```
*******************************************************************************
** Created: 2014-SEPT-18, Version 2.1
Modified: 2015-MAY-26; a semicolon that was erroneously excluded from the initial 160
minute sleep macro has been replaced (see page 5, highlighted)
** Project: Identifying children's nocturnal sleep using 24-hour waist accelerometry
Barreira, T.V., Schuna Jr., J.M., Mire, E.F., Katzmarzyk, P.T., Chaput, J-P., Leduc, G., & Tudor-
Locke, C. Identifying children's nocturnal sleep using 24-hour waist accelerometry. Medicine and
Science in Sports and Exercise, in press.
** Coding: Emily F. Mire and John M. Schuna Jr., Pennington Biomedical Research Center,
2013
** Source Code: a commonly used and publicly available non-wear algorithm developed by
the National Cancer Institute
(http://riskfactor.cancer.gov/tools/nhanes_pam/); current code is a refinement and
extension of a previously published algorithm available at
http://www.pbrc.edu/pdf/PBRCSleepPeriodTimeMacroCode.pdf and described in Tudor-Locke C,
Barreira TV, Schuna JM Jr, Mire EF, Katzmarzyk PT. Fully automated waist-worn accelerometer
algorithm for detecting children's sleep-period time separate from 24-h physical activity or
sedentary behaviors. Appl Physiol Nutr Metab. 2014 Jan;39(1):53-7.
** The program included here will allow users to label each minute in an accelerometer
file as a waking, sleeping or non-wear minute, determine nocturnal sleep onset and offset
times and calculate total sleep episode time. USERDATA must be a compilation of
participants' accelerometer data processed in one minute epochs. The first day of data
for each participant must begin at midnight - data management code preempts the total
sleep episode time algorithm below to adjust any files from devices that were not
initialized at midnight. The algorithm was developed using data from the ActiGraph GT3X+,
but could be applied to other Actigraph devices that output the variable types below
(must be included in the dataset for the processes to work):
-Unique subject identification number (must be named PID)
-Data Time Stamp - variable containing the date and time for each minute (must be split
into separate variables called DATE and ACCELTIME)
- Axis 1 - Intensity value from the vertical axis for the minute (must be re-named
PAXINTEN)
-INCLINOMETER = indicator for position of accelerometer at time of output (0 = Off, 1 =
Standing, 2 = Lying down, 3 = Sitting); may be output as a single variable or as 4
variables, one for each position
*******************************************************************************
```

```
/*Check to see if the first minute of the first day for each subject is anything other
than midnight (i.e. did not follow protocol to initialize device at midnight) and remove
those dates; PROGRAM WILL NOT FUNCTION AS WRITTEN IF FIRST DAY DOES NOT BEGIN AT
MIDNIGHT*/
data initial;
      set USERDATA;
      by pid;
      if first.pid and acceltime ne '0:00:00't;
run;
data PAX1;
      merge initial(in=in_invalid) USERDATA;
      by pid date;
      if in_invalid  then delete;
run;
```

```
/*Deleting last day - all have < 24 hours and are not full days because device was
returned and stopped*/
data lastday;
set PAX1;
by pid;
if last.pid;
keep pid date;
run;
data pax1c;
merge pax1 lastday(in=last);
by pid date;
if last then delete;
run;



/*DELETION COMPLETE - Creating variables to identify each minute and label each midnight-
to-midnight and noon-to-noon day*/
data PAX2;
      PAXN+1;
      set PAX1C;
      by pid;
      if first.pid then PAXN=1; /*PAXN = Sequential observation number for each minute*/
run;
data PAX3;
  set PAX2;
      Noon = round(PAXN/1440);/*Sequential noon-to-noon day; WILL RUN FROM 11:59 TO
11:58; MIDNIGHT TO NOON ON FIRST DAY OF INITIALIZATION WILL RETURN 0 */
      Day=ceil(paxn/1440); /*DAY = Sequential midnight-to-midnight day*/
  label day='Sequential Day';
       PAXDAY=weekday(date); /*PAXDAY = Day of the week*/
  format paxday wkday.;
  label paxday = "Day of the Week";
run;


/*If accelerometer software outputs 4 inclinometer variables, then the following code
will also need to be run:
DATA PAX3;
SET PAX3;
ARRAY nums [*] InclineOFF InclineStanding InclineSitting InclineLying;
Highest=MAX(InclineOFF, InclineStanding, InclineSitting, InclineLying);
DO i=1 TO DIM(nums);
IF nums[i]= Highest THEN Inc=VNAME(nums[i]);
if inc = "InclineOff" then Inclinometer = 0;
if inc = "InclineStanding" then Inclinometer = 1;
if inc = "InclineLying" then Inclinometer = 2;
if inc = "InclineSitting" then Inclinometer = 3;
drop i Highest INC inclineoff inclinelying inclinestanding inclinesitting;
END;
RUN;*/
```

```
*------------ Calculating variables for Sadeh's algorithm -------------*;

data setvar;
      set pax3;
      by pid;

      /*Natural log of current minute*/
      LogAct = log(paxinten+1);

      /*Indicates if current minute intensity between 50-99, to be totaled later*/
      if paxinten ge 50 and paxinten < 100 then SedInd = 1; else if paxinten = . then
sedind = .; else sedind = 0;

      /*Rolling STD of current minute and 5 preceding*/
      STDpre5 = std(paxinten, lag(paxinten), lag2(paxinten), lag3(paxinten),
lag4(paxinten), lag5(paxinten));

      if Paxn < 6 then STDpre5 = .;
run;

/*Rolling mean of previous and next 5 minutes and total count of minutes with intensity
between 50 and 100*/
proc expand data = setvar OUT = rollmean;
      convert paxinten = MeanW5 / method = none TRANSFORMOUT = (cmovave 11);
      by pid;
run;
data rollmean2;
      set rollmean;
      format meanw5 8.2;
      drop time;
run;
proc expand data = rollmean2 out = countout;
      convert sedind = sumind / method = none transformout = (cmovsum 11);
      by pid;
run;

/*Set mean and sum of first 5 and last 5 of total wear time to missing since there are no
future minutes to use in calculation*/
data lastmin;
      set pax3;
      by pid;
      if last.pid then do;
      lastpax = paxn;
      output;
      end;
      keep pid lastpax;
run;
data sets;
      merge lastmin countout;
      by pid;
      if (paxn > (lastpax - 5)) or (paxn < 6) then do;
      meanw5 = .; sumind = .; end;
run;


/*Create a file with minute by minute information from each PID including Sadeh sleeping
classification*/

data libname.pampermin;
      set sets;
```

```
      ProbSleep = 7.601 - (0.065*MeanW5) - (1.08*sumind) - (0.056*STDpre5) -
(0.703*logact);
      if probsleep = . THEN DO; PROBSLEEP = 1; END;
      if probsleep < 0 and probsleep ne . then SleepIncOff = 0; if (probsleep ge 0) or
inclinometer = 0 then SleepIncOff = 1;
      drop lastpax time;
run;



/*160-minute macro to isolate sleep episodes (at least 160 minutes of Sadeh-defined sleep
minutes, beginning with 5 consecutive minutes of sleep, ending with 10 or 20 consecutive
minutes of awake)*/

data monitors;
      set libname.pampermin;
      by pid;
run;
proc sort data=monitors;
      by pid noon paxn;
run;

%macro nw(nwperiod=);
data nw_all;
      set monitors;
      by pid NOON paxn;

      if first.NOON then nw_num=0;    /*SLEEP period number*/

      if first.NOON or reset or stopped then do;
      strt_nw=0;      /*starting minute for the SLEEP period*/
      end_nw=0;       /*ending minute for the SLEEP period*/
      start=0;        /*indicator for starting to count the SLEEP period*/
      dur_nw=0;       /*duration for the SLEEP period*/
      reset=0;        /*indicator for resetting and starting over*/
      stopped=0;      /*indicator for stopping the SLEEP period*/
      cnt_non_zero=0; /*counter for the number of minutes with SLEEP = 0*/
            BEDTIME = 0;
            END_MIN = 0;            /*LITERAL ENDING MINUTE (TIME THEY WOKE UP)*/
            CNT_SLEEP = 0;    /*COUNTER FOR NUMBER OF MINUTES SLEEPING*/
      end;
      retain nw_num strt_nw end_nw stopped reset start cnt_non_zero dur_nw END_MIN
CNT_SLEEP BEDTIME;

/*ADDING in a counter for sleep minutes so that it will start a sleep period*/
      if SLEEPINCOFF = 1 then
    cnt_sleep=cnt_sleep+1;

/*before reaching the 5 consecutive minutes of sleep, if encounter one minute with wake,
reset the counter*/
      if SLEEPINCOFF = 0 then do; cnt_sleep=0; end;

/*The SLEEP period starts with 5 CONSECUTIVE MINS OF SLEEP*/
      if start=0  and cnt_sleep ge 5 then do;
        strt_nw=paxn-4;               /*assign the starting minute of SLEEP*/
        start=1;
            BEDTIME = ACCELTIME - 240; /*4 MINUTES PRIOR*/
  end;
      FORMAT BEDTIME TIME8.2;

/*accumulate the number of SLEEP minutes*/
      if start and SLEEPINCOFF = 1 then do;
      end_nw=paxn;         /*keep track of the end of the SLEEP period*/
```

```
            END_MIN=ACCELTIME; END;        /*keep track of the LITERAL TIME FOR THE END
OF THE SLEEP period*/
      FORMAT END_MIN TIME8.2;

/*keep track of the number of AWAKE MINUTES*/
