1; /* MI */
IF (evtype=7 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN any_chd=1; /* PTCA and Stent */
IF (evtype=8 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN any_chd=1; /* CABG */
IF (cause80 IN (1) OR (evtype=11 AND fatal=1) AND evdt > blpsgdate) THEN any_chd=1;
```

9.9.4. Dataset processing specifics for CHS CVD outcome variables

```
IF (evtype=1 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN any_cvd=1; /* MI */
IF (evtype=7 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN any_cvd=1; /* PTCA and Stent */
IF (evtype=8 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN any_cvd=1; /* CABG */
IF (evtype=4 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN any_cvd=1; /* CHF */
IF (cause80 IN (1) OR (evtype=11 AND fatal=1) AND evdt > blpsgdate) THEN any_cvd=1;
IF (cause80 IN (1 2 3 4) AND evdt > blpsgdate) THEN any_cvd=1; /* CVD death */

IF stblmod=1 OR (evtype=3 AND evdt <= blpsgdate) THEN prev_stk=prev_stk+1;
IF (evtype=3 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN DO;
    stroke=stroke+1;
    IF fatal=1 THEN DO;
        stk_fatal=(fatal=1);
        IF stktype=0 THEN fstk_type="HEMORRHAGIC";
        ELSE IF stktype=1 THEN fstk_type="ISCHEMIC";
        ELSE IF stktype=. THEN fstk_type="UNKNOWN";
    END;
    IF stk_date=. OR stk_date > evdt THEN DO;
        stk_date=evdt;
    END;
    IF stk_type="" OR stk_date > evdt THEN DO;
        IF stktype=0 THEN stk_type="HEMORRHAGIC";
        ELSE IF stktype=1 THEN stk_type="ISCHEMIC";
        ELSE IF stktype=. THEN stk_type="UNKNOWN";
    END;
    event='STROKE';
    event_dt=evdt;
    OUTPUT event;
END;

IF (evtype=2 AND evdt <= blpsgdate) THEN prev_ang=prev_ang+1;
IF (evtype=2 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN DO;
    angina=angina+1;
    IF ang_date=. OR ang_date > evdt THEN DO;
        ang_date=evdt;
    END;
    event='ANGINA';
    event_dt=evdt;
    OUTPUT event;
END;
```

9.9.4. Dataset processing specifics for CHS CVD outcome variables

```
IF (evtype IN(7,8) AND evdt <= blpsgdate) THEN prev_revpro=prev_revpro+1;
IF (evtype IN(7,8) AND (evdt > blpsgdate AND evdt <= cutdt)) THEN DO;
        revasc_proc=revasc_proc+1;
        IF revpro_date=. OR revpro_date > evdt THEN DO;
                revproc_date=evdt;
        END;
        event='REVASC PROCEDURE';
        event_dt=evdt;
        OUTPUT event;
END;

IF (evtype=8 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN DO;
        cabg=cabg+1;
        IF cabg_date=. OR cabg_date > evdt THEN DO;
                cabg_date=evdt;
        END;
        event='CABG';
        event_dt=evdt;
        OUTPUT event;
END;

IF (evtype=7 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN DO;
        ptca=ptca+1;
        IF ptca_date=. OR ptca_date > evdt THEN DO;
                ptca_date=evdt;
        END;
        IF evtype=7 THEN event='ANGIOPLASTY';
        IF evtype=8 THEN event='CABG';
        event_dt=evdt;
        OUTPUT event;
END;

/* Make STENT variable missing since not collected */
IF stent=0 THEN stent=.;

IF (evtype=4 AND evdt <= blpsgdate) THEN prev_chf=prev_chf+1;
IF (evtype=4 AND (evdt > blpsgdate AND evdt <= cutdt)) THEN DO;
        chf=chf+1;
        IF chf_date=. OR chf_date > evdt THEN DO;
                chf_date=evdt;
        END;
        event='CHF';
        event_dt=evdt;
        OUTPUT event;
END;
```

9.9.4. Dataset processing specifics for CHS CVD outcome variables

```
LABEL clinic='site code';
LABEL pptid='ID';
LABEL blpsgdate='baseline psg date';
LABEL vital='vital status (0=dead, 1=alive)';
LABEL censdate='date of last contact/death';
LABEL prev_mi='mi prior to bl psg (0=no, 1+=yes)';
LABEL prev_stk='stroke prior to bl psg (0=no, 1+=yes)';
LABEL mi='mi since bl psg (0=no, 1+=yes)';
LABEL mi_fatal='fatal mi (0=no, 1=yes)';
LABEL stk_fatal='fatal stroke (0=no, 1=yes)';
LABEL stroke='stroke since bl psg (0=no, 1+=yes)';
LABEL mi_date='date of first mi since bl psg';
LABEL stk_date='date of first stroke since bl psg';
LABEL chd_death='CHD death (0=no, 1=yes)';
LABEL cvd_death='CVD death (0=no, 1=yes)';
LABEL event='event';
LABEL event_dt='date of event';
LABEL stk_type='type of stroke';
LABEL angina='Angina pectoris (0=no, 1+=yes)';
LABEL ang_date='date of angina pectoris';
LABEL revasc_proc='Revascularization procedures (0=no, 1+=yes)';
LABEL revpro_date='date of revascularization procedure';
LABEL ptca='PTCA (0=no, 1+=yes)';
LABEL ptca_date='date of PTCA';
LABEL cabg='CABG (0=no, 1+=yes)';
LABEL cabg_date='date of CABG';
LABEL event='event';
LABEL event_dt='date of event';
LABEL stk_type='type of stroke';
LABEL fstk_type='type of fatal stroke';
LABEL chf='CHF (0=no, 1=yes)';
LABEL prev_chf='CHF prior to bl psg (0=no, 1+=yes)';
LABEL any_chd='CHD event since bl psg (0=no, 1=yes)';
LABEL any_cvd='CVD event since bl psg (0=no, 1=yes)';

IF (cause80 IN (1) OR (evtype=11 AND fatal=1) AND evdt > blpsgdate) THEN DO;
    chd_death=1;
    chd_dthdt=evdt;
END;

IF (cause80 IN (1 2 3 4) AND evdt > blpsgdate) THEN DO;
    cvd_death=1;
    cvd_dthdt=evdt;
END;
```

9.9.4. Dataset processing specifics for CHS CVD outcome variables

```
    IF last.pptid THEN DO;
        IF (death=1 AND mortdstr <= cutdt) THEN DO;
            vital=0;
            censdate=fcdstr;
        END;

        IF (death=1 AND mortdstr > cutdt) THEN DO;
            vital=1;
            censdate=fcdstr;
        END;

        IF (death=0) THEN vital=1;
        IF (death=0) THEN censdate=nfcdstr;

        IF censdate > cutdt THEN censdate=MDY(4,1,2006);
        OUTPUT status;
    END;
RUN;

DATA data.chs_status;
    SET status;
RUN;

DATA data.chs_event;
    SET event;
RUN;
```

9.9.5. Dataset processing specifics for FHS CVD outcome variables

```
LIBNAME data     "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\ALL";
LIBNAME dirs     "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA";
LIBNAME mmdat  "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\M&M requests\mm_dsr05_21aug07";
LIBNAME fhs      "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\FHS";

DATA schedule (KEEP=clinic pptid blpsgdate s1pptid);
	LENGTH pptid $ 8;
	ATTRIB blpsgdate FORMAT=mmddyy8.;
	SET dirs.psgdate6441;
	blpsgdate=DATEPART(stdydtqa);
	pptid=TRIM(LEFT(UPCASE(pptid)));
	clinic=site;
	IF clinic IN ('11');

	s1pptid=pptid;
RUN;

PROC SORT;
	BY pptid;
RUN;

DATA final;
	LENGTH pptid $ 8.;
	SET fhs.framdata_all;
	DROP spptid;
RUN;

PROC SORT;
	BY pptid;
RUN;

DATA fhs;
	LENGTH pptid $ 8;
	SET final;
	ATTRIB survdate FORMAT=mmddyy8.;

	IF pdata=1 AND pptid="S1-3342" THEN  DO;
		datej=MDY(5,24,2006);
		lastcon=datej;
	END;

