Paper 9881-2016

Using SAS® Arrays to Calculate Bouts of Moderate to Vigorous Physical Activity from Minute-by-Minute Fitbit Data

Faith E Parsons, Keith M Diaz, and Jacob E Julian, Center for Behavioral Cardiovascular Health, Columbia University Medical Center

ABSTRACT

The increasing popularity and affordability of wearable fitness tracking devices, together with their ability to provide granular physical activity data down to the minute-level, has enabled researchers to conduct advanced studies on the effects of physical activity on health and disease. This technology also provides statistical programmers the challenge of processing these data and translating them into analyzable measures. One such measure is the number of time-specific bouts of moderate to vigorous physical activity (MVPA), (akin to 'exercise') which is needed to determine if the participant meets current physical activity guidelines (e.g., 150 min. of MVPA/week performed in bouts of at least 10 min.). In this paper, we illustrate how we used SAS® arrays to calculate the number of 10-min. bouts of MVPA per day and provide a working code that can be ran on a sample file.

INTRODUCTION

Physical activity is an important lifestyle behavior that can improve health, prevent chronic disease, and enhance the general well-being for people of all ages. According to public health guidelines, for physical activity to be health enhancing it should (1) be of moderate or vigorous in intensity, (2) occur in bouts of 10 consecutive minutes or more (e.g. 'exercise'), and (3) total at least 150 minutes per week. Perceived physical activity levels however, have often been subjective (particularly for intensity of physical activity), and depend on an individual's knowledge of the common parameters related to physical activity, including heart rate and caloric expenditure. Technology has long sought to help individuals keep track of their heart rate and caloric expenditure however, like most new technologies, was cost prohibitive, uncomfortable, unfashionable, and required proprietary software to access data. Recent advances in technology has enabled companies such as Fitbit® to introduce slim fitting, inexpensive devices, that work with any internet-connected computing device. As these products become more widely used, they are opening new opportunities---and providing new challenges--- for data analysts and researchers to quantify and monitor individual's physical activity.

In a study assessing the potential bidirectional relationship between stress and physical activity (e.g. does stress influence physical activity behaviors and vice versa), the Center of Behavioral Cardiovascular Health (CBCH) at Columbia University Medical Center provided Fitbit® Flex devices to healthy volunteers and monitored their daily physical activity for 12 months. Given the voluminous nature of the data collected in this study (e.g. minute-by-minute measures of physical activity for 12 months); we endeavored to translate these data into a quantitative estimate of physical activity that is meaningful and interpretable.

We used Fitabase®, *"a research platform that collects data from internet connected devices"* (Small Steps Labs LLC, 2016), to download comma-separated values (CSV) files of the activity data from the Fitbit Flex devices, and derived measures of physical activity. Raw data were available for number of steps taken, physical activity intensity, calories burned, and METs, in the form of daily totals, hourly totals and minute-by-minute counts. For this paper we focus exclusively on the minute-by-minute physical activity intensities.