      if SLEEPINCOFF = 0 then
    cnt_non_zero=cnt_non_zero+1;

/*before reaching the 10 consecutive minutes of AWAKE, if encounter one minute with
SLEEP, reset the counter*/
      if SLEEPINCOFF = 1 then do; cnt_non_zero=0; end;

/*duration of SLEEP period*/
      dur_nw=end_nw-strt_nw+1;

/*A SLEEP period ends with 10 consecutive minutes of SLEEP = 0 (AWAKE) after 5 am and 20
mins before 5 am*/
      if ((ACCELTIME ge '19:00:00't) or (ACCELTIME < '05:00:00't)) AND (DUR_NW GE 160)
and (cnt_non_zero >20) then do;
    if dur_nw<&nwperiod then reset=1;
    else stopped=1;
      end;
      if ((ACCELTIME ge '19:00:00't) or (ACCELTIME < '05:00:00't)) AND ((DUR_NW) < 160)
and (cnt_non_zero >10) then do;
    if dur_nw<&nwperiod then reset=1;
    else stopped=1;
      end;
      if ((ACCELTIME < '19:00:00't) and (ACCELTIME ge '05:00:00't)) and (cnt_non_zero>10)
then do;
    if dur_nw<&nwperiod then reset=1;
    else stopped=1;
      end;

/*last minute of the day*/
      if last.NOON and dur_nw>=&nwperiod then stopped=1;

/*output one record for each SLEEP period*/
      if stopped=1 then do;
       nw_num=nw_num+1;
       keep pid NOON PAXDAY nw_num strt_nw end_nw dur_nw END_MIN BEDTIME;
       output;
  end;
run;


%mend nw;

%nw(nwperiod=160);

/*Correcting negative bedtimes due to kids going to bed between 0:00 - 0:03 and macro
subtracting 4 minutes from the 5th minute of sleep start time*/
data fixbed;/*4861*/
      set nw_all;
            if bedtime = '-0:04:00't then do; bedtime = '0:00:00't; end;
            if bedtime = '-0:03:00't then do; bedtime = '0:01:00't; end;
            if bedtime = '-0:02:00't then do; bedtime = '0:02:00't; end;
            if bedtime = '-0:01:00't then do; bedtime = '0:03:00't; end;
RUN;


/*IDENTIFY WAKE TIME, BED TIME AND SLEEP TIME BY ONLY CONSIDERING NOON DAYS 1-7 AND
NOCTURNAL BED TIMES, AND COMBINING SLEEP PERIODS THAT ARE < 20 MINUTES APART */
```

```
/*Analyze noon days with only one sleep period.*/
data ind1;
      set fixbed;
      by pid noon;
            if first.noon and last.noon;
            if noon > 0 and noon le 7;
run;
data newsleep;
      set ind1;
      by pid noon;
            Bed1 = bedtime; Wake1 = end_min;
      format wake1 time8.2 bed1 time8.2;
      drop bedtime end_min strt_nw end_nw dur_nw nw_num;
run;

data sleeptotal1;
set newsleep;
      if (bed1 ge '19:00:00't) or (bed1 < '06:00:00't) then do;
            if wake1 < bed1 then SleepNWmins = ((wake1 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                  else SleepNWmins = ((wake1-bed1) + '00:01:00't)/60;
            end;
      else do;
            SleepNWmins = 0;
      end;
if SleepNWmins = . then do; SleepNWmins = 0; end;
run;


/*Identify noon days with > 1 sleep period */
data monmult;
      set fixbed;
      by pid noon;
      if last.noon and nw_num > 1;
      Mult = 1;
      keep pid noon mult;
run;
data ind;
      merge fixbed monmult;
      by pid noon;
      if mult = 1 and noon > 0 and noon le 7;
      drop mult;
run;


/*Analyze noon days with TWO sleep periods*/
data ind2;
      set ind;
      by pid noon;
      if first.noon or (last.noon and nw_num = 2);
run;
data ind22;
      set ind2;
      by pid noon;
      if first.noon and last.noon then delete;
run;
data newsleep2;
      set ind22;
      by pid noon;
      if first.noon then do; B1 = Bedtime; W1 = end_min; end;
            else do; B2 = bedtime; W2 = end_min; end;
```

```
      format w1 time8.2 b1 time8.2 w2 time8.2 b2 time8.2;
run;
proc summary data=newsleep2 ;
      by pid NOON;
      var b1 w1 b2 w2 ;
      output out=sumtimes
       sum = Bed1 Wake1 Bed2 Wake2;
run;
data diff;
      set sumtimes;
      if bed2 < wake1 then Diff = (('24:00:00't + bed2) - wake1)/60;
      else Diff = (bed2 - wake1)/60;
run;

data sleeptotal2;
  set diff;
      if diff < 20 then do;

            if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge '19:00:00't)
or (bed2 < '06:00:00't)) then do;
                  if wake2 < bed1 then SleepNWmins = ((wake2 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                        else SleepNWmins = ((wake2-bed1) + '00:01:00't)/60;
                  end;

            if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ('06:00:00't<= bed2 <
'19:00:00't) then do;
                  if wake1 < bed1 then SleepNWmins = ((wake1 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                        else SleepNWmins = ((wake1-bed1) + '00:01:00't)/60;
                  end;

            if ('06:00:00't <= bed1 < '19:00:00't) and ((bed2 ge '19:00:00't) or (bed2 <
'06:00:00't)) then do;
                  if wake2 < bed2 then SleepNWmins = ((wake2 + ('24:00:00't-
bed2))+'00:01:00't)/60;
                        else SleepNWmins = ((wake2-bed2) + '00:01:00't)/60;
                  end;

            if ('06:00:00't <= bed1 < '19:00:00't) and ('06:00:00't <= bed2 <
'19:00:00't) then SleepNWmins = 0;
      end;

      else do;

            if (bed1 ge '19:00:00't) or (bed1 < '06:00:00't) then do;
                  if wake1 < bed1 then Sleepbed1 = ((wake1 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                        else Sleepbed1 = ((wake1-bed1) + '00:01:00't)/60;
                  end;
            else do Sleepbed1 = 0;
                  end;

            if (bed2 ge '19:00:00't) or (bed2 < '06:00:00't) then do;
                  if wake2 < bed2 then Sleepbed2 = ((wake2 + ('24:00:00't-
bed2))+'00:01:00't)/60;
                        else Sleepbed2 = ((wake2-bed2) + '00:01:00't)/60;
                  end;
            else do Sleepbed2 = 0;
                  end;

            SleepNWmins = Sleepbed1 + Sleepbed2;
      end;
```

```
  if SleepNWmins= . then do; SleepNWmins= 0; end;
  drop _type_ _freq_ Sleepbed1 Sleepbed2;
run;


/*Analyze noon days with THREE sleep periods*/
data three;
      set ind;
      by pid noon;
      if last.noon and nw_num =3;
      Mult = 3;
      keep pid noon mult;
run;
data ind3;
      merge ind three;
      by pid noon;
      if mult = 3;
      drop mult;
run;
data newsleep3;
      set ind3;
            if nw_num = 1 then do; B1 = Bedtime; W1 = end_min; end;
            if nw_num = 2 then do; B2 = bedtime; W2 = end_min; end;
            if nw_num = 3 then do; B3 = bedtime; W3 = end_min; end;
      format w1 time8.2 b1 time8.2 w2 time8.2 b2 time8.2 w3 time8.2 b3 time8.2;
run;
proc summary data=newsleep3 ;
      by pid NOON;
      var b1 w1 b2 w2 b3 w3;
      output out=sumtimes3
    sum = Bed1 Wake1 Bed2 Wake2 Bed3 Wake3;
run;
data diff3;
      set sumtimes3;
            if bed2 < wake1 then Diff12 = (('24:00:00't + bed2) - wake1 +'00:01:00't)/60;
                  else Diff12 = (bed2 - wake1 +'00:01:00't)/60;
            if bed3 < wake2 then Diff23 = (('24:00:00't + bed3) - wake2 +'00:01:00't)/60;
                  else Diff23 = (bed3 - wake2 +'00:01:00't)/60;
run;

data sleeptotal3;
  set diff3;
            if diff12 < 20 and diff23 < 20 then do;
                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't))
then do;
                        if wake3 < bed1 then SleepNWmins = ((wake3 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                              else SleepNWmins = ((wake3-bed1) + '00:01:00't)/60;
                        end;

                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ('06:00:00't<=
bed2 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if wake1 < bed1 then SleepNWmins = ((wake1 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                              else SleepNWmins = ((wake1-bed1) + '00:01:00't)/60;
                        end;

                  if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't<=
bed1 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if wake2 < bed2 then SleepNWmins = ((wake2 + ('24:00:00't-
bed2))+'00:01:00't)/60;
                              else SleepNWmins = ((wake2-bed2) + '00:01:00't)/60;
```

```
                        end;

                    if ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't<=
bed1 < '19:00:00't) and ('06:00:00't <= bed2 < '19:00:00't) then do;
                        if wake3 < bed3 then SleepNWmins = ((wake3 + ('24:00:00't-
bed3))+'00:01:00't)/60;
                                else SleepNWmins = ((wake3-bed3) + '00:01:00't)/60;
                        end;

                    if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if wake2 < bed1 then SleepNWmins = ((wake2 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                                else SleepNWmins = ((wake2-bed1) + '00:01:00't)/60;
                        end;

                    if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge
'19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't <= bed1 < '19:00:00't) then do;
                        if wake3 < bed2 then SleepNWmins = ((wake3 + ('24:00:00't-