	IF pdata=1 AND pptid="S1-3940" THEN  DO;
		datej=MDY(6,5,2006);
		lastcon=datej;
	END;
```

```
    IF pdata=1 AND pptid="S1-3977" THEN  DO;
        datej=MDY(6,7,2006);
        lastcon=datej;
    END;

    IF pdata=1 AND pptid="S1-5366" THEN  DO;
        datej=MDY(6,7,2006);
        lastcon=datej;
    END;

    IF pdata=1 AND pptid="S1-5520" THEN  DO;
        datej=MDY(3,22,2006);
        lastcon=datej;
    END;

    IF pdata=1 AND pptid="S1-5968" THEN  DO;
        datej=MDY(5,17,2006);
        lastcon=datej;
    END;

    IF pdata=1 AND pptid="S1-8459" THEN  DO;
        datej=MDY(7,8,2006);
        lastcon=datej;
    END;

    IF pdata=1 AND pptid="S7-6654" THEN  DO;
        datej=MDY(12,31,2007);
        lastcon=datej;
    END;

    IF pdata=1 AND pptid="S7-6569" THEN  DO;
        datej=MDY(9,27,2007);
        lastcon=datej;
    END;

    IF datedth NE . THEN  survdate=datedth;
    IF datedth=. THEN  survdate=lastcon;
RUN;

PROC SORT;
    BY pptid;
RUN;
```

9.9.5. Dataset processing specifics for FHS CVD outcome variables

```
DATA final;
    MERGE schedule (IN=a) fhs (IN=b);
    BY pptid;
    RENAME event=evtype date=evdate;
RUN;

DATA all;
    SET final;
RUN;

PROC SORT DATA=all;
    BY pptid evdate;
RUN;

DATA status (KEEP=pptid clinic blpsgdate vital censdate mip prev_mip prev_mi prev_stk mi mi_fatal mi_date
    stroke stk_fatal stk_date chd_death chd_dthdt cvd_death cvd_dthdt prev_ang angina ang_date revasc_proc
    revpro_date ptca ptca_date cabg cabg_date chf prev_chf chf_date any_chd any_cvd)

    event (KEEP=pptid clinic blpsgdate event event_dt stk_type fstk_type);
    SET all;
    BY pptid evdate;

    LENGTH vital censdate mip prev_mip prev_mi prev_stk mi stroke chd_death cvd_death prev_ang angina
    revasc_proc chf prev_chf any_chd any_cvd 8. event $16. stk_type $40.;

    ATTRIB censdate mi_date stk_date event_dt ang_date revpro_date ptca_date cabg_date chf_date chd_dthdt
    cvd_dthdt format=mmddyy8.;

    IF first.pptid THEN  DO;
        vital=.;
        censdate=.;
        mip=0;
        prev_mip=0;
        prev_mi=0;
        prev_stk=0;
        mi=0;
        mi_fatal=0;
        mi_date=.;
        stroke=0;
        stk_fatal=0;
        stk_date=.;
        chd_death=0;
        chd_dthdt=.;
        cvd_death=0;
        cvd_dthdt=.;
        prev_ang=0;
```

```
        angina=0;
        ang_date=.;
        revasc_proc=0;
        revpro_date=.;
        ptca=0;
        ptca_date=.;
        cabg=0;
        cabg_date=.;
        chf=0;
        prev_chf=0;
        chf_date=.;
        any_chd=0;
        any_cvd=0;
END;

RETAIN vital censdate mip prev_mip prev_mi prev_stk mi mi_fatal mi_date stroke stk_fatal stk_date  chd_death
chd_dthdt cvd_death cvd_dthdt prev_ang angina ang_date revasc_proc revpro_date ptca ptca_date cabg cabg_date
chf prev_chf chf_date any_chd any_cvd;


ATTRIB edate format=mmddyy8.;
cutdt=MDY(4,1,2006);
edate=evdate;
pdate=proc_date;
IF (evtype IN(1 2 3) AND edate <= blpsgdate) THEN  prev_mip=prev_mip+1; /* MI */
IF (PROC_NO=140 AND pdate <= blpsgdate) THEN  prev_mip=prev_mip+1; /* PTCA */
IF (PROC_NO=141 AND pdate <= blpsgdate) THEN  prev_mip=prev_mip+1; /* CABG */

IF (evtype IN(1 2 3) AND (edate > blpsgdate AND edate <= cutdt)) THEN  mip=mip+1; /* MI */
IF (PROC_NO=140 AND (pdate > blpsgdate AND edate <= cutdt)) THEN  mip=mip+1; /* PTCA */
IF (PROC_NO=141 AND (pdate > blpsgdate AND edate <= cutdt)) THEN  mip=mip+1; /* CABG */

IF evtype IN (1 2 3) AND edate <= blpsgdate THEN  prev_mi=prev_mi+1;
IF evtype IN (1 2 3) AND (edate > blpsgdate AND edate <= cutdt) THEN  DO;
        mi=mi+1;
        mi_fatal=(edate=survdate); /* mi_date equals date of death -- THERE ARE NO FATAL MIs*/
        IF mi_date=. OR mi_date > edate THEN  DO;
                mi_date=edate;
        end;
        event='MI';
        event_dt=edate;
        output event;
end;

IF (evtype IN (1 2 3) AND (edate > blpsgdate AND edate <= cutdt)) THEN  any_chd=1; /* MI */
IF (PROC_NO=140 AND (pdate > blpsgdate AND edate <= cutdt)) THEN  any_chd=1; /* PTCA */
```

9.9.5. Dataset processing specifics for FHS CVD outcome variables

```
IF (PROC_NO=141 AND (pdate > blpsgdate AND edate <= cutdt)) THEN  any_chd=1; /* CABG */
IF evtype IN (21 22 23 24) AND (edate > blpsgdate AND edate <= cutdt) THEN  any_chd=1; /* CHD dth */

IF (evtype IN (1 2 3) AND (edate > blpsgdate AND edate <= cutdt)) THEN  any_cvd=1; /* MI */
IF (PROC_NO=140 AND (pdate > blpsgdate AND edate <= cutdt)) THEN  any_cvd=1; /* PTCA */
IF (PROC_NO=141 AND (pdate > blpsgdate AND edate <= cutdt)) THEN  any_cvd=1; /* CABG */
IF evtype IN (21 22 23 24) AND (edate > blpsgdate AND edate <= cutdt) THEN  any_cvd=1; /* CHD dth */
IF evtype IN (11 13 14 15 16) AND (edate > blpsgdate AND edate <= cutdt) THEN  any_cvd=1; /* CVD dth */
IF (evtype IN(40 41) AND (edate > blpsgdate AND edate <= cutdt)) THEN  any_cvd=1; /* CHF */

IF evtype IN (11 13 14 15 16) AND edate <= blpsgdate THEN  prev_stk=1;
IF evtype IN (11 13 14 15 16) AND (edate > blpsgdate AND edate <= cutdt) THEN  DO;
      stroke=stroke+1;
      stk_fatal=(edate=survdate);

      IF stk_fatal=1 THEN  DO;
            IF evtype=11 THEN  fstk_type="ATHEROTHROMBOTIC INFARCTION";
            ELSE IF evtype=13 THEN  fstk_type="CEREBRAL EMBOLISM";
            ELSE IF evtype=14 THEN  fstk_type="INTRACEREBRAL HEMORRHAGE";
            ELSE IF evtype=15 THEN  fstk_type="SUBARACHNOID HEMORRHAGE";
            ELSE IF evtype=16 THEN  fstk_type="OTHER CVA";
      END;

      IF stk_date=. OR stk_date > edate THEN  DO;
            stk_date=edate;
      END;

      event='STROKE';
      event_dt=edate;
      IF evtype=11 THEN  stk_type="ATHEROTHROMBOTIC INFARCTION";
      ELSE IF evtype=13 THEN  stk_type="CEREBRAL EMBOLISM";
      ELSE IF evtype=14 THEN  stk_type="INTRACEREBRAL HEMORRHAGE";
      ELSE IF evtype=15 THEN  stk_type="SUBARACHNOID HEMORRHAGE";
      ELSE IF evtype=16 THEN  stk_type="OTHER CVA";
      OUTPUT event;
END;