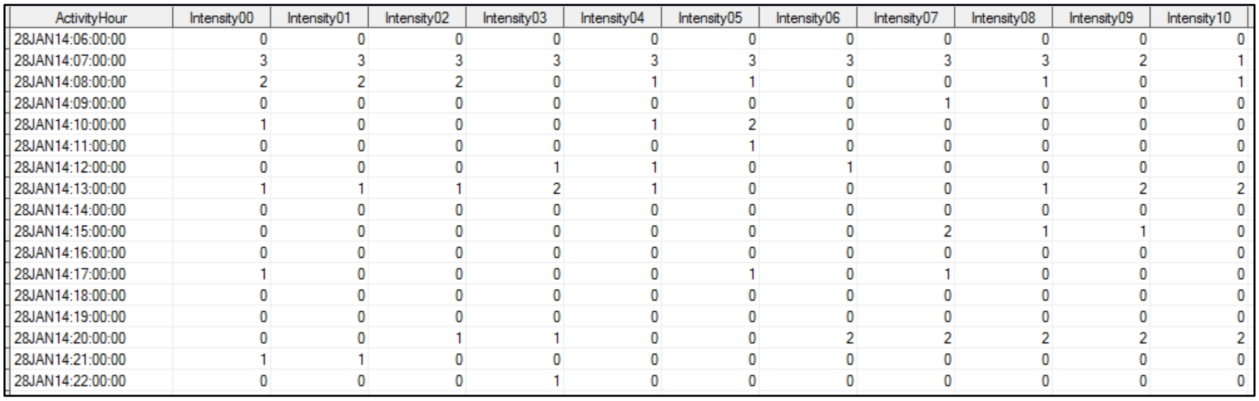
The purpose of this paper is to provide a rationale for, and step-by-step guide explaining, how we used SAS to derive the number of 10-min. episodes of moderate-to-vigorous physical activity (MVPA) per day from the minute-by-minute physical activity intensities data. These data can be used to determine 'exercise' bouts and ascertain if the participant meets current physical activity guidelines (e.g., 150 min. of MVPA/week performed in bouts of at least 10 min.) Specifically, we created a MACRO that used SAS arrays to (1) transpose the raw data,(2) create variables containing the number of MVPA minutes within

10 consecutive minutes, and (3) count the number of 10-min. MPVA bouts per *activity day*. We conclude by discussing the potential value of this approach.

RAW DATA

We used the IMPORT procedure to convert the CSV file into a SAS dataset. Figure 1 shows a snapshot of the raw SAS data. The data file is, by default, sorted in ascending order on the `ActivityHour` variable which uniquely identifies each row based on the date and hour of the activity. The variables `Intensity00` to `Intensity59` correspond to the exercise intensity for each minute of the hour, and are coded as level 0 (sedentary), 1 (lightly active), 2 (fairly active) and 3 (very active) (Fitbit Inc., 2016). For the calculation of MVPA bouts, we used level 2 as *moderate-intensity* and 3 as *vigorous intensity*.

Figure 1. Raw data for minute-by-minute exercise intensities



ActivityHour	Intensity00	Intensity01	Intensity02	Intensity03	Intensity04	Intensity05	Intensity06	Intensity07	Intensity08	Intensity09	Intensity10
28JAN14:06:00:00	0	0	0	0	0	0	0	0	0	0	0
28JAN14:07:00:00	3	3	3	3	3	3	3	3	3	2	1
28JAN14:08:00:00	2	2	2	0	1	1	0	0	1	0	1
28JAN14:09:00:00	0	0	0	0	0	0	0	1	0	0	0
28JAN14:10:00:00	1	0	0	0	1	2	0	0	0	0	0
28JAN14:11:00:00	0	0	0	0	0	1	0	0	0	0	0
28JAN14:12:00:00	0	0	0	1	1	0	1	0	0	0	0
28JAN14:13:00:00	1	1	1	2	1	0	0	0	1	2	2
28JAN14:14:00:00	0	0	0	0	0	0	0	0	0	0	0
28JAN14:15:00:00	0	0	0	0	0	0	0	2	1	1	0
28JAN14:16:00:00	0	0	0	0	0	0	0	0	0	0	0
28JAN14:17:00:00	1	0	0	0	0	1	0	1	0	0	0
28JAN14:18:00:00	0	0	0	0	0	0	0	0	0	0	0
28JAN14:19:00:00	0	0	0	0	0	0	0	0	0	0	0
28JAN14:20:00:00	0	0	1	1	0	0	2	2	2	2	2
28JAN14:21:00:00	1	1	0	0	0	0	0	0	0	0	0
28JAN14:22:00:00	0	0	0	1	0	0	0	0	0	0	0

PREPARING DATA FOR TRANSPOSITION

We defined an *activity day* to commence at 3:00 am and complete at 2:59 am the following day. This allowed us to account for users who are active after midnight. We then limited the range of the data to include dates and times between the first and last *activity days.*