bed2))+'00:01:00't)/60;
                                else SleepNWmins = ((wake3-bed2) + '00:01:00't)/60;
                        end;

                    if ('06:00:00't <= bed1 < '19:00:00't) and ('06:00:00't <= bed2 <
'19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then SleepNWmins = 0;
                end;

                if diff12 < 20 and diff23 ge 20 then do;
                    if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't))
then do;
                        if wake2 < bed1 then Sleepbed1 = ((wake2 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                                else Sleepbed1 = ((wake2-bed1) + '00:01:00't)/60;
                        if wake3 < bed3 then Sleepbed2 = ((wake3 + ('24:00:00't-
bed3))+'00:01:00't)/60;
                                else Sleepbed2 = ((wake3-bed3) + '00:01:00't)/60;
                        SleepNWmins = Sleepbed1 + Sleepbed2;
                    end;

                    if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ('06:00:00't<=
bed2 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if wake1 < bed1 then SleepNWmins = ((wake1 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                                else SleepNWmins = ((wake1-bed1) + '00:01:00't)/60;
                        end;

                    if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't<=
bed1 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if wake2 < bed2 then SleepNWmins = ((wake2 + ('24:00:00't-
bed2))+'00:01:00't)/60;
                                else SleepNWmins = ((wake2-bed2) + '00:01:00't)/60;
                        end;

                    if ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't<=
bed1 < '19:00:00't) and ('06:00:00't <= bed2 < '19:00:00't) then do;
                        if wake3 < bed3 then SleepNWmins = ((wake3 + ('24:00:00't-
bed3))+'00:01:00't)/60;
                                else SleepNWmins = ((wake3-bed3) + '00:01:00't)/60;
                        end;

                    if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't <= bed3 < '19:00:00't) then do;
```

```
                        if wake2 < bed1 then SleepNWmins = ((wake2 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                              else SleepNWmins = ((wake2-bed1) + '00:01:00't)/60;
                        end;

                  if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge
'19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't <= bed1 < '19:00:00't) then do;
                        if wake2 < bed2 then Sleepbed1 = ((wake2 + ('24:00:00't-
bed2))+'00:01:00't)/60;
                              else Sleepbed1 = ((wake2-bed2) + '00:01:00't)/60;
                        if wake3 < bed3 then Sleepbed2 = ((wake3 + ('24:00:00't-
bed3))+'00:01:00't)/60;
                              else Sleepbed2 = ((wake3-bed3) + '00:01:00't)/60;
                        SleepNWmins = Sleepbed1 + Sleepbed2;
                  end;

                  if ('06:00:00't <= bed1 < '19:00:00't) and ('06:00:00't <= bed2 <
'19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then SleepNWmins = 0;
            end;

            if diff12 ge 20 and diff23 < 20 then do;
                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't))
then do;
                        if wake1 < bed1 then Sleepbed1 = ((wake1 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                              else Sleepbed1 = ((wake1-bed1) + '00:01:00't)/60;
                        if wake3 < bed2 then Sleepbed2 = ((wake3 + ('24:00:00't-
bed2))+'00:01:00't)/60;
                              else Sleepbed2 = ((wake3-bed2) + '00:01:00't)/60;
                        SleepNWmins = Sleepbed1 + Sleepbed2;
                  end;

                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ('06:00:00't<=
bed2 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if wake1 < bed1 then SleepNWmins = ((wake1 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                              else SleepNWmins = ((wake1-bed1) + '00:01:00't)/60;
                        end;

                  if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't<=
bed1 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if wake2 < bed2 then SleepNWmins = ((wake2 + ('24:00:00't-
bed2))+'00:01:00't)/60;
                              else SleepNWmins = ((wake2-bed2) + '00:01:00't)/60;
                        end;

                  if ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't<=
bed1 < '19:00:00't) and ('06:00:00't <= bed2 < '19:00:00't) then do;
                        if wake3 < bed3 then SleepNWmins = ((wake3 + ('24:00:00't-
bed3))+'00:01:00't)/60;
                              else SleepNWmins = ((wake3-bed3) + '00:01:00't)/60;
                        end;

                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if wake1 < bed1 then Sleepbed1 = ((wake1 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                              else Sleepbed1 = ((wake1-bed1) + '00:01:00't)/60;
                        if wake2 < bed2 then Sleepbed2 = ((wake2 + ('24:00:00't-
bed2))+'00:01:00't)/60;
                              else Sleepbed2 = ((wake2-bed2) + '00:01:00't)/60;
                        SleepNWmins = Sleepbed1 + Sleepbed2;
```

```
                  end;

                  if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge
'19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't <= bed1 < '19:00:00't) then do;
                        if wake3 < bed2 then SleepNWmins = ((wake3 + ('24:00:00't-
bed2))+'00:01:00't)/60;
                              else SleepNWmins = ((wake3-bed2) + '00:01:00't)/60;
                  end;

                  if ('06:00:00't <= bed1 < '19:00:00't) and ('06:00:00't <= bed2 <
'19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then SleepNWmins = 0;
            end;

            if diff12 ge 20 and diff23 ge 20 then do;
                  if (bed1 ge '19:00:00't) or (bed1 < '06:00:00't) then do;
                        if wake1 < bed1 then Sleepbed1 = ((wake1 + ('24:00:00't-
bed1))+'00:01:00't)/60;
                              else Sleepbed1 = ((wake1-bed1) + '00:01:00't)/60;
                        end;
                  else do Sleepbed1 = 0;
                  end;

                  if (bed2 ge '19:00:00't) or (bed2 < '06:00:00't) then do;
                        if wake2 < bed2 then Sleepbed2 = ((wake2 + ('24:00:00't-
bed2))+'00:01:00't)/60;
                              else Sleepbed2 = ((wake2-bed2) + '00:01:00't)/60;
                        end;
                  else do Sleepbed2 = 0;
                  end;

                  if (bed3 ge '19:00:00't) or (bed3 < '06:00:00't) then do;
                        if wake3 < bed3 then Sleepbed3 = ((wake3 + ('24:00:00't-
bed3))+'00:01:00't)/60;
                              else Sleepbed3 = ((wake3-bed3) + '00:01:00't)/60;
                        end;
                  else do Sleepbed3 = 0;
                  end;

                  SleepNWmins = Sleepbed1 + Sleepbed2 + Sleepbed3;
            end;
      if SleepNWmins= . then do; SleepNWmins= 0; end;
      drop _type_ _freq_ Sleepbed1 Sleepbed2 Sleepbed3;
run;


/*RUN A CHECK TO SEE IF THERE ARE ANY SUBJECTS WITH 4 SLEEP PERIODS IN A NOON-TO-NOON DAY
(NW_NUM = 4). THESE WERE HANDLED ON A CASE-BY-CASE BASIS SPECIFIC TO PID, EXAMPLE
COMMENTED OUT BELOW*/
data count;
      set fixbed;
            if noon > 0 and noon le 7;
run;
proc freq data=count;
      tables nw_num;
run;


/*data four;
      set ind;
      by pid noon;
      if last.noon and nw_num =4;
      Mult = 4;
      keep pid noon mult;
```

```
run;
data ind4;
      merge ind four;
      by pid noon;
      if mult = 4;
      drop mult;
run;
data newsleep4;
      set ind4;
            if nw_num = 1 then do; B1 = Bedtime; W1 = end_min; end;
            if nw_num = 2 then do; B2 = bedtime; W2 = end_min; end;
            if nw_num = 3 then do; B3 = bedtime; W3 = end_min; end;
            if nw_num = 4 then do; B4 = bedtime; W4 = end_min; end;
            format w1 time8.2 b1 time8.2 w2 time8.2 b2 time8.2 w3 time8.2 b3 time8.2 w4
time8.2 b4 time8.2;
run;
proc summary data=newsleep4 ;
      by pid NOON;
      var b1 w1 b2 w2 b3 w3 b4 w4 ;
      output out=sumtimes4
    sum = Bed1 Wake1 Bed2 Wake2 Bed3 Wake3 Bed4 Wake4;
run;
data diff4;
      set sumtimes4;
            if bed2 < wake1 then Diff12 = (('24:00:00't + bed2) - wake1 +'00:01:00't)/60;
                  else Diff12 = (bed2 - wake1 +'00:01:00't)/60;
            if bed3 < wake2 then Diff23 = (('24:00:00't + bed3) - wake2 +'00:01:00't)/60;
                  else Diff23 = (bed3 - wake2 +'00:01:00't)/60;
            if bed4 < wake3 then Diff34 = (('24:00:00't + bed4) - wake3 +'00:01:00't)/60;
                  else Diff34 = (bed4 - wake3 +'00:01:00't)/60;
run;
/*EXAMPLE!