IF (evtype=6 AND edate <= blpsgdate) THEN  prev_ang=prev_ang+1;
IF (evtype=6 AND (edate > blpsgdate AND edate <= cutdt)) THEN  DO;
      angina=angina+1;
      IF ang_date=. OR ang_date > edate THEN  DO;
            ang_date=edate;
      END;
      event='ANGINA';
      event_dt=edate;
      OUTPUT event;
END;
```

9.9.5. Dataset processing specifics for FHS CVD outcome variables

```
IF (PROC_NO IN (140 141) AND (pdate > blpsgdate AND edate <= cutdt)) THEN  DO;
      revasc_proc=revasc_proc+1;
      IF revpro_date=. OR revpro_date > pdate THEN  DO;
            revproc_date=pdate;
      END;
      event='REVASC PROCEDURE';
      event_dt=pdate;
      OUTPUT event;
END;

IF (PROC_NO=141 AND (pdate > blpsgdate AND edate <= cutdt)) THEN  DO;
      cabg=cabg+1;
      IF cabg_date=. OR cabg_date > pdate THEN  DO;
            cabg_date=pdate;
      END;
      event='CABG';
      event_dt=pdate;
      OUTPUT event;
END;

IF (PROC_NO=140 AND (pdate > blpsgdate AND edate <= cutdt)) THEN  DO;
      ptca=ptca+1;
      IF ptca_date=. OR ptca_date > pdate THEN  DO;
            ptca_date=pdate;
      END;
      event='ANGIOPLASTY';
      event_dt=pdate;
      OUTPUT event;
END;

/* STENT data not collected */

IF evtype IN (40 41) AND edate <= blpsgdate THEN  prev_chf=prev_chf+1;
IF (evtype IN(40 41) AND (edate > blpsgdate AND edate <= cutdt)) THEN  DO;
      chf=chf+1;
      IF chf_date=. OR chf_date > edate THEN  DO;
            chf_date=edate;
      END;
      event='CHF';
      event_dt=edate;
      OUTPUT event;
END;

LABEL clinic='site code';
LABEL pptid='ID';
LABEL blpsgdate='baseline psg date';
LABEL vital='vital status (0=dead, 1=alive)';
```

9.9.5. Dataset processing specifics for FHS CVD outcome variables

```
LABEL censdate='date of last contact/death';
LABEL prev_mi='mi prior to bl psg (0=no, 1+=yes)';
LABEL prev_stk='stroke prior to bl psg (0=no, 1+=yes)';
LABEL mi='mi since bl psg (0=no, 1+=yes)';
LABEL mi_fatal='fatal mi (0=no, 1=yes)';
LABEL stk_fatal='fatal stroke (0=no, 1=yes)';
LABEL stroke='stroke since bl psg (0=no, 1+=yes)';
LABEL mi_date='date of first mi since bl psg';
LABEL stk_date='date of first stroke since bl psg';
LABEL chd_death='CHD death (0=no, 1=yes)';
LABEL cvd_death='CVD death (0=no, 1=yes)';
LABEL event='event';
LABEL event_dt='date of event';
LABEL stk_type='type of stroke';
LABEL angina='Angina pectoris (0=no, 1+=yes)';
LABEL ang_date='date of angina pectoris';
LABEL revasc_proc='Revascularization procedures (0=no, 1+=yes)';
LABEL revpro_date='date of revascularization procedure';
LABEL ptca='PTCA (0=no, 1+=yes)';
LABEL ptca_date='date of PTCA';
LABEL cabg='CABG (0=no, 1+=yes)';
LABEL cabg_date='date of CABG';
LABEL event='event';
LABEL event_dt='date of event';
LABEL stk_type='type of stroke';
LABEL fstk_type='type of fatal stroke';
LABEL chf='CHF (0=no, 1=yes)';
LABEL prev_chf='CHF prior to bl psg (0=no, 1+=yes)';
LABEL any_chd='CHD event since bl psg (0=no, 1=yes)';
LABEL any_cvd='CVD event since bl psg (0=no, 1=yes)';

IF last.pptid THEN  DO;
    IF evtype ^IN (21 22 23 24 25 26 27 28 29) THEN  vital=1;
    IF (evtype IN (21 22 23 24 25 26 27 28 29) AND datedth <= cutdt) THEN  vital=0;
    IF (evtype IN (21 22 23 24 25 26 27 28 29) AND datedth > cutdt) THEN  vital=1;

    censdate=survdate;
    IF censdate > cutdt THEN  censdate=MDY(4,1,2006);

    IF evtype IN (21 22 23 24) THEN  DO;
        chd_death=1;
        chd_dthdt=datedth;
    END;
```

9.9.5. Dataset processing specifics for FHS CVD outcome variables

```
            IF evtype IN (21 22 23 24 25 26) THEN  DO;
                cvd_death=1;
                cvd_dthdt=datedth;
            END;

            OUTPUT status;
        END;
RUN;

DATA data.fhs_status;
        SET status;
        ** PER George O'Connor -- Remove two FRAM participants that had PSG study and then left the study;
        IF pptid="S7-6594" THEN DELETE;
        IF pptid="S7-6747" THEN DELETE;
RUN;

DATA data.fhs_event;
        SET event;
RUN;
```

9.9.6. Dataset processing specifics for SHS CVD outcome variables

```
%LET drive=D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1; /*working directory*/
%LET study=shhs;
%LET year=2006; /*year outcomes data transmitted from parent studies to CC*/
%LET cutyr=06;  /*events cut-off year*/

LIBNAME shhs1    "D:\SHHS\Dataset Requests\NHLBI _ Data Release\CD RELEASE_nov2006\SHHS 1";
LIBNAME shhs2    "D:\SHHS\Dataset Requests\NHLBI _ Data Release\CD RELEASE_nov2006\SHHS 2";
LIBNAME data     "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\ALL";
LIBNAME dirs     "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA";
LIBNAME mmdat  "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\M&M
requests\mm_dsr05_21aug07";
LIBNAME shs      "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\SHS";
LIBNAME shs_sa   "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\SHS\SHHS-adjudicated";
LIBNAME shs_pa   "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\SHS\Parent-adjudicated";
LIBNAME shs_2004 "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\data\2004\shs";

DATA shhs2valids;
      FORMATpptid $12.;
      SET shhs2.shhs2valids_4586;
      IF clinic ^IN(61,62,63) THEN DELETE;
      s2=1;
      KEEP pptid s2;
RUN;

PROC SORT DATA=shhs2valids;
      BY pptid;
RUN;

DATA sa_data;
      FORMAT pptid $12.;
      SET shs_sa.shs_sa_data;
      pptid=TRIM(LEFT(studyid));
      IF ac_date = . THEN pvital=0;
      IF ac_date NE . THEN pvital=1;
      DROP studyid;
RUN;

DATA pa_data;
      FORMAT pptid $12.;
      SET shs_pa.shsdata;
      pptid=TRIM(LEFT(pptid));
      RENAME vital=pvital;
RUN;

PROC SORT DATA=sa_data;
      BY pptid;
RUN;
```

9.9.6. Dataset processing specifics for SHS CVD outcome variables

```
PROC SORT DATA=pa_data;
	BY pptid;
RUN;

DATA s2indicator;
	MERGE pa_DATA (IN=a) shhs2valids (IN=b);
	BY pptid;
	IF a;
RUN;

DATA shs.shs_data;
	MERGE s2indicator (IN=a) sa_DATA (IN=b);
	BY pptid;
	sadata=0;
	IF a;
	IF b THEN sadata=1;
	IF b THEN DO;
		s1examdt=.;
		s2examdt=.;
		s3examdt=.;
		dod=.;
		s1dmi_c=.;
		s1dmidt=.;
		s1pmi_c=.;
		s1pmidt=.;
		defmi_c=.;
		defmidt=.;
		newdefmi=.;
		ndmi_dt=.;
		newposmi=.;
		npmi_dt=.;
		posmi_c=.;
		posmidt=.;
		s1defmi=.;
		s1posmi=.;
		deadcode=.;
		fatalmi=. ;
		defstk_c=.;
		defstkdt=.;
		stktyped=.;
		posstk_c=.;
		posstkdt=.;
		nwdefstk=.;
		ndstk_dt=.;
		nwposstk=.;
		npstk_dt=.;
		fatalstk=.;
```