Step 1**:** Using a macro variable, we defined the beginning hour of the *activity day* using the standard military time format where 0 hours =12:00am and 23 hours =11:00pm. In the following steps, we use the `%LET` statement to assign the commencement hour to the macro variable `&starthour:`

```
%let starthour = 3;

%if &starthour. = 0 %then %do;
   %let endhour = 23; %end;
%else %do;
   %let endhour = %sysevalf(&starthour. - 1);
%end;
%put &starthour. &endhour.;
```

An alternative method, which we used in the sample working code, is to define `starthour` as a parameter in the macro that will process the succeeding steps to calculate the number of 10-min. bouts of MPVA:

```
%macro mvpa (starthour=);
   %if &starthour. = 0 %then %do;
         %let endhour = 23; %end;
   %else %do;
         %let endhour = %sysevalf(&starthour. - 1); %end;
   %put &starthour. &endhour.;
   …
%mend;
```

Because we defined `starthour` as a parameter, we specified the starting hour when we call the macro.

```
%mvpa(starthour=3);
```

Step 2: We extracted the date from *`ActivityHour`*, and created an integer variable corresponding to each hour of the day from 0 to 23.

```
   ActDate = DatePart(ActivityHour);
   ActHour = Hour(TimePart(ActivityHour));
```

Step 3: We identified the date of the first *activity day* in the dataset and assigned to a macro variable called `startdate`.

```
data _null_;
   set int2 (firstobs= 1 obs=1);
   call symput('startdate', compress(ActDate));
run;
```

Step 4: We identified the date of the last *activity day* in the dataset and assigned to the macro variable `enddate`. We used the `NUMOBS` macro to identify the number of observations in the *intensities* dataset, which is then passed to a macro variable `&NUM`. We then used `&NUM` in the `FIRSTOBS` and OBS option to retain the last row in the dataset, and the `CALL SYMPUT` statement to assign the last activity date to `&ENDDATE`.

```
%MACRO NUMOBS (DSN=) ;
   %GLOBAL NUM ;
   DATA _NULL_ ;
      IF 0 THEN SET &DSN NOBS=COUNT ;
      CALL SYMPUT ('NUM', LEFT(PUT(COUNT,8.))) ;
      STOP ;
   RUN;
%MEND;
%NUMOBS (DSN=int2) ;
data _null_;
   set int2  (firstobs= &NUM. obs=&NUM.);
   call symput ('enddate',  compress(ActDate));
run;
```

Step 5: We deleted all rows prior to the start of the first *activity day* and after the last *activity day*. This step was necessary to ensure that the data was transposed correctly, particularly when the `&starthour` is not 0:

```
data int3 ;
retain ActDate ActHour;
   set int2;
   ActDate = DatePart(ActivityHour);
   ActHour = Hour(TimePart(ActivityHour));
   format ActDate mmddyy10.;
   if ActivityHour >= dhms(&startdate., &starthour.,0,0) and
      ActivityHour <= dhms(&enddate.,&endhour.,0,0);
run;
```

TRANPOSING TO A LONGITUDINAL DATASET

After the data had been restricted to contain only rows between the first and last *activity day*, we transform the data into a longitudinal format. The purpose of this section of the macro is to create a dataset with the following variables:

- ActDate – activity date
- ActHour – integer from 0 to 23 corresponding to the hour of the activity time
- Ind_DayHour – indicator variable with values 1 to 24, corresponding the nth hour of the *activity day*. In our example, 03:00 corresponds to the 1st hour of the day.
- Ind_DayNum – indicator variable, starting with 1, corresponding the the nth date activity date in the datasets
- min – minute within each hour with values from 0 to 59
- intensity – the exercise intensity for a specific minute with values 0 (sedentary), 1 (light), 2 (moderate), and 3 (vigorous)
- ModVig – a binary variable with values 1 if intensity is 2 or 3, and 0 if intensity is 0 or 1.