data sleeptotal4;
      set diff4;
            if pid = 99999 then do;
                  Sleepmins1 = 0; /*BED TIME NOT WITHIN THE 7 PM-6AM WINDOW
                  Sleepmins2 = 0; /*BED TIME NOT WITHIN THE 7 PM-6AM WINDOW
                  SleepMins3 = ((wake3 + ('24:00:00't-bed3))+'00:01:00't)/60;
                  /*DIFFERENCE BETWEEN WAKE 3 AND BED 4 WAS > 20 MINS
                  SleepMins4 = ((wake4-bed4) + '00:01:00't)/60;
                  SleepNWmins = sleepmins1 + sleepmins2 + sleepmins3 + sleepmins4;
            end;
      if SleepNWmins= . then do; SleepNWmins= 0; end;
      drop _type_ _freq_ Sleepmins1 Sleepmins2 Sleepmins3 Sleepmins4;
run;*/


/*Combining all sleep results and assigning sleep/NW indicator to each minute (INCLUSION
OF ANALYSIS WHEN 4 SLEEP PERIODS ARE PRESENT IS COMMENTED OUT*/

data sleeptotals;
      set sleeptotal1 sleeptotal2 sleeptotal3 /*sleeptotal4*/;
DROP PAXDAY;
proc sort;
by pid noon;
run;

data reintegrate;
      merge monitors sleeptotals;
      by pid noon;

/*#############################################################################*/
```

```
/*1 sleep period*/
            if bed2 = . then do;
                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) then do;
                        if (wake1 < bed1) then do;
                              if acceltime ge bed1 OR acceltime le wake1 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                        else do;
                              if acceltime ge bed1 and acceltime le wake1 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                  end;
                  else do;
                        AsleepNW = 0;
                  end;
            end;


/*##########################################################################################
##########################################################################################*/

/*2 sleep periods*/
            if diff < 20 and diff ne . then do;
                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) then do;
                        if wake2 < bed1 then do;
                              if acceltime ge bed1 OR acceltime le wake2 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                        else do;
                              if acceltime ge bed1 and acceltime le wake2 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                  end;

                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ('06:00:00't <=
bed2 < '19:00:00't) then do;
                        if wake1 < bed1 then do;
                              if acceltime ge bed1 or acceltime le wake1 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                        else do;
                              if acceltime ge bed1 and acceltime le wake1 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                  end;

                  if ('06:00:00't <= bed1 < '19:00:00't) and ((bed2 ge '19:00:00't) or
(bed2 < '06:00:00't)) then do;
                        if wake2 < bed2 then do;
                              if acceltime ge bed2 or acceltime le wake2 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                        else do;
                              if acceltime ge bed2 and acceltime le wake2 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                  end;

                  if ('06:00:00't <= bed1 < '19:00:00't) and ('06:00:00't <= bed2 <
'19:00:00't) then do;
                        AsleepNW = 0;
                  end;
```

```
            end;

            if diff ge 20 then do;
                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) then do;
                        if (wake1 < bed1 and (acceltime ge bed1 OR acceltime le wake1) or
(wake1 ge bed1 and acceltime ge bed1 and acceltime le wake1) or
                        (wake2 < bed2 and (acceltime ge bed2 OR acceltime le wake2)) or
(wake2 ge bed2 and acceltime ge bed2 and acceltime le wake2)) then AsleepNW = 1; else
AsleepNW = 0;
                  end;

                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ('06:00:00't <=
bed2 < '19:00:00't) then do;
                        if (wake1 < bed1 and (acceltime ge bed1 or acceltime le wake1) or
(wake1 ge bed1 and acceltime ge bed1 and acceltime le wake1)) then AsleepNW = 1; else
AsleepNW = 0;
                  end;

                  if ('06:00:00't <= bed1 < '19:00:00't) and ((bed2 ge '19:00:00't) or
(bed2 < '06:00:00't)) then do;
                        if (wake2 < bed2 and (acceltime ge bed2 or acceltime le wake2) or
(wake2 ge bed2 and acceltime ge bed2 and acceltime le wake2)) then AsleepNW = 1; else
AsleepNW = 0;
                  end;

                  if ('06:00:00't <= bed1 < '19:00:00't) and ('06:00:00't <= bed2 <
'19:00:00't) then do;
                        AsleepNW = 0;
                  end;
            end;


/*###########################################################################################
###########################################################################################
###########################################################################################*/

/*3 sleep periods*/
            if diff12 < 20 and diff12 ne . and diff23 < 20 and diff23 ne . then do;
                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't))
then do;
                        if wake3 < bed1 then do;
                              if acceltime ge bed1 OR acceltime le wake3 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                        else do;
                              if acceltime ge bed1 and acceltime le wake3 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                  end;

                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ('06:00:00't<=
bed2 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if wake1 < bed1 then do;
                              if acceltime ge bed1 or acceltime le wake1 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                        else do;
                              if acceltime ge bed1 and acceltime le wake1 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                  end;
```

```
                  if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't<=
bed1 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if wake2 < bed2 then do;
                              if acceltime ge bed2 or acceltime le wake2 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                        else do;
                              if acceltime ge bed2 and acceltime le wake2 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                  end;

                  if ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't<=
bed1 < '19:00:00't) and ('06:00:00't <= bed2 < '19:00:00't) then do;
                        if wake3 < bed3 then do;
                              if acceltime ge bed3 or acceltime le wake3 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                        else do;
                              if acceltime ge bed3 and acceltime le wake3 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                  end;

                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if wake2 < bed1 then do;
                              if acceltime ge bed1 or acceltime le wake2 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                        else do;
                              if acceltime ge bed1 and acceltime le wake2 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                  end;

                  if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge
'19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't <= bed1 < '19:00:00't) then do;
                        if wake3 < bed2 then do;
                              if acceltime ge bed2 or acceltime le wake3 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                        else do;
                              if acceltime ge bed2 and acceltime le wake3 then AsleepNW =
1; else AsleepNW = 0;
                        end;
                  end;

                  if ('06:00:00't <= bed1 < '19:00:00't) and ('06:00:00't <= bed2 <
'19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        AsleepNW = 0;
                  end;
            end;