```
            s1defstk=.;
            s1posstk=.;
            s1dstk_c=.;
            s1dstkdt=.;
            s1pstk_c=.;
            s1pstkdt=.;
            stktypel=.;
            defchd_c=.;
            defchddt=.;
            fatalchd=.;
            nwdefchd=.;
            ndchd_dt=.;
            poschd_c=.;
            poschddt=.;
            nwposchd=.;
            npchd_dt=.;
            s1dchd_c=.;
            s1dchddt=.;
            s1pchd_c=.;
            s1pchddt=.;
            s1defchd=.;
            s1poschd=.;
            s1apecg=.;
            s2apecg=.;
            s3apecg=.;
            s1chf_c=.;
            s1chfdt=.;
            shschf_c=.;
            shschfdt=.;
            s1shschf=.;
            nwshschf=.;
            nwchf_dt=.;
            fatalchf=.;
        END;
RUN;

%MACRO makedata;
PROC CONTENTS DATA=shs.shs_data;
RUN;
```

9.9.6. Dataset processing specifics for SHS CVD outcome variables

```
DATA schedule (KEEP=clinic pptid blpsgdate);
      LENGTH pptid $ 12;
      ATTRIB blpsgdate format=mmddyy8.;
      SET dirs.psgdate6441;
      IF site NOT in(61,62,63) THEN DELETE;
      namecd=TRIM(LEFT(UPCASE(pptid)));
      blpsgdate=DATEPART(stdydtqa);
      clinic=site;
RUN;

PROC SORT;
      BY pptid;
RUN;

DATA shs;
      SET shs.shs_data;
      pptid=TRIM(LEFT(UPCASE(pptid)));
      IF dod NE . THEN death=1;
      RENAME defchd_c=defchd;
      RENAME defmi_c=defmi;
      RENAME defstk_c=defstk;
      RENAME poschd_c=poschd;
      RENAME posmi_c=posmi;
      RENAME posstk_c=posstk;
      IF namecd='103011' THEN namecd='103067';
RUN;

PROC SORT;
      BY namecd;
RUN;

DATA final;
      MERGE schedule (IN=a) shs (IN=b);
      BY pptid;
      IF a;
RUN;

DATA all;
      SET final;
RUN;

PROC FREQ DATA=all;
      TABLES defmi*newdefmi s1defmi*s1dmi_c /MISSING;
RUN;
```

9.9.6. Dataset processing specifics for SHS CVD outcome variables

```
PROC FREQ;
    TABLE deadcode defchd defmi newdefmi s1defmi s1dmi_c cm1004 defstk poschd posmi posstk death
    /MISSING;
RUN;

PROC FREQ;
    TABLE s1dstk_c s1defstk defstk nwdefstk st1002 st1004;
RUN;

PROC FREQ;
    TABLE s1apecg s2apecg s3apecg shschf_c s1shschf nwshschf s1chf_c fr1004b fr1003b pvital
    ac_date ac1005;
RUN;

DATA status  (KEEP=clinic pptid blpsgdate vital pvital censdate mip prev_mip prev_mi prev_stk
    mi mi_fatal mi_date stroke stk_fatal stk_date chd_death chd_dthdt cvd_death
    cvd_dthdt angina ang_date revasc_proc revpro_date ptca ptca_date cabg cabg_date
    chf prev_chf chf_date any_chd any_cvd prev_revpro)

    event (KEEP=clinic pptid blpsgdate event event_dt stk_type fstk_type);
    SET all;
    BY pptid sleepday;

    LENGTH vital censdate mip prev_mip prev_mi prev_stk mi mi_fatal mi_date stroke stk_fatal stk_date
    chd_death cvd_death angina revasc_proc chf prev_chf any_chd any_cvd prev_revpro 8. event $16.
    event_dt 8. stk_type $40.;

    ATTRIB censdate mi_date stk_date event_dt ang_date revpro_date ptca_date cabg_date chf_date
    chd_dthdt cvd_dthdt format=mmddyy8.;

    IF first.pptid THEN DO;
        vital=.;
        censdate=.;
        mip=0;
        prev_mip=0;
        prev_mi=0;
        prev_stk=0;
        mi=0;
        mi_fatal=0;
        mi_date=.;
        stroke=0;
        stk_fatal=0;
        stk_date=.;
        chd_death=0;
        chd_dthdt=.;
        cvd_death=0;
```

```
        cvd_dthdt=.;
        angina=0;
        ang_date=.;
        revasc_proc=0;
        revpro_date=.;
        ptca=0;
        ptca_date=.;
        cabg=0;
        cabg_date=.;
        chf=0;
        prev_chf=0;
        chf_date=.;
        any_chd=0;
        any_cvd=0;
        prev_revpro=0;
END;

ATTRIB cutdt format=mmddyy8.;
cutdt=MDY(4,1,2006);

RETAIN vital censdate mip prev_mip prev_mi prev_stk mi mi_fatal mi_date stroke stk_fatal stk_date
chd_death chd_dthdt cvd_death cvd_dthdt angina ang_date revasc_proc revpro_date ptca ptca_date cabg
cabg_date chf prev_chf chf_date any_chd any_cvd prev_revpro cutdt;

IF defmi=1 AND defmidt <= blpsgdate THEN DO;
        prev_mi=prev_mi+1;
        prev_mip=prev_mip+1;
END;

IF defmi=1 AND (defmidt > blpsgdate AND defmidt <= cutdt) THEN DO;
        mi=mi+1;
        mip=mip+1;
        mi_fatal=(deadcode=1);
        IF mi_date=. OR mi_date > defmidt THEN DO;
                mi_date=defmidt;
        END;
        event='MI';
        event_dt=defmidt;
        OUTPUT event;
END;

IF newdefmi=1 AND ndmi_dt <= blpsgdate THEN DO;
        prev_mi=prev_mi+1;
        prev_mip=prev_mip+1;
END;
```

```
IF s1defmi=1 THEN DO;
      prev_mi=prev_mi+1;
      prev_mip=prev_mip+1;
END;

IF (s1dmi_c=1 AND prev_mi = 0) AND s1examdt <= blpsgdate THEN prev_mi=1;
IF (s1dmi_c=1 AND prev_mip = 0) AND s1examdt <= blpsgdate THEN prev_mip=1;

IF newdefmi=1 AND (ndmi_dt > blpsgdate AND ndmi_dt <= cutdt) THEN DO;
      mi=mi+1;
      mip=mip+1;
      mi_fatal=(deadcode=1);
      IF mi_date=. OR mi_date > ndmi_dt THEN DO;
            mi_date=ndmi_dt;
      END;
      event='MI';
      event_dt=ndmi_dt;
      OUTPUT event;
END;

IF (fr1003a=1 AND (me_date > blpsgdate AND me_date <= cutdt)) THEN DO;
      mi=mi+1;
      IF mi_date=. OR mi_date > me_date THEN DO;
            mi_date=me_date;
      END;
      event='MI';
      event_dt=ndmi_dt;
      OUTPUT event;
END;

IF cm1004=1 AND ac_date <= blpsgdate THEN DO;
      prev_mi=prev_mi+1;
      prev_mip=prev_mip+1;
END;

IF cm1004=1 AND (ac_date > blpsgdate AND ac_date <= cutdt) THEN DO;
      mi=mi+1;
      mip=mip+1;
      IF (cm1004=1 AND ac_date NE "") THEN mi_fatal=1;
      IF mi_date=. OR mi_date > ac_date THEN DO;
            mi_date=ac_date;
      END;
      event='MI';
      event_dt=ac_date;
      OUTPUT event;
END;
```

```
othcard=fr1003ds;
i=INDEX(othcard, 'ptca');
ii=INDEX(othcard, 'PTCA');
iii=INDEX(othcard, 'cabg');
iv=INDEX(othcard, 'CABG');

IF s1dmi_c=1 AND (s1examdt > blpsgdate AND s1examdt <= cutdt) THEN any_chd=1;
IF defmi=1 AND (defmidt > blpsgdate AND defmidt <= cutdt) THEN any_chd=1;
IF newdefmi=1 AND (ndmi_dt > blpsgdate AND ndmi_dt <= cutdt) THEN any_chd=1;
IF cm1004=1 AND (ac_date > blpsgdate AND ac_date <= cutdt) THEN any_chd=1;
IF defchd=1 AND (defchddt > blpsgdate AND defchddt <= cutdt) THEN any_chd=1;
IF (fr1003d=1 AND (i NE 0 OR ii NE 0 OR iii NE 0 OR iv NE 0) AND (me_date > blpsgdate AND
me_date <= cutdt)) THEN any_chd=1;
IF (deadcode IN (1 2 3 4) AND (dod > blpsgdate AND dod <= cutdt)) THEN any_chd=1; /* CHD death */