Step 6: We converted the data into a longitudinal format that will contain the variables ActDate, ActHour, ActivityHour, min and intensity using the following DATA step:

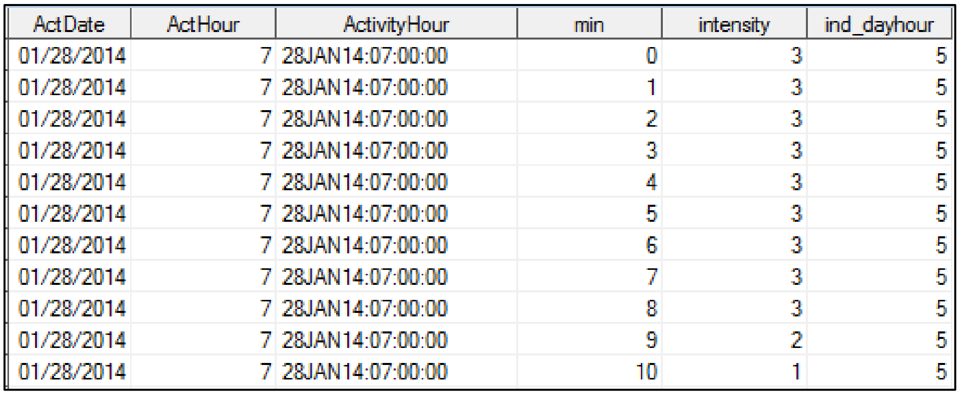
```
data int_long;
   set int_2;
   array mins(0:59) intensity00-intensity59;
         do min = 0 to 59;
         intensity = mins(min);
         output;
   end;

   drop intensity00-intensity59;
run;
```

Step 7: Create the ind_dayhour variable. **Figure 2.**shows a snapshot of the first 10 rows from this step:

```
data ind;
   set int_long;
   temp1 = ActHour - &startHour. + 1;
   if temp1 <=0 then ind_dayhour = 24+temp1;
      else ind_dayhour = temp1;
      drop temp1;
run;
```

Figure 2. Intensities data in longitudinal format with the indicator variable, ind_dayhour.



ActDate	ActHour	ActivityHour	min	intensity	ind_dayhour
01/28/2014	7	28JAN14:07:00:00	0	3	5
01/28/2014	7	28JAN14:07:00:00	1	3	5
01/28/2014	7	28JAN14:07:00:00	2	3	5
01/28/2014	7	28JAN14:07:00:00	3	3	5
01/28/2014	7	28JAN14:07:00:00	4	3	5
01/28/2014	7	28JAN14:07:00:00	5	3	5
01/28/2014	7	28JAN14:07:00:00	6	3	5
01/28/2014	7	28JAN14:07:00:00	7	3	5
01/28/2014	7	28JAN14:07:00:00	8	3	5
01/28/2014	7	28JAN14:07:00:00	9	2	5
01/28/2014	7	28JAN14:07:00:00	10	1	5

Step 8: Note that the dataset illustrated in **Figure 2** contains repeating rows of the same ActDate and Ind_dayhour. We need to create another dataset that contains only unique combinations of these two variables in order to count the total number of *activity days* — one way to achieve this is to retain only the

first minute of each hour:

```
    data ind2;
        set ind;
        if min = 0;
```

Because the purpose of this step is to create a dataset that will count the number of *activity days*, we can drop the `min` and `intensity` variables.

```
        drop min intensity;
    run;
```

Step 9: In this step we created the `ind_daynum` variable, and the resulting dataset will be merged with the longitudinal *intensities* data file from **Step 6**. A section of the output dataset is shown in **Figure 3**:

```
    proc sort data=ind2;
        by ind_dayhour actdate;
    run;

    data ind3;
        set ind2;
        ind_daynum + 1;
        by ind_dayhour;
        if first.ind_dayhour then ind_daynum= 1;
    run;
```

Step 10: We sorted the datasets from **Step 6** and **Step 9**, merged by `ActDate` and `ActHour`, and created the binary variable, `ModVig`:

```
  proc sort data=ind3;
    by ActDate ActHour;
  proc sort data=Int_long;
    by ActDate ActHour;
  run;

  Data Intensities ;
    merge ind3 Int_long;
    by ActDate ActHour;

    modvig = (intensity >=2);
  run;
```