            if diff12 < 20 and diff12 ne . and diff23 ge 20 then do;
                  if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't))
then do;
                        if (wake2 < bed1 and (acceltime ge bed1 or acceltime le wake2) or
(wake2 ge bed1 and acceltime ge bed1 and acceltime le wake2) or
```

```
                                (wake3 < bed3 and (acceltime ge bed3 or acceltime le
wake3)) or (wake3 ge bed3 and acceltime ge bed3 and acceltime le wake3)) then AsleepNW =
1; else AsleepNW = 0;
                        end;

                if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ('06:00:00't<=
bed2 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if (wake1 < bed1 and (acceltime ge bed1 or acceltime le wake1) or
(wake1 ge bed1 and acceltime ge bed1 and acceltime le wake1)) then AsleepNW = 1; else
AsleepNW = 0;
                end;

                if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't<=
bed1 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if (wake2 < bed2 and (acceltime ge bed2 or acceltime le wake2) or
(wake2 ge bed2 and acceltime ge bed2 and acceltime le wake2)) then AsleepNW = 1; else
AsleepNW = 0;
                end;

                if ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't<=
bed1 < '19:00:00't) and ('06:00:00't <= bed2 < '19:00:00't) then do;
                        if (wake3 < bed3 and (acceltime ge bed3 or acceltime le wake3) or
(wake3 ge bed3 and acceltime ge bed3 and acceltime le wake3)) then AsleepNW = 1; else
AsleepNW = 0;
                end;

                if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if (wake2 < bed1 and (acceltime ge bed1 or acceltime le wake2) or
(wake2 ge bed1 and acceltime ge bed1 and acceltime le wake2)) then AsleepNW = 1; else
AsleepNW = 0;
                end;

                if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge
'19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't <= bed1 < '19:00:00't) then do;
                        if (wake2 < bed2 and (acceltime ge bed2 or acceltime le wake2) or
(wake2 ge bed2 and acceltime ge bed2 and acceltime le wake2) or
                                (wake3 < bed3 and (acceltime ge bed3 or acceltime le
wake3)) or (wake3 ge bed3 and acceltime ge bed3 and acceltime le wake3)) then AsleepNW =
1; else AsleepNW = 0;
                end;

                if ('06:00:00't <= bed1 < '19:00:00't) and ('06:00:00't <= bed2 <
'19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        AsleepNW = 0;
                end;
            end;

            if diff12 ge 20 and diff23 < 20 and diff23 ne . then do;
                if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't))
then do;
                        if (wake1 < bed1 and (acceltime ge bed1 or acceltime le wake1) or
(wake1 ge bed1 and acceltime ge bed1 and acceltime le wake1) or
                                (wake3 < bed2 and (acceltime ge bed2 or acceltime le
wake3)) or (wake3 ge bed2 and acceltime ge bed2 and acceltime le wake3)) then AsleepNW =
1; else AsleepNW = 0;
                        end;

                if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ('06:00:00't<=
bed2 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
```

```
                        if (wake1 < bed1 and (acceltime ge bed1 or acceltime le wake1) or
(wake1 ge bed1 and acceltime ge bed1 and acceltime le wake1)) then AsleepNW = 1; else
AsleepNW = 0;
                    end;

                    if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't<=
bed1 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if (wake2 < bed2 and (acceltime ge bed2 or acceltime le wake2) or
(wake2 ge bed2 and acceltime ge bed2 and acceltime le wake2)) then AsleepNW = 1; else
AsleepNW = 0;
                    end;

                    if ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't<=
bed1 < '19:00:00't) and ('06:00:00't <= bed2 < '19:00:00't) then do;
                        if (wake3 < bed3 and (acceltime ge bed3 or acceltime le wake3) or
(wake3 ge bed3 and acceltime ge bed3 and acceltime le wake3)) then AsleepNW = 1; else
AsleepNW = 0;
                    end;

                    if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if (wake1 < bed1 and (acceltime ge bed1 or acceltime le wake1) or
(wake1 ge bed1 and acceltime ge bed1 and acceltime le wake1) or
                                (wake2 < bed2 and (acceltime ge bed2 or acceltime le
wake2)) or (wake2 ge bed2 and acceltime ge bed2 and acceltime le wake2)) then AsleepNW =
1; else AsleepNW = 0;
                    end;

                    if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge
'19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't <= bed1 < '19:00:00't) then do;
                        if (wake3 < bed2 and (acceltime ge bed2 or acceltime le wake3) or
(wake3 ge bed2 and acceltime ge bed2 and acceltime le wake3)) then AsleepNW = 1; else
AsleepNW = 0;
                    end;

                    if ('06:00:00't <= bed1 < '19:00:00't) and ('06:00:00't <= bed2 <
'19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        AsleepNW = 0;
                    end;
            end;

            if diff12 ge 20 and diff23 ge 20 then do;
                    if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't))
then do;
                        if (wake1 < bed1 and (acceltime ge bed1 OR acceltime le wake1) or
(wake1 ge bed1 and acceltime ge bed1 and acceltime le wake1) or
                            (wake2 < bed2 and (acceltime ge bed2 OR acceltime le wake2)) or
(wake2 ge bed2 and acceltime ge bed2 and acceltime le wake2) or
                        (wake3 < bed3 and (acceltime ge bed3 OR acceltime le wake3)) or
(wake3 ge bed3 and acceltime ge bed3 and acceltime le wake3)) then AsleepNW = 1; else
AsleepNW = 0;
                    end;

                    if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ('06:00:00't <=
bed2 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if (wake1 < bed1 and (acceltime ge bed1 or acceltime le wake1) or
(wake1 ge bed1 and acceltime ge bed1 and acceltime le wake1)) then AsleepNW = 1; else
AsleepNW = 0;
                    end;

                    if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't <=
bed1 < '19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
```

```
                        if (wake2 < bed2 and (acceltime ge bed2 or acceltime le wake2) or
(wake2 ge bed2 and acceltime ge bed2 and acceltime le wake2)) then AsleepNW = 1; else
AsleepNW = 0;
                    end;

                    if ((bed3 ge '19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't <=
bed1 < '19:00:00't) and ('06:00:00't <= bed2 < '19:00:00't) then do;
                        if (wake3 < bed3 and (acceltime ge bed3 or acceltime le wake3) or
(wake3 ge bed3 and acceltime ge bed3 and acceltime le wake3)) then AsleepNW = 1; else
AsleepNW = 0;
                    end;

                    if ((bed1 ge '19:00:00't) or (bed1 < '06:00:00't)) and ((bed2 ge
'19:00:00't) or (bed2 < '06:00:00't)) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        if (wake1 < bed1 and (acceltime ge bed1 OR acceltime le wake1) or
(wake1 ge bed1 and acceltime ge bed1 and acceltime le wake1) or
                        (wake2 < bed2 and (acceltime ge bed2 OR acceltime le wake2)) or
(wake2 ge bed2 and acceltime ge bed2 and acceltime le wake2)) then AsleepNW = 1; else
AsleepNW = 0;
                    end;

                    if ((bed2 ge '19:00:00't) or (bed2 < '06:00:00't)) and ((bed3 ge
'19:00:00't) or (bed3 < '06:00:00't)) and ('06:00:00't <= bed1 < '19:00:00't) then do;
                        if (wake2 < bed2 and (acceltime ge bed2 OR acceltime le wake2) or
(wake2 ge bed2 and acceltime ge bed2 and acceltime le wake2) or
                         (wake3 < bed3 and (acceltime ge bed3 OR acceltime le wake3)) or
(wake3 ge bed3 and acceltime ge bed3 and acceltime le wake3)) then AsleepNW = 1; else
AsleepNW = 0;
                    end;

                    if ('06:00:00't <= bed1 < '19:00:00't) and ('06:00:00't <= bed2 <
'19:00:00't) and ('06:00:00't <= bed3 < '19:00:00't) then do;
                        AsleepNW = 0;
                    end;
            end;


/*##########################################################################################