IF s1dmi_c=1 AND (s1examdt > blpsgdate AND s1examdt <= cutdt) THEN any_cvd=1;
IF defmi=1 AND (defmidt > blpsgdate AND defmidt <= cutdt) THEN any_cvd=1;
IF cm1004=1 AND (ac_date > blpsgdate AND ac_date <= cutdt) THEN any_cvd=1;
IF defchd=1 AND (defchddt > blpsgdate AND defchddt <= cutdt) THEN any_cvd=1;
IF (fr1003d=1 AND (i NE 0 OR ii NE 0 OR iii NE 0 OR iv NE 0) AND (me_date > blpsgdate AND
me_date <= cutdt)) THEN any_cvd=1;
IF (deadcode IN (1 2 3 4) AND (dod > blpsgdate AND dod <= cutdt)) THEN any_cvd=1; /* CHD death */
IF (nwshschf=1 AND (nwchf_dt > blpsgdate AND nwchf_dt <= cutdt)) THEN any_cvd=1; /* CHF */
IF (fr1003b=1 AND (me_date > blpsgdate AND me_date <= cutdt)) THEN any_cvd=1;
IF deadcode IN (5 6 7 8 9) THEN any_cvd=1; /* CVD death */

IF s1dstk_c=1 AND s1examdt <= blpsgdate THEN prev_stk=prev_stk+1;
IF s1defstk=1 AND s1examdt <= blpsgdate THEN prev_stk=prev_stk+1;

IF nwdefstk=1 AND ndstk_dt <= blpsgdate THEN prev_stk=prev_stk+1;
IF nwdefstk=1 AND (ndstk_dt > blpsgdate AND ndstk_dt <= cutdt) THEN DO;
    stroke=stroke+1;
    stk_fatal=(deadcode in(5,6));

    IF stk_fatal=1 THEN DO;
        IF stktyped=1 THEN fstk_type='ISCH-CARDIOEMBOLIC';
        IF stktyped=2 THEN fstk_type='SUBARACHNOID HEMORRHAGE';
        IF stktyped=3 THEN fstk_type='INTRACEREBRAL HEMORRHAGE';
        IF stktyped=4 THEN fstk_type='ISCH-LACUNAR';
        IF stktyped=5 THEN fstk_type='ISCH-UNKNOWN';
        IF stktyped=6 THEN fstk_type='TIA';
        IF stktyped=7 THEN fstk_type='UNKNOWN';
        IF stktyped=8 THEN fstk_type='ATHEROTHROMBOTIC INFARCTION';
    END;
```

```
        IF stk_fatal=0 THEN DO;
            IF stktypel=1 THEN fstk_type='ISCH-CARDIOEMBOLIC';
            IF stktypel=2 THEN fstk_type='SUBARACHNOID HEMORRHAGE';
            IF stktypel=3 THEN fstk_type='INTRACEREBRAL HEMORRHAGE';
            IF stktypel=4 THEN fstk_type='ISCH-LACUNAR';
            IF stktypel=5 THEN fstk_type='ISCH-UNKNOWN';
            IF stktypel=6 THEN fstk_type='TIA';
            IF stktypel=7 THEN fstk_type='UNKNOWN';
            IF stktypel=8 THEN fstk_type='ATHEROTHROMBOTIC INFARCTION';
        END;

        IF stk_date=. OR stk_date > ndstk_dt THEN DO;
            stk_date=ndstk_dt;
        END;
        event='STROKE';
        event_dt=ndstk_dt;
        OUTPUT event;
    END;

    IF st1002=1 AND ac_date <= blpsgdate THEN prev_stk=prev_stk+1;
    IF st1002=1 AND (ac_date > blpsgdate AND ac_date <= cutdt) THEN DO;
        stroke=stroke+1;
        IF (st1002=1 AND ac_date NE "") THEN DO;
            stk_fatal=1;

            IF st1004=1 THEN DO;
                IF st1004a=1 THEN fstk_type='SUBARACHNOID HEMORRHAGE';
                IF st1004a=2 THEN fstk_type='INTRACEREBRAL HEMORRHAGE';
                IF st1004a=3 THEN fstk_type='UNKNOWN';
            END;

            IF st1004=2 THEN DO;
                IF st1004b=1 THEN fstk_type='ISCH-LACUNAR';
                IF st1004b=2 THEN fstk_type='ISCH-CARDIOEMBOLIC';
                IF st1004b=3 THEN fstk_type='ISCH-ATHEROSCLEROTIC';
                IF st1004b=4 THEN fstk_type='ISCH-HEMORRHAGIC';
                IF st1004b=5 THEN fstk_type='ISCH-UNKNOWN';
            END;

            IF st1004=3 THEN fstk_type='UNKNOWN';
        END;

        IF st1004=1 THEN DO;
            IF st1004a=1 THEN stk_type='SUBARACHNOID HEMORRHAGE';
            IF st1004a=2 THEN stk_type='INTRACEREBRAL HEMORRHAGE';
            IF st1004a=3 THEN stk_type='UNKNOWN';
        END;
```

```
        IF st1004=2 THEN DO;
            IF st1004b=1 THEN stk_type='ISCH-LACUNAR';
            IF st1004b=2 THEN stk_type='ISCH-CARDIOEMBOLIC';
            IF st1004b=3 THEN stk_type='ISCH-ATHEROSCLEROTIC';
            IF st1004b=4 THEN stk_type='ISCH-HEMORRHAGIC';
            IF st1004b=5 THEN stk_type='ISCH-UNKNOWN';
        END;

        IF st1004=3 THEN stk_type='UNKNOWN';

        IF stk_date=. OR stk_date > ac_date THEN DO;
            stk_date=ac_date;
        END;
        event='STROKE';
        event_dt=ac_date;
        OUTPUT event;
    END;

    IF (fr1003g AND (me_date > blpsgdate AND me_date <= cutdt)) THEN DO;
        stroke=stroke+1;
        IF stk_date=. OR stk_date > me_date THEN DO;
            stk_date=me_date;
        END;

        IF fr1005=1 THEN DO;
            IF fr1005a=1 THEN stk_type='SUBARACHNOID HEMORRHAGE';
            IF fr1005a=2 THEN stk_type='INTRACEREBRAL HEMORRHAGE';
            IF fr1005a=3 THEN stk_type='UNKNOWN';
        END;

        IF fr1005=2 THEN DO;
            IF fr1005b=1 THEN stk_type='ISCH-LACUNAR';
            IF fr1005b=2 THEN stk_type='ISCH-CARDIOEMBOLIC';
            IF fr1005b=3 THEN stk_type='ISCH-ATHEROSCLEROTIC';
            IF fr1005b=4 THEN stk_type='ISCH-HEMORRHAGIC';
            IF fr1005b=5 THEN stk_type='ISCH-UNKNOWN';
        END;

        IF fr1005=3 THEN stk_type='UNKNOWN';
    END;
```

9.9.6. Dataset processing specifics for SHS CVD outcome variables

```
IF (S1APECG=1 OR S2APECG=1 OR S3APECG=1) THEN angina=angina+1;
IF S1APECG=1 AND (S1EXAMDT > blpsgdate AND S1EXAMDT <= cutdt) THEN DO;
      angina=angina+1;
      IF ang_date=. OR ang_date > S1EXAMDT THEN DO;
            ang_date=S1EXAMDT;
       END;
END;

IF S2APECG=1 AND (S2EXAMDT > blpsgdate AND S2EXAMDT <= cutdt) THEN DO;
      angina=angina+1;
      IF ang_date=. OR ang_date > S2EXAMDT THEN DO;
             ang_date=S2EXAMDT;
      END;
END;