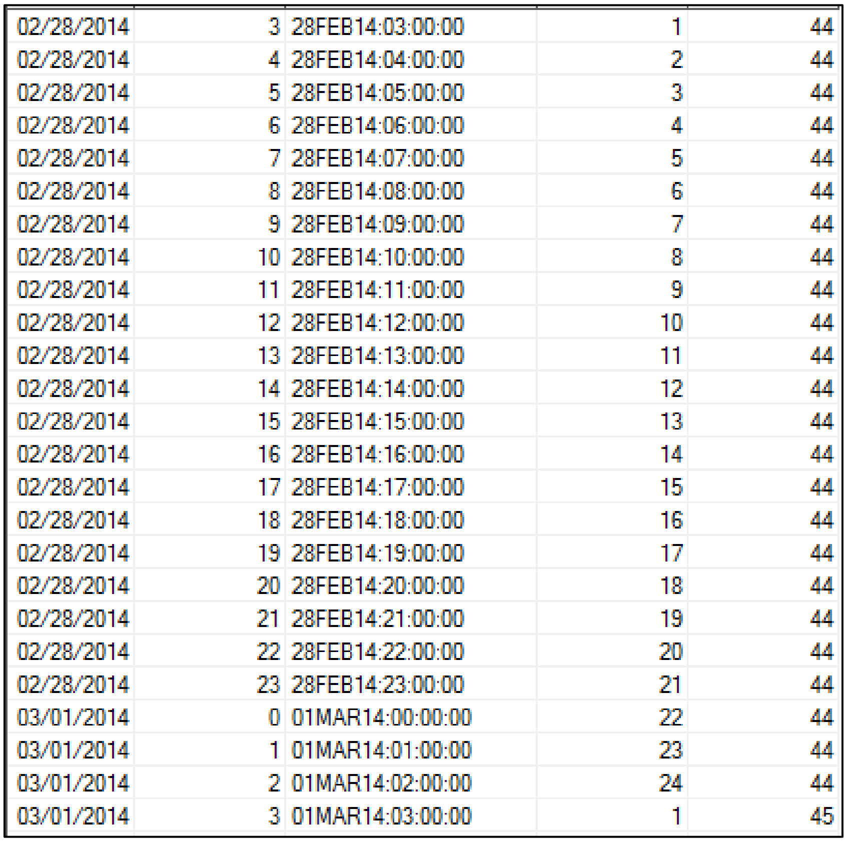
CONVERTING DATA TO WIDE FORMAT

We created the following steps to create a dataset in wide format where each row corresponds to one 'activity day', with one column for the activity date, and 1440 columns corresponding to the number of minutes in in one day.

Step 11: We sorted the *intensities* dataset by `ind_daynum`, `ind_dayhour` and `min`.

```
  proc sort data=intensities;
    by ind_daynum ind_dayhour min;
  run;
```

Figure 3. Dataset with both indicator variables.



02/28/2014	3	28FEB14:03:00:00	1	44
02/28/2014	4	28FEB14:04:00:00	2	44
02/28/2014	5	28FEB14:05:00:00	3	44
02/28/2014	6	28FEB14:06:00:00	4	44
02/28/2014	7	28FEB14:07:00:00	5	44
02/28/2014	8	28FEB14:08:00:00	6	44
02/28/2014	9	28FEB14:09:00:00	7	44
02/28/2014	10	28FEB14:10:00:00	8	44
02/28/2014	11	28FEB14:11:00:00	9	44
02/28/2014	12	28FEB14:12:00:00	10	44
02/28/2014	13	28FEB14:13:00:00	11	44
02/28/2014	14	28FEB14:14:00:00	12	44
02/28/2014	15	28FEB14:15:00:00	13	44
02/28/2014	16	28FEB14:16:00:00	14	44
02/28/2014	17	28FEB14:17:00:00	15	44
02/28/2014	18	28FEB14:18:00:00	16	44
02/28/2014	19	28FEB14:19:00:00	17	44
02/28/2014	20	28FEB14:20:00:00	18	44
02/28/2014	21	28FEB14:21:00:00	19	44
02/28/2014	22	28FEB14:22:00:00	20	44
02/28/2014	23	28FEB14:23:00:00	21	44
03/01/2014	0	01MAR14:00:00:00	22	44
03/01/2014	1	01MAR14:01:00:00	23	44
03/01/2014	2	01MAR14:02:00:00	24	44
03/01/2014	3	01MAR14:03:00:00	1	45