/*##########################################################################################
/*##########################################################################################
/*##########################################################################################*/

/*4 sleep periods

            if pid = 99999 and noon = 4 then do;
/*1ST AND 2ND SLEEP PERIODS WERE NOT NOCTURNAL; WAKE3 AND BED4 DIFFERED BY > 20 MINS
                    if (wake3 < bed3 and (acceltime ge bed3 OR acceltime le wake3) or
(wake3 ge bed3 and acceltime ge bed3 and acceltime le wake3) or
                    (wake4 < bed4 and (acceltime ge bed4 OR acceltime le wake4)) or (wake4
ge bed4 and acceltime ge bed4 and acceltime le wake4)) then AsleepNW = 1; else AsleepNW =
0;
            end;*/

      if noon = 0 or noon > 7 then AsleepNW = .;
run;



/*DETERMINING WAKE TIME AND SLEEP TIME FOR DAY 1 (NOON = 0)*/

/*Analyze noon days with only one sleep period*/
data NOON1;
      set FIXBED;
```

```
      by pid noon;
      IF NOON = 0;
      if first.noon and last.noon;
run;
data TIME1;
      set NOON1;
      by pid noon;
      Bed1 = bedtime; Wake1 = end_min;
      format wake1 time8.2 bed1 time8.2;
      drop bedtime end_min strt_nw end_nw dur_nw nw_num;
run;

/*Identify noon days with more than 1 sleep period*/
data MULTI;
      set fixbed;
      by pid noon;
      if last.noon and nw_num > 1;
      Mult = 1;
      keep pid noon mult;
run;
data MULTIS;
      merge fixbed MULTI;
      by pid noon;
      if mult = 1 and noon = 0;
      drop mult;
run;

/*Analyzing noon days with 2 sleep periods
FOR THIS PROJECT, ALL NOON=0 DAYS HAD 1 or 2 SLEEP PERIODS. ADDITIONAL CODE WOULD HAVE TO
BE CREATED FOR 3 OR MORE SLEEP PERIODS*/

data MULTI2;
      set MULTIS;
      by pid noon;
      if first.noon or (last.noon and nw_num = 2);
run;
data TIME2;
      set MULTI2;
      by pid noon;
            if first.noon then do; B1 = Bedtime; W1 = end_min; end;
                  else do; B2 = bedtime; W2 = end_min; end;
      format w1 time8.2 b1 time8.2 w2 time8.2 b2 time8.2;
run;
proc summary data=TIME2 ;
      by pid NOON PAXDAY;
      var b1 w1 b2 w2 ;
      output out=TIME22
    sum = Bed1 Wake1 Bed2 Wake2;
run;
data diff2;
      set TIME22;
      if bed2 < wake1 then Diff = (('24:00:00't + bed2) - wake1)/60;
      else Diff = (bed2 - wake1)/60;
run;
DATA TIMES;
      SET TIME1 DIFF2;
DROP _TYPE_ _FREQ_;
PROC SORT;
      BY PID;
RUN;

/*Combining sleep periods that are < 20 minutes apart*/
DATA DAY1;
```

```
      SET TIMES;
            IF DIFF = . AND (bed1 < '06:00:00't) THEN Day1Wake = WAKE1;
                  ELSE Day1Wake = '00:00:00't;
            IF DIFF < 20 AND DIFF NE . then do;
                  IF ((bed1 < '06:00:00't) and (bed2 < '06:00:00't)) then Day1Wake =
Wake2;
                  IF ((bed1 < '06:00:00't) and (bed2 ge '06:00:00't)) then Day1Wake =
Wake1;
                  IF ((bed1 ge '06:00:00't) and (bed2 ge '06:00:00't)) then Day1Wake =
'00:00:00't;
            END;
            IF DIFF GE 20 THEN DO;
                  IF ((bed1 < '06:00:00't) and (bed2 < '06:00:00't)) THEN DO; Day1Wake =
Wake1; Day1Wake2 = Wake2;
                        end;
                  IF ((bed1 < '06:00:00't) and (bed2 ge '06:00:00't)) then Day1Wake =
Wake1;
                  IF ((bed1 ge '06:00:00't) and (bed2 ge '06:00:00't)) then Day1Wake =
'00:00:00't;
            END;
      FORMAT DAY1WAKE TIME8.2 DAY1WAKE2 TIME8.2;
KEEP PID DAY1WAKE DAY1WAKE2 Bed1 Wake1 Bed2 Wake2 NOON PAXDAY;
RUN;
data day1;
set day1;
Day1Bed1 = Bed1; Day1Bed2 = Bed2;
format day1bed1 time8.2 day1bed2 time8.2;
drop bed1 bed2 wake1 wake2;
run;

/*Assigning sleep/NW indicator to each minute*/
DATA DAY1WAKE;
      MERGE REINTEGRATE DAY1;
      BY PID NOON;
      IF NOON = 0 and DAY1WAKE2 = . THEN DO;
            IF (ACCELTIME < DAY1WAKE and ACCELTIME ge Day1Bed1) THEN AsleepNW = 1;
            ELSE AsleepNW = 0;
      END;
      IF NOON = 0 and DAY1WAKE2 ne . THEN DO;
            IF (ACCELTIME < DAY1WAKE and ACCELTIME ge Day1Bed1) or (ACCELTIME < DAY1WAKE2
and ACCELTIME ge Day1Bed2) THEN AsleepNW = 1;
            ELSE AsleepNW = 0;
      END;
RUN;

/*Sequentially labeling alternating periods of sleep and wake for each noon day*/
data sleep1;
      set day1wake;
      where AsleepNW = 1;
      drop Bed2 Wake2 Diff Bed3 Wake3 Diff12 Diff23 Bed4 Wake4 Diff34;
run;
data sleepper;
      set sleep1;
      by pid noon;
      var = paxn;
      var2 = noon;
      lag_var = lag(var);
      lag_var2 = lag(var2);
      retain slpperiod;
            if first.pid then slpperiod = 1;
            else do;
            if ((var = paxn and (var-1 = lag_var)) and (var2 = noon and (var2 =
lag_var2))) then slpperiod=slpperiod;
```

```
            else slpperiod+1;
            end;
      drop var var2 lag_var lag_var2;
run;
data wake1;
      set day1wake;
      where AsleepNW = 0;
      drop Bed2 Wake2 Diff Bed3 Wake3 Diff12 Diff23 Bed4 Wake4 Diff34;
run;
data wakeper;
      set wake1;
      by pid day;
      var = paxn;
      var2 = day;
      lag_var = lag(var);
      lag_var2 = lag(var2);
      retain wkperiod;
            if first.pid then wkperiod = 1;
            else do;
            if ((var = paxn and (var-1 = lag_var)) and (var2 = day and (var2 =
lag_var2))) then wkperiod=wkperiod;
            else wkperiod+1;
            end;
      drop var var2 lag_var lag_var2;
run;
data periods; /*this dataset only includes days of interest (noon = 0 to 7)*/
      merge sleepper wakeper;
      by pid noon paxn;
run;
data libname.withsleepid;
      set periods;
            if sleepnwmins = . and noon > 0 and noon le 7 then do;
      SleepNWmins = 0; end;
drop bed1 wake1 day1wake day1wake2 DAY1BED1 DAY1BED2;
run;


/*Now applying a non-wear macro (at least 20 consecutive minutes of zero intensity) on
WAKING MINUTES in a midnight to midnight day to identify non-wear*/

data monitors;
      set libname.withsleepid;
      if day ge 1 and day le 7;
run;
proc sort data=monitors;
      by pid day paxn;
run;

%macro nw(nwperiod=);
data nw_all;
  set monitors;
  by PID day paxn;

  if first.day then nw_num=0;    /*non-wear period number*/

  if first.day or reset or stopped then do;
     strt_nw=0;       /*starting minute for the non-wear period*/
     end_nw=0;        /*ending minute for the non-wear period*/
     start=0;         /*indicator for starting to count the non-wear period*/
     dur_nw=0;        /*duration for the non-wear period*/
     reset=0;         /*indicator for resetting and starting over*/
     stopped=0;       /*indicator for stopping the non-wear period*/
```

```
     cnt_non_zero=0; /*counter for the number of minutes with NON-ZERO intensity OR
SLEEP*/
       CNT_NW = 0;
  end;
  retain nw_num strt_nw end_nw stopped reset start cnt_non_zero dur_nw CNT_NW;

  /*ADDING in a counter for minutes of ZERO INTENSITY so that it will start a NW period*/
    if paxinten = 0 AND ASLEEPNW = 0 then
     cnt_NW=cnt_NW+1;

  /*before reaching the 20 consecutive minutes of NW, if encounter one minute with NON-
    ZERO INTENSITY OR ONE MINUTE OF SLEEP, reset the counter*/
  if PAXINTEN > 0 OR ASLEEPNW = 1 then do; cnt_NW=0; end;

  /*THE NW PERIOD STARTS WITH 20 CONSECUTIVE MINUTES OF ZERO INTENSITY*/
  if start=0  and cnt_NW ge 20 then do;
        strt_nw=paxn-19;    /*assign the starting minute of NW*/
        start=1;
  end;

  /*accumulate the number of NW minutes*/
  if start and paxinten=0 AND ASLEEPNW = 0 then
     end_nw=paxn;         /*keep track of the ending minute for the non-wear period*/

  /*keep track of the number of minutes with NON-ZERO intensity OR SLEEP*/
  if 0<paxinten OR ASLEEPNW = 1 then
     cnt_non_zero=cnt_non_zero+1;

  /*duration of NW period*/
  dur_nw=end_nw-strt_nw+1;

  /*A non-wear period ends with a minute of NON-ZERO intensity, one missing count, or one
    minute OF SLEEP*/
  if (cnt_non_zero ge 1 or paxinten=. ) then do;
      if dur_nw<&nwperiod then reset=1;    /*reset if less than &nwperiod minutes of NW*/
    else stopped=1;
  end;

  /*last minute of the day*/
  if last.day and dur_nw>=&nwperiod then stopped=1;

  /*output one record for each NW period*/
  if stopped=1 then do;
                  nw_num=nw_num+1;
                  keep PID day nw_num strt_nw end_nw dur_nw;