IF S3APECG=1 AND (S3EXAMDT > blpsgdate AND S3EXAMDT <= cutdt) THEN DO;
      angina=angina+1;
      IF ang_date=. OR ang_date > S3EXAMDT THEN DO;
            ang_date=S3EXAMDT;
      END;
      event='ANGINA';
      IF S1APECG=1 AND (S1EXAMDT > blpsgdate AND S1EXAMDT <= cutdt) THEN
      evdt=S1APECG;
      IF S2APECG=1 AND (S2EXAMDT > blpsgdate AND S2EXAMDT <= cutdt) THEN
      evdt=S2APECG;
      IF S3APECG=1 AND (S3EXAMDT > blpsgdate AND S3EXAMDT <= cutdt) THEN
      evdt=S3APECG;
      event_dt=evdt;
      OUTPUT event;
END;

othcard=fr1003ds;
i=INDEX(othcard, 'Angina');
ii=INDEX(othcard, 'angina');
IF (fr1003d=1 AND (i NE 0 OR ii NE 0) ) THEN DO;
      angina=angina+1;
      IF ang_date=. OR (ang_date > me_date) THEN DO;
            ang_date=me_date;
      END;
      event='ANGINA';
      event_dt=me_date;
      OUTPUT event;
END;
```

9.9.6. Dataset processing specifics for SHS CVD outcome variables

```
/* No variables specific to CABG, PTCA, OR Stent */
ptca=.;
cabg=.;
ptca_date=.;
cabg_date=.;

IF (defchd=1 AND defchddt <= blpsgdate) THEN DO;
	prev_revpro=prev_revpro+1;
	prev_mip=prev_mip+1;
END;

IF (defchd=1 AND (defchddt > blpsgdate AND defchddt <= cutdt)) THEN DO;
	 revasc_proc=revasc_proc+1;
	mip=mip+1;
	IF revpro_date=. OR revpro_date > defchddt THEN DO;
		revpro_date=defchddt;
	END;
	event='REVASC PROCEDURE';
	event_dt=defchddt;
	OUTPUT event;
END;

othcard=fr1003ds;
i=INDEX(othcard, 'ptca');
ii=INDEX(othcard, 'PTCA');
iii=INDEX(othcard, 'cabg');
iv=INDEX(othcard, 'CABG');

IF (fr1003d=1 AND (i NE n0 OR ii NE 0 OR iii NE 0 OR iv NE 0) AND me_date <= cutdt) THEN DO;
	revasc_proc=revasc_proc+1;
	mip=mip+1;
	IF revpro_date=. OR (revpro_date > me_date) THEN DO;
		revpro_date=me_date;
	END;
	event='REVASC PROCEDURE';
	event_dt=me_date;
	OUTPUT event;
END;

IF shschf_c=1 AND shschfdt <= blpsgdate THEN prev_chf=prev_chf+1;
IF s1chf_c=1 AND s1chfdt <= blpsgdate THEN prev_chf=prev_chf+1;
IF s1shschf=1 THEN DO;
	prev_chf=prev_chf+1;
	event='CHF';
	event_dt=.;
	OUTPUT event;
END;
```

```
IF (nwshschf=1 AND (nwchf_dt > blpsgdate AND nwchf_dt <= cutdt)) THEN DO;
      chf=chf+1;
      IF chf_date=. OR chf_date > nwchf_dt THEN DO;
            chf_date=nwchf_dt;
      END;
      event='CHF';
      event_dt=nwchf_dt;
      OUTPUT event;
END;

IF (fr1003b=1 AND (me_date NE . AND me_date > blpsgdate AND me_date <= cutdt)) THEN DO;
      chf=chf+1;
      IF chf_date="" OR chf_date > me_date THEN DO;
            chf_date=me_date;
      END;
      event='CHF';
      event_dt=me_date;
      OUTPUT event;
END;

LABEL clinic='site code';
LABEL pptid='ID';
LABEL blpsgdate='baseline psg date';
LABEL vital='vital status (0=dead, 1=alive)';
LABEL censdate='date of last contact/death';
LABEL prev_mi='mi prior to bl psg (0=no, 1+=yes)';
LABEL prev_stk='stroke prior to bl psg (0=no, 1+=yes)';
LABEL mi='mi since bl psg (0=no, 1+=yes)';
LABEL mi_fatal='fatal mi (0=no, 1=yes)';
LABEL stk_fatal='fatal stroke (0=no, 1=yes)';
LABEL stroke='stroke since bl psg (0=no, 1+=yes)';
LABEL mi_date='date of first mi since bl psg';
LABEL stk_date='date of first stroke since bl psg';
LABEL chd_death='CHD death (0=no, 1=yes)';
LABEL cvd_death='CVD death (0=no, 1=yes)';
LABEL event='event';
LABEL event_dt='date of event';
LABEL stk_type='type of stroke';
LABEL angina='Angina pectoris (0=no, 1+=yes)';
LABEL ang_date='date of angina pectoris';
LABEL revasc_proc='Revascularization procedures (0=no, 1+=yes)';
LABEL revpro_date='date of revascularization procedure';
LABEL ptca='PTCA (0=no, 1+=yes)';
LABEL ptca_date='date of PTCA';
LABEL cabg='CABG (0=no, 1+=yes)';
LABEL cabg_date='date of CABG';
LABEL event='event';
```

9.9.6. Dataset processing specifics for SHS CVD outcome variables

```
LABEL event_dt='date of event';
LABEL stk_type='type of stroke';
LABEL fstk_type='type of fatal stroke';
LABEL chf='CHF (0=no, 1=yes)';
LABEL prev_chf='CHF prior to bl psg (0=no, 1+=yes)';
LABEL any_chd='CHD event since bl psg (0=no, 1=yes)';
LABEL any_cvd='CVD event since bl psg (0=no, 1=yes)';

IF (pvital=1 AND dod <= cutdt) THEN vital=0;
IF (pvital=1 AND dod > cutdt) THEN vital=1;
IF pvital=1 THEN censdate=dod;
IF pvital=0 THEN vital=1;
IF pvital=0 THEN censdate=studyEND;

IF (ac1005 NE . AND ac1005 in('1' '2' '3' '4' '5')) THEN DO;
	vital=0;
	censdate=ac_date;
END;

IF vital=1 AND me_date NE . THEN DO;
	censdate=mdy(4,1,06);
END;

IF pptid="302483" THEN censdate=MDY(1,28,06);
IF pptid="103078" THEN censdate=MDY(9,25,05);
IF pptid="303244" THEN censdate=MDY(10,24,05);
IF pptid="202246" THEN censdate=MDY(6,20,05);
IF pptid="303040" THEN censdate=MDY(6,8,05);
IF pptid="103205" THEN censdate=MDY(10,9,05);
IF pptid="103030" THEN censdate=MDY(2,10,04);
IF pptid="103067" THEN censdate=MDY(5,14,02);

IF mi=0 THEN DO;
	IF deadcode IN (1) THEN DO;
		mi=mi+1;
		mip=mip+1;
		mi_date=dod;
		mi_fatal=1;
		event='MI';
		event_dt=mi_date;
		OUTPUT event;
	END;
END;
```

9.9.6. Dataset processing specifics for SHS CVD outcome variables

```
    IF stroke=0 THEN DO;
        IF deadcode IN (5) THEN DO;
            stroke=stroke+1;
            stk_date=dod;
            stk_fatal=1;
            event='STROKE';
            event_dt=stk_date;
            OUTPUT event;
        END;
    END;

    IF (deadcode IN (1 2 3 4) AND (dod > blpsgdate AND dod <= cutdt)) THEN DO;
        chd_death=1;
        chd_dthdt=dod;
    END;

    IF (deadcode IN (1 2 3 4 5 6 7 8 9) AND (dod > blpsgdate AND dod <= cutdt)) THEN DO;
        cvd_death=1;
        cvd_dthdt=dod;
    END;

    IF pptid="103030" THEN vital=0;
    IF pptid="103067" THEN vital=0;

    IF last.pptid THEN DO;
        OUTPUT status;
    END;
RUN;

DATA data.shs_status;
    SET status;
    DROP pvital;
RUN;

DATA data.shs_event;
    SET event;
RUN;

%MEND;

%makedata;
```

9.9.7. Dataset processing specifics for Tucson CVD outcome variables