Step 12. We used arrays to convert the *intensities* data into wide format:

```
data modvig_wide check.modvig_wide_&pid.;
   retain ActDate modvig_hr01_min00-modvig_hr01_min59 …
      modvig_hr24_min00-modvig_hr24_min59;
   keep ActDate modvig_hr01_min00-modvig_hr01_min59 …
      modvig_hr24_min00-modvig_hr24_min59;
   array mv [1:24,0:59] modvig_hr01_min00-modvig_hr01_min59  …
      modvig_hr24_min00-modvig_hr24_min59;

   set Intensities;
   by ind_daynum ;
      if first.ind_daynum then
         do i = 1 to 24;
            do j = 0 to 59;
               mv[i, j] = .;
            end;
         end;
      mv[ind_dayhour, min] = modvig;
```

Recall that we previously redefined the *activity day* to commence at 3am (Step **1**), and the current dataset is sorted by `ind_daynum, ind_dayhour,` and `min`. Because this DATA step will only retain the last instance of `ind_daynum` (see second line of code below), we need to adjust `ActDate` by 1. However, we also needed to account for when we need to start the *activity day* at 0 hours, therefore we used the following IF statement:

```
      if &starthour. ne 0 then ActDate = Actdate-1;
      if last.ind_daynum then output;
   run;
```

Step 13. In a new DATA step, we used SAS arrays to count the number of MVPA minutes within 10 consecutive minutes, and created binary variable, `MPVA_10m_any`, with value 1 if there are any 10-minute episodes of MVPA:

```
data modvig_sums;
   retain actdate
      mvsum_hr01_min00-mvsum_hr01_min59 …
      mvsum_hr24_min00-mvsum_hr24_min59;
   array mv [1440] modvig_hr01_min00-modvig_hr01_min59 …
      modvig_hr24_min00-modvig_hr24_min59;
```

The variables in the array definition below corresponds to the number of MVPA minutes in the last 10 minutes, inclusive. For example, `mvsum_hr01_min10` is the number of MVPA minutes from minute 1 to minute 10 of hour 1..

```
array mvsum [1431]
   mvsum_hr01_min09-mvsum_hr01_min59 … mvsum_hr24_min00-mvsum_hr24_min59;

set modvig_wide;
MPVA_10m_any = 0;

do i= 1 to 1431;
mvsum[i] = sum(mv[i], mv[i+1], mv[i+2], mv[i+3], mv[i+4], mv[i+5],
           mv[i+6], mv[i+7], mv[i+8], mv[i+9], mv[i+10]);
if mvsum[i] >= 10 then MPVA_10m_any = 1;
end;
```

Step 14. Note that the array `mvsum` in the previous step started with `mvsum_hr01_min09`, while the RETAIN statement above started with `mvsum_hr01_min00`. This is because we we hard-coded the 9 variables corresponding to the sum of MVPA from the 1st to the 9th minute of the *activity day*:

```
mvsum_hr01_min00 = modvig_hr01_min00;
mvsum_hr01_min01 = sum(modvig_hr01_min00, modvig_hr01_min01);
mvsum_hr01_min02 = sum(modvig_hr01_min00, …, modvig_hr01_min02);
…
mvsum_hr01_min08 = sum(modvig_hr01_min00, …, modvig_hr01_min08);
```