                  output;

  end;
run;



*-------------------------------------------------------------------------*;
*define hours of wear                                                     *;
*-------------------------------------------------------------------------*;
/*create a dataset with one record per minute, for the non-wear periods only*/
data nw_minutes(keep=PID day paxn);
  set nw_all;
  by PID day nw_num;
  do i=strt_nw to end_nw by 1;
     paxn=i;
     output;
  end;
```

```
run;

%mend nw;

%nw(nwperiod=20);


data addnwid;
merge monitors nw_minutes(in=nw);
by pid day paxn;
if nw then AddNWid = 1;
else addnwid = 0;
run;


/*NOW APPLYING A NON-WEAR MACRO TO IDENTIFY NW THAT OCCURRED IN A PREVIOUSLY DEFINED
SLEEP PERIOD. 90 minute window of minutes with zero intensity, with allowance for up to 2
minutes of non-zero intensity (can be non-consecutive). */

DATA MONITORS;
SET addnwid;
proc sort;
      by PID SLPPERIOD PAXN;
run;

%macro nw(nwperiod=);
data nw_all;
  set MONITORS;
  by PID SLPPERIOD paxn ;

  if FIRST.SLPPERIOD then nw_num=0;    /*non-wear period number*/

  if FIRST.SLPPERIOD OR reset or stopped then do;
     strt_nw=0;       /*starting minute for the non-wear period*/
     end_nw=0;        /*ending minute for the non-wear period*/
     start=0;         /*indicator for starting to count the non-wear period*/
     dur_nw=0;        /*duration for the non-wear period*/
     reset=0;         /*indicator for resetting and starting over*/
     stopped=0;       /*indicator for stopping the non-wear period*/
     cnt_non_zero=0; /*counter for the number of minutes with NON-ZERO INTENSITY*/
       CNT_NW = 0;
  end;
  retain nw_num strt_nw end_nw stopped reset start cnt_non_zero dur_nw CNT_NW ;

  /*ADDING in a counter for NW minutes so that it will start a NW period*/
    if paxinten = 0 then
     cnt_NW=cnt_NW+1;

  /*keep track of the number of minutes with NON-ZERO intensity*/
  if 0 < paxinten  then
     cnt_non_zero=cnt_non_zero+1;

  /*before reaching the 90 consecutive minutes of NW, if encounter 3 minutes with NON-
    ZERO INTENSITY, reset the counter*/
  if cnt_non_zero = 3 then do; cnt_NW=0; end;

  /*THE NW PERIOD STARTS WITH 90 MINUTES, WITH UP TO 2 OF THOSE MINUTES BEING NON-ZERO
INTENSITY*/
  if start=0  and (cnt_NW+CNT_NON_ZERO = 90) then do;
        strt_nw=paxn-89;    /* assign the starting minute of NW*/
        start=1;
  end;
```

```
  /*accumulate the number of the non-wear minutes*/
  if start and paxinten=0 then do;
     end_nw=paxn;         /*keep track of the ending minute for the NW period*/
       end;

  /*duration of NW period*/
  dur_nw=end_nw-strt_nw+1;

  /*A NW period ends with 3 minutes of ANY intensity, one missing count, or one wake
minute*/
  if (cnt_non_zero=3 or paxinten=. OR ASLEEPNW = 0) then do;
      if dur_nw<&nwperiod then reset=1;    /*reset if less than &nwperiod minutes of NW*/
    else stopped=1;
  end;

  /*last minute of the sleep period*/
  if last.SLPPERIOD and dur_nw>=&nwperiod then stopped=1;

  /*output one record for each NW period*/
  if stopped=1 then do;
                  nw_num=nw_num+1;
                  keep PID NOON SLPPERIOD nw_num strt_nw end_nw dur_nw ;
                  output;

  end;
run;


*-------------------------------------------------------------------------*;
*summarize the non-wear periods to one record per NOON                    *;
*-------------------------------------------------------------------------*;
proc summary data=nw_all ;
  by PID SLPPERIOD ;
  var dur_nw ;
  output out=sum_nw
       sum=tot_dur_nw
run;

*-------------------------------------------------------------------------*;
*summarize the total number of valid minutes for everyone in the analysis.*;
*-------------------------------------------------------------------------*;

data monc;
set monitors;
if slpperiod = . then delete;
run;
proc summary data=monc;
  by PID SLPPERIOD  ;
  var paxinten;
  output out=sum_all
         n=tot_min;
run;

*-------------------------------------------------------------------------*;
*define hours of wear                                                     *;
*-------------------------------------------------------------------------*;
/*create a dataset with one record per minute, for the non-wear periods only*/
data nw_minutes(keep=PID SLPPERIOD NOON paxn);
  set nw_all;
  by PID  SLPPERIOD nw_num;
  do i=strt_nw to end_nw by 1;
     paxn=i;
     output;
```

```
  end;
run;

/*create a dataset from the monitor data, restricted to the wear periods*/
data wear_minute(keep=PID NOON SLPPERIOD paxn paxinten);
  merge monitors(in=in_all) nw_minutes(in=in_nw);
  by PID  SLPPERIOD paxn;
  if in_all and not in_nw;
run;

/*summarize the wear minutes */
proc summary data=wear_minute;
  by PID SLPPERIOD;
  var paxinten;
  output out=sum_wear
         sum=tot_cnt_wr
         n=tot_min_wr;
run;

*-----------------------------------------------------------------------------*;
*final data for one record per SLEEP PERIOD for everyone in the analysis *;
*-----------------------------------------------------------------------------*;
data nwperiod;
  merge sum_all(in=in_all) sum_nw(in=in_nw) sum_wear;
  by PID SLPPERIOD;
  if in_all;

  if tot_dur_nw=. then tot_dur_nw=0;
  if tot_min_wr=. then tot_min_wr=0;
  if tot_cnt_wr=. then tot_cnt_wr=0;

  label
  tot_dur_nw='Total duration of non-wear periods in a sleep period'
  tot_min='Total number of valid minutes within a sleep period'
  tot_cnt_wr='Total intensity counts from all wear minutes in a sleep period'
  tot_min_wr='Total number of wear minutes in a sleep period'
  ;
  keep PID SLPPERIOD tot_min tot_min_wr tot_cnt_wr tot_dur_nw;
run;

%mend nw;

%nw(nwperiod=90);

/*Summarizing sleep time for each sleep period to calculate percent of sleep period
minutes that were actually NW*/
proc summary data=monitors;
  by pid  slpperiod;
  var asleepnw;
  output out=AllSleep
       sum = tot_sleep;
run;
data AllSleep;
set AllSleep;
if slpperiod = . then delete;
run;

DATA COMPARE;
MERGE AllSleep NWPERIOD;
BY PID slpperiod;
PERCENTNW = round(TOT_DUR_NW/TOT_SLEEP, .0001);
DROP TOT_CNT_WR;
proc sort;
```

```
by pid slpperiod;
run;
/*Creating an identifier for ALL minutes in a sleep period when 90% of the sleep period
is NW*/
data addnwtosleep;
merge monitors compare(in=nw);
by pid slpperiod;
if nw and percentnw ge .90 then  NWinSleep = 1;
else NWinSleep = 0;
drop percentnw sleepnwmins tot_dur_nw _type_ _freq_ tot_sleep tot_min tot_min_wr ;
run;


/*Creating final NW and sleep identifiers for each minute; censoring minutes with
intensity counts of at least 20,000 to change them to NW or sleep*/
data nwsleep;
set addnwtosleep;
if asleepnw = 1 and nwinsleep = 0 then do; FinalNW = 0; FinalSleep = 1; end;
if asleepnw = 1 and nwinsleep = 1 then do; FinalNW = 1; FinalSleep = 0; end;
if addnwid = 1 then do; FinalNW = 1; FinalSleep = 0; end;
if asleepnw = 0 and addnwid = 0 then do; FinalNW = 0; FinalSleep = 0; end;

if paxinten ge 20000 and (asleepnw = 0 or nwinsleep = 1) then do; FinalNW = 1; FinalSleep
= 0; end;
      else if paxinten ge 20000 and asleepnw = 1 and nwinsleep = 0 then do; FinalNW = 0;
FinalSleep = 1; end;

if finalsleep = 1 or finalnw = 1 then SleepNWIdentifier = 1; else SleepNWIdentifier = 0;
drop sleepincoff;
proc sort;
  by pid paxn ;
run;

data libname.finalpermin;
set nwsleep;
drop asleepnw addnwid nwinsleep;
format finalnw yesno. finalsleep yesno. sleepnwidentifier yesno.;
label finalnw = "Identifier for minute of non-wear (yes/no)" finalsleep = "Identifier for
minute of sleep (yes/no)"
sleepnwidentifier = "Identifier for minute of sleep OR non-wear (yes/no)"
Steps = "Total steps for the minute processed with Low Frequency Extension OFF" Paxn =
"Sequential Minute"
Paxinten = "Total intensity counts in the vertical axis for the minute"
Axis2 = "Total intensity counts in the anterior-posterior axis for the minute"
Axis3 = "Total intensity counts in the medial-lateral axis for the minute"
slpperiod = "Sequential sleep period" wkperiod = "Sequential wake period";
run;

/*At this point, a minute-by-minute dataset has been created that labels each minute as
waking wear (FinalNW = 0, FinalSleep = 0, SleepNWIdentifier = 0), sleeping wear
(FinalSleep = 1, SleepNWIdentifier = 1) or non-wear (FinalNW = 1, SleepNWIdentifier = 1).