```
LIBNAME data    "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\ALL";
LIBNAME dirs    "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA";
LIBNAME tuc     "D:\SHHS\Dataset Requests\Miscellaneous\CVD outcomes SHHS FUP1\DATA\2006\tucson";
LIBNAME s1      "D:\SHHS\Dataset Requests\NHLBI _ Data Release\CD RELEASE_nov2006\SHHS 1";
LIBNAME s2      "D:\SHHS\Dataset Requests\NHLBI _ Data Release\CD RELEASE_nov2006\SHHS 2";

DATA schedule;
    SET dirs.psgdate6441;
    IF site NE 64 THEN DELETE;
    ATTRIB blpsgdate format=mmddyy8.;
    blpsgdate=DATEPART(stdydtqa);
    clinic=site;
    DROP stdydtqa site;
RUN;

PROC SORT;
    BY pptid;
RUN;

DATA tuc;
    SET tuc.tucdata_all;
    ATTRIB nmorbdt format=mmddyy8.;
    ATTRIB nsddate format=mmddyy8.;
    nmorbdt=DATEPART(morbdt);
    nsddate=DATEPART(sddate);
    ncddate=DATEPART(cddate);
    pchf=chf;
    pprev_chf=prev_chf;

    DROP chf prev_chf morbdt sddate cddate;
    RENAME _2=cdcod _7=cdmech nmorbdt=morbdt nsddate=sddate ncddate=cddate;
RUN;

DATA all;
    MERGE schedule (IN=a) tuc (IN=b);
    BY pptid;
    IF a AND ^b THEN PUT 'no matching outcomes data: ' pptid=;
    IF a AND b;
    cardproc=0; /*check text field to make sure capturing all variations each time updating the file*/
    IF SUBSTR(specify_other_card, 1, 4) IN ('PTCA' 'CABG') OR
        SUBSTR(specify_other_card, 1, 5) IN ('Angio' 'Stent') OR
        INDEX(specify_other_card, 'stent') > 0 THEN cardproc=1;
RUN;

PROC FREQ DATA=all; TABLES cdcod*chd_type; RUN;
```

9.9.7. Dataset processing specifics for Tucson CVD outcome variables

```
DATA status (KEEP=clinic pptid blpsgdate vital censdate prev_mi prev_mip prev_stk mi mip mi_fatal
    mi_date mip_date stroke stk_fatal stk_date chd_death chd_dthdt cvd_death
    cvd_dthdt angina ang_date revasc_proc revpro_date ptca ptca_date cabg cabg_date
    chf prev_chf chf_date any_chd any_cvd)

    event (KEEP=clinic pptid blpsgdate event event_dt stk_type fstk_type);
    SET all;
    BY pptid;
    LENGTH vital censdate prev_mi prev_mip prev_stk mi mip mi_fatal mi_date mip_date stroke stk_fatal stk_date
    chd_death cvd_death angina revasc_proc ptca cabg chf prev_chf any_chd any_cvd 8. event $16. event_dt 8.
    stk_type fstk_type $40.;
    ATTRIB censdate mi_date mip_date stk_date ang_date revpro_date ptca_date cabg_date chf_date chd_dthdt
    cvd_dthdt format=mmddyy8.;

    IF first.pptid THEN DO;
        vital=.;
        censdate=.;
        prev_mi=0;
        prev_mip=0;
        prev_stk=0;
        mi=0;
        mip=0;
        mi_fatal=0;
        mi_date=.;
        mip_date=.;
        stroke=0;
        stk_fatal=0;
        stk_date=.;
        chd_death=0;
        chd_dthdt=.;
        cvd_death=0;
        cvd_dthdt=.;
        angina=0;
        ang_date=.;
        revasc_proc=0;
        revpro_date=.;
        ptca=0;
        ptca_date=.;
        cabg=0;
        cabg_date=.;
        chf=0;
        prev_chf=0;
        chf_date=.;
        any_chd=0;
        any_cvd=0;
    END;
```

9.9.7. Dataset processing specifics for Tucson CVD outcome variables

```
RETAIN vital censdate prev_mi prev_mip prev_stk mi mip mi_date mip_date stroke stk_date mi_fatal stk_fatal
cvd_death cvd_dthdt chd_death chd_dthdt angina ang_date revasc_proc revpro_date ptca ptca_date cabg cabg_date
chf prev_chf chf_date any_chd any_cvd;

/*mi+procedure*/
IF pmi=1 OR psurg=1 THEN prev_mip=prev_mip+1;
IF mi_now=1 AND morbdt <= blpsgdate THEN prev_mip=prev_mip+1;
IF mi_now=1 AND morbdt  > blpsgdate THEN DO;
      mip=mip+1;
      IF mip_date=. OR mip_date > morbdt THEN DO;
            mip_date=morbdt;
      END;
END;

IF mip=0 THEN DO;
      IF chd_type=1 AND cddate  > blpsgdate THEN DO;
            mip=mip+1;
            IF mip_date=. OR mip_date > morbdt THEN DO;
                  mip_date=cddate;
             END;
      END;
END;

othcard=specify_other_card;
i=INDEX(othcard,'PTCA');
ii=INDEX(othcard,'Angio');
iii=INDEX(othcard,'CABG');
IF (other_cardio=1 AND (i NE 0 OR ii NE 0 OR iii NE 0) AND (morbdt <= blpsgdate)) THEN
prev_mip=prev_mip+1;
      IF (other_cardio=1 AND (i NE 0 OR ii NE 0 OR iii NE 0) AND (morbdt > blpsgdate)) THEN DO;
            mip=mip+1;
            IF mip_date=. OR mip_date > morbdt THEN DO;
                  mip_date=morbdt;
            END;
      END;
```

9.9.7. Dataset processing specifics for Tucson CVD outcome variables

```
/*mi*/
IF pmi=1 THEN prev_mi=prev_mi+1;
IF mi_now=1 AND morbdt <= blpsgdate THEN prev_mi=prev_mi+1;
IF mi_now=1 AND morbdt  > blpsgdate THEN DO;
        mi=mi+1;
        IF mi_date=. OR mi_date > morbdt THEN DO;
                mi_date=morbdt;
        END;
        event='MI';
        event_dt=morbdt;
        OUTPUT event;
END;

/*stroke*/
IF pstroke=1 THEN prev_stk=1;
IF strokenow=1 AND morbdt <= blpsgdate THEN prev_stk=1;
IF strokenow=1 AND morbdt  > blpsgdate THEN DO;
        stroke=stroke+1;
        event='STROKE';
        event_dt=morbdt;
        IF stroketype=3 THEN stk_type='UNKNOWN';                                /*3=Unknown */
        IF hemstrokesubtypenow=1 THEN stk_type='SUBARACHNOID HEMORRHAGE';   /*1=Subarachnoid  */
        IF hemstrokesubtypenow=2 THEN stk_type='INTRACEREBRAL HEMORRHAGE'; /*2=Intraparenchymal */
        IF hemstrokesubtypenow=3 THEN stk_type='UNKNOWN';                       /*3=Indeterminate */
        IF ischstrokesubtypenow=1 THEN stk_type='ISCH-LACUNAR';                 /*1=Isch/Lacunar */
        IF ischstrokesubtypenow=2 THEN stk_type='ISCH-CARDIOEMBOLIC';           /*2=Isch/Cardioembolic*/
        IF ischstrokesubtypenow=3 THEN stk_type='ISCH-UNKNOWN';                 /*3=Isch/Unknown */
        IF ischstrokesubtypenow=4 THEN stk_type='ISCH-ATHEROSCLEROTIC';             /*4=Atherosclerotic  */
        IF ischstrokesubtypenow=5 THEN stk_type='ISCH-HEMORRHAGIC';             /*5=Hemorrhagic */
        OUTPUT event;

        IF stk_date=. OR stk_date > morbdt THEN DO;
                stk_date=morbdt;
        END;
END;

LABEL clinic='site code';
LABEL pptid='ID';
LABEL blpsgdate='baseliNE psg date';
LABEL vital='vital status (0=dead, 1=alive)';
LABEL censdate='date of last contact/death';
LABEL prev_mi='mi prior to bl psg (0=no, 1+=yes)';
LABEL prev_stk='stroke prior to bl psg (0=no, 1+=yes)';
LABEL mi='mi since bl psg (0=no, 1+=yes)';
LABEL mi_fatal='fatal mi (0=no, 1=yes)';
LABEL stk_fatal='fatal stroke (0=no, 1=yes)';
LABEL stroke='stroke since bl psg (0=no, 1+=yes)';
```

9.9.7. Dataset processing specifics for Tucson CVD outcome variables

```
LABEL mi_date='date of first mi since bl psg';
LABEL stk_date='date of first stroke since bl psg';
LABEL chd_death='CHD death (0=no, 1=yes)';
LABEL cvd_death='CVD death (0=no, 1=yes)';
LABEL event='event';
LABEL event_dt='date of event';
LABEL stk_type='type of stroke';
LABEL angina='Angina pectoris (0=no, 1+=yes)';
LABEL ang_date='date of angina pectoris';
LABEL revasc_proc='Revascularization procedures (0=no, 1+=yes)';
LABEL revpro_date='date of revascularization procedure';
LABEL ptca='PTCA (0=no, 1+=yes)';
LABEL ptca_date='date of PTCA';
LABEL cabg='CABG (0=no, 1+=yes)';
LABEL cabg_date='date of CABG';
LABEL event='event';
LABEL event_dt='date of event';
LABEL stk_type='type of stroke';
LABEL fstk_type='type of fatal stroke';
LABEL chf='CHF (0=no, 1=yes)';
LABEL prev_chf='CHF prior to bl psg (0=no, 1+=yes)';
LABEL any_chd='CHD event since bl psg (0=no, 1=yes)';
LABEL any_cvd='CVD event since bl psg (0=no, 1=yes)';

IF mi=0 THEN DO;
    IF chd_type=1 THEN DO;
        mi=mi+1;
        mi_date=cddate;
        mi_fatal=1;
        IF mi_fatal=1 THEN censdate=mi_date;
        event='MI';
        event_dt=morbdt;
        OUTPUT event;
    END;
END;

IF stroke=0 THEN DO;
    IF classification=1 THEN DO;
        stroke=stroke+1;
        stk_date=sddate;
        stk_fatal=1;
        IF stk_fatal=1 THEN censdate=stk_date;
        event='STROKE';
        event_dt=sddate;
        IF stroketype=3 THEN fstk_type='UNKNOWN';
        IF hemstrokesubtype=1 THEN fstk_type='SUBARACHNOID HEMORRHAGE';
        IF hemstrokesubtype=2 THEN fstk_type='INTRACEREBRAL HEMORRHAGE';/*2=Intraparenchymal*/
```

```
            IF hemstrokesubtype=3 THEN fstk_type='UNKNOWN';
            IF ischstrokesubtype=1 THEN fstk_type='ISCH-LACUNAR'
            IF ischstrokesubtype=2 THEN fstk_type='ISCH-CARDIOEMBOLIC';          /*2=Isch/Cardioembolic*/
            IF ischstrokesubtype=3 THEN fstk_type='ISCH-UNKNOWN'
            IF ischstrokesubtype=4 THEN fstk_type='ISCH-ATHEROSCLEROTIC';
            IF ischstrokesubtype=5 THEN fstk_type='ISCH-HEMORRHAGIC';
            OUTPUT event;
        END;
    END;

    IF pchf=1 THEN prev_chf=1;
    IF (chf_now=1 AND (morbdt <= blpsgdate)) THEN prev_chf=1;
    IF (chf_now=1 AND (morbdt > blpsgdate)) THEN DO;
        chf=chf+1;
        event='CHF';
        event_dt=morbdt;
        OUTPUT event;

        IF chf_date=. OR chf_date > morbdt THEN DO;
            chf_date=morbdt;
        END;
    END;

    IF (cdcod IN (1 2 3 4) AND cddate > blpsgdate) THEN DO;
        cvd_death=1;
        cvd_dthdt=cddate;
    END;

    IF (cdcod IN (1) AND (chd_type=1 OR chd_type=2) AND cddate > blpsgdate) THEN DO;
        chd_death=1;
        chd_dthdt=cddate;
    END;

    othcard=specify_other_card;
    i=INDEX(othcard,'PTCA');
    ii=INDEX(othcard,'Angio');
    iii=INDEX(othcard,'CABG');
    IF ((mi_now=1 OR chd_type=1) AND morbdt > blpsgdate) THEN any_chd=1; /* MI */
    IF (other_cardio=1 AND (i NE 0 OR ii NE 0 OR iii NE 0) AND (morbdt > blpsgdate)) THEN any_chd=1;
    IF ((cdcod in (1) AND (chd_type=1 OR chd_type=2)) AND cddate > blpsgdate) THEN any_chd=1;

    IF ((mi_now=1 OR chd_type=1) AND morbdt > blpsgdate) THEN any_cvd=1;
    IF (other_cardio=1 AND (i NE 0 OR ii NE 0 OR iii NE 0) AND (morbdt > blpsgdate)) THEN any_cvd=1;
    IF (chf_now=1 AND (morbdt > blpsgdate)) THEN any_cvd=1;
    IF ((cdcod in (1) AND (chd_type=1 OR chd_type=2)) AND cddate > blpsgdate) THEN any_cvd=1;
    IF (cdcod in (1 2 3 4) AND (cddate > blpsgdate)) THEN any_cvd=1;
```

9.9.7. Dataset processing specifics for Tucson CVD outcome variables

```
/* Angina */
IF (other_cardio=1 AND specify_other_card in('angina')) AND (morbdt > blpsgdate) THEN DO;
	angina=angina+1;
	IF ang_date=. OR (ang_date > morbdt) THEN DO;
		ang_date=morbdt;
	END;
	event='ANGINA';
	event_dt=morbdt;
	OUTPUT event;
END;

/* Revascularization procedures */
othcard=specify_other_card;
i=INDEX(othcard,'PTCA');
ii=INDEX(othcard,'Angio');
iii=INDEX(othcard,'CABG');
IF (other_cardio=1 AND (i NE 0 OR ii NE 0 OR iii NE 0) AND (morbdt > blpsgdate)) THEN DO;
	revasc_proc=revasc_proc+1;
	IF revpro_date=. OR revpro_date > morbdt THEN DO;
		revpro_date=morbdt;
	END;
	event='REVASC PROCEDURE';
	event_dt=morbdt;
	OUTPUT event;
END;

/* PTCA */
i=INDEX(othcard,'PTCA');
ii=INDEX(othcard,'Angio');
IF (other_cardio=1 AND (i NE 0 OR ii NE 0)) AND (morbdt > blpsgdate) THEN DO;
	ptca=ptca+1;
	IF ptca_date=. OR ptca_date > morbdt THEN DO;
		ptca_date=morbdt;
	END;
	event='PTCA';
	event_dt=morbdt;
	OUTPUT event;
END;
```

9.9.7. Dataset processing specifics for Tucson CVD outcome variables

```
    /* CABG */
    i=INDEX(othcard,'CABG');
    IF (other_cardio=1 AND i NE 0) AND (morbdt > blpsgdate) THEN DO;
        cabg=cabg+1;
        IF cabg_date=. OR cabg_date > morbdt THEN DO;
            cabg_date=morbdt;
        END;
        event='CABG';
        event_dt=morbdt;
        OUTPUT event;
    END;

    enddt=MDY(4,1,2006);
    IF last.pptid THEN DO;
        IF (status=1) THEN DO;
            vital=0;
            censdate=DATEPART(date);
        END;
        IF status NE 1 THEN DO;
            vital=1;
            censdate=DATEPART(date);
            IF censdate > enddt THEN censdate=enddt;
        END;

        IF pptid="3156701" THEN DO;
            vital=1;
            censdate=enddt;
        END;

        IF pptid="3311301" THEN DO;
            vital=1;
            censdate=enddt;
        END;

        IF pptid="1228502" THEN censdate=cddate;
        IF pptid="1310202" THEN censdate=cddate;
        IF pptid="2370401" THEN censdate=sddate;
        OUTPUT status;
    END;
RUN;

DATA data.tucson_status;
    SET status;
RUN;
```

9.9.7. Dataset processing specifics for Tucson CVD outcome variables

```
DATA data.tucson_event;
	SET event;
	IF event NE 'STROKE' THEN DO;
		stk_type="";
		fstk_type="";
	END;
RUN;
```