Any physical activity summaries or analysis of waking wear time should only be performed
for minutes where SleepNWIdentifier = 0.



Code below further summarizes these minutes to create a daily-level sleep dataset based
on noon-to-noon days*/

/*Bed/wake times*/
proc sort data= libname.finalpermin;
by pid slpperiod paxn ;
```

```
run;
data bed1;
      set libname.finalpermin;
      where finalsleep = 1 and noon ge 1 and noon < 7;
run;
proc sort data=bed1 nodupkey dupout=mult out=newbed;
by pid noon;
run;
data bed2;
set newbed;
bedtime = acceltime;
format bedtime time8.2;
keep pid noon bedtime;
run;

proc sort data = bed1;
by pid slpperiod descending paxn;
run;
data wake;
      set bed1;
      by pid noon ;
      var = paxn;
      var2 = noon;
      lag_var = lag(var);
      lag_var2 = lag(var2);
      retain slpperiod;
            if first.pid then waketime = acceltime;
            else do;
            if ((var = paxn and (var+1 = lag_var)) and (var2 = noon and (var2 =
lag_var2))) then waketime = .;
            else waketime = acceltime;
            end;
            format waketime time8.2;
      if waketime = . then delete;
run;
proc sort data=wake;
by pid noon descending slpperiod;
run;
proc sort data=wake nodupkey dupout=mult out=newwake;
by pid noon;
run;

data inbed;
merge bed2 newwake(keep=pid noon waketime);
by pid noon;
run;

proc sort data= libname.finalpermin;
by pid noon paxn;
run;
proc summary data= libname.finalpermin;
  by pid noon;
  var finalsleep finalnw sleepnwidentifier;
  output out=sleepnoon
       sum =  tot_sleep tot_nw tot_sleepnw;
run;

/*Retains the paxday variable which corresponds to the day they went to bed*/
proc sort data= libname.finalpermin nodupkey dupout=mult out=noons;
by pid noon;
run;
data noons2;
set noons;
```

```
keep pid noon paxday;
run;
data noonsleep;
merge noons2 sleepnoon(keep=pid noon tot_sleep tot_nw tot_sleepnw) inbed(in=b);
by pid noon;
if noon in (0,7) then delete;
if waketime = '23:59:00't then delete;
if waketime > '19:00:00't and waketime < '24:00:00't then do; waketime = .; end;
/*Setting these rare instances to . so as not to affect mean wake time in a weekly
summary*/
label tot_sleep = "Total sleep episode time" bedtime = "Sleep onset" waketime = "Sleep
offset";
format tot_nw 12. tot_sleep 12. tot_sleepnw 12.;
run;

/*Creating a daily-level dataset related to sleep*/
data libname.noonsleep;/*3031*/
set noonsleep;
if tot_sleep = 0 then do; tot_sleep = .; end; /*Setting TSET for days with no detectable
sleep to missing so as not to affect mean sleep time in a weekly summary*/
drop _type_ _freq_;
run;



/*Averaging daily data to create a weekly summary of sleep based on noon-to-noon days*/

proc sort data= libname.noonsleep;
by pid noon;
run;
data sleep2;
set libname.noonsleep;
if bedtime < '6:00:00't then do; bedtime = bedtime+'24:00:00't; end;
run;
proc summary data=sleep2;
  by pid;
  var TOT_SLEEP
        bedtime
        waketime;
  output out=week
  mean(TOT_SLEEP
        bedtime
        waketime)=
      MeanSleep
        MeanBedtime
        MeanWaketime
n(tot_sleep)=WeekValidSleep;
run;
data week;
set week;
if  MeanBedtime ge '24:00:00't then do;
      MeanBedtime = MeanBedtime-'24:00:00't; end;
label MeanSleep = "Mean TSET (mins) from all valid sleep nights"
        MeanBedtime = "Mean sleep onset from all valid sleep nights"
        MeanWaketime = "Mean sleep offset from all valid sleep nights"
        WeekValidSleep = "Number of nights with valid sleep";
drop _type_ _freq_;
run;


/*Summarizing data by day of the week*/
proc summary data= libname.noonsleep;
```

```
  by pid;
  where  paxday = 2;
  var TOT_SLEEP
        bedtime
        waketime;
  output out=mon
  mean(TOT_SLEEP
        bedtime
        waketime)=
      MonSleep
        MonBedtime
        MonWaketime;
run;
data Mon;
set Mon;
label MonSleep = "TSET (mins) from Monday night to Tuesday morning"
        MonBedtime = "Sleep onset on Monday night"
        MonWaketime = "Sleep offset on Tuesday morning";
drop _type_ _freq_;
run;

proc summary data= libname.noonsleep;
  by pid;
  where  paxday = 3;
  var TOT_SLEEP
        bedtime
        waketime;
  output out=Tues
  mean(TOT_SLEEP
        bedtime
        waketime)=
      TuesSleep
        TuesBedtime
        TuesWaketime;
run;
data Tues;
set Tues;
label TuesSleep = "TSET (mins) from Tuesday night to Wednesday morning"
        TuesBedtime = "Sleep onset on Tuesday night"
        TuesWaketime = "Sleep offset on Wednesday morning";
drop _type_ _freq_;
run;

proc summary data= libname.noonsleep;
  by pid;
  where  paxday = 4;
  var TOT_SLEEP
        bedtime
        waketime;
  output out=Wed
  mean(TOT_SLEEP
        bedtime
        waketime)=
      WedSleep
        WedBedtime
        WedWaketime;
run;
data Wed;
set Wed;
label WedSleep = "TSET (mins) from Wednesday night to Thursday morning"
        WedBedtime = "Sleep onset on Wednesday night"
        WedWaketime = "Sleep offset on Thursday morning";
drop _type_ _freq_;
```

```
run;

proc summary data= libname.noonsleep;
  by pid;
  where  paxday = 5;
  var TOT_SLEEP
        bedtime
        waketime;
  output out=Thur
  mean(TOT_SLEEP
        bedtime
        waketime)=
      ThurSleep
        ThurBedtime
        ThurWaketime;
run;
data Thur;
set Thur;
label ThurSleep = "TSET (mins) from Thursday night to Friday morning"
        ThurBedtime = "Sleep onset on Thursday night"
        ThurWaketime = "Sleep offset on Friday morning";
drop _type_ _freq_;
run;



proc summary data= libname.noonsleep;
  by pid;
  where  paxday = 6;
  var TOT_SLEEP
        bedtime
        waketime;
  output out=fri
  mean(TOT_SLEEP
        bedtime
        waketime)=
      FriSleep
        FriBedtime
        FriWaketime;
run;
data Fri;
set Fri;
label FriSleep = "TSET (mins) from Friday night to Saturday morning"
        FriBedtime = "Sleep onset on Friday night"
        FriWaketime = "Sleep offset on Saturday morning";
drop _type_ _freq_;
run;

proc summary data= libname.noonsleep;
  by pid;
  where paxday = 7;
  var TOT_SLEEP
        bedtime
        waketime;
  output out=Sat
  mean(TOT_SLEEP
        bedtime
        waketime)=
      SatSleep
        SatBedtime
        SatWaketime;
run;
data sat;
```

```
set sat;
label satSleep = "TSET (mins) from Saturday night to Sunday morning"
        satBedtime = "Sleep onset on Saturday night"
        satWaketime = "Sleep offset on Sunday morning";
drop _type_ _freq_;
run;

proc summary data= libname.noonsleep;
  by pid;
  where  paxday = 1;
  var TOT_SLEEP
        bedtime
        waketime;
  output out=Sun
  mean(TOT_SLEEP
        bedtime
        waketime)=
      SunSleep
        SunBedtime
        SunWaketime;
run;
data Sun;
set Sun;
label SunSleep = "TSET (mins) from Sunday night to Monday morning"
        SunBedtime = "Sleep onset on Sunday night"
        SunWaketime = "Sleep offset on Monday morning";
drop _type_ _freq_;
run;


/*Creating a dataset with one row per subject that only retains sleep summary data for
subjects with valid sleep*/
data SLEEPperson;
merge week  mon tues wed thur fri sat sun;
by pid;
if ((sunsleep>0) + (monsleep>0) + (tuessleep>0) + (wedsleep>0) + (thursleep>0)) ge 2 and
((frisleep>0) or (satsleep >0));
Valid_Sleep = 1;
label Valid_sleep = "At least 3 valid nights of sleep, including one weekend night";
run;
data libname.sleepperson;
merge week(in=a keep=pid WeekValidSleep) sleepperson;
by pid;
if valid_sleep = . then do; valid_sleep = 0;end;
run;
```