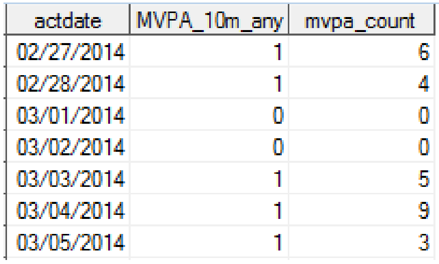
Step 15. Next, we counted the number of non-overlapping successive bouts of 10-minute MVPAs.

```
Data modvig_counts;
   array mvsum [1431]
      mvsum_hr01_min09-mvsum_hr01_min59 …
      mvsum_hr24_min00-mvsum_hr24_min59;
   set modvig_sums (keep = ActDate MPVA_30m_any
      mvsum_hr01_min09-mvsum_hr01_min59 …
      mvsum_hr24_min00-mvsum_hr24_min59);
   i=1;
   j=1;
   mvpa_count = 0;
   do until (i=1440);
      if mvsum(i) = 10 and j >=10 then do;
         mvpa_count +1;
         j=1;
      end;
      else do;
         j+1;
      end;
      i+1;
   end;
run;
```

Step 16. For the final step, we created a limited dataset containing only the derived MVPA variables, and a snapshot of the final data as shown in **Figure 4.**

```
      Data MVPA_10min;
            set modvig_counts (keep = actdate MVPA_10m_any mvpa_count);
      run;
   %mend;
   %mvpa(starthour=3);
```

Figure 4. Final MVPA dataset.



actdate	MVPA_10m_any	mvpa_count
02/27/2014	1	6
02/28/2014	1	4
03/01/2014	0	0
03/02/2014	0	0
03/03/2014	1	5
03/04/2014	1	9
03/05/2014	1	3

THE MVPA MACRO

Below is the full working code that will read the minute-by-minute physical activity intensities CSV file and create a final dataset similar to the one in **Figure 4**.

```
options mprint;


%let intensities = C:\SASGF16\Paper 9881\Minute_Intensities.csv;

/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
/*       This macro passes the number of observation into a macro variable    */
/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
%MACRO NUMOBS (DSN=) ;
       %GLOBAL NUM ;
    DATA  NULL  ;
               IF 0 THEN SET &DSN NOBS=COUNT ;
        CALL SYMPUT ('NUM', LEFT(PUT(COUNT,8.))) ;
        STOP ;
    RUN;
%MEND;

%macro mvpa (starthour=);

   %if &starthour. = 0 %then %do;
        %let endhour = 23; %end;
   %else %do;
        %let endhour = %sysevalf(&starthour. - 1);
   %end;
   %put &starthour. &endhour.;

   /* Import minute-by-minute intensities from CSV */
   proc import out = int
        datafile = "&intensities"
```

```
    dbms = csv replace;
run;

/*Extract date and hour from ActivityHour. */
data int2;
    retain ActDate ActHour;
    set int;
       ActDate = DatePart(ActivityHour);
       ActHour = Hour(TimePart(ActivityHour));
       format ActDate mmddyy10.;
    run;

proc sort data=int2;
    by ActivityHour;
run;

/* Identify first activity date */
data _null_;
    set int2 (firstobs= 1 obs=1);
    call sympút('startdate', compress(ActDate));
run;

/* Identify last activity date */
%NUMOBS (DSN=int2  ) ;
data _null_;
    set int2  (firstobs= &NUM. obs=&NUM.);
    call symput ('enddate',  compress(ActDate));
run;

/* Only retain the rows between the first and last activity day (inclusive)*/
data int3 ;
    retain ActDate ActHour;
       set int2;
       ActDate = DatePart(ActivityHour);
       ActHour = Hour(TimePart(ActivityHour));
       format ActDate mmddyy10.;
       if ActivityHour >= dhms(&startdate., &starthour.,0,0) and
            ActivityHour <= dhms(&enddate.,&endhour.,0,0);
run;

/*Convert dataset to longitudinal version*/
data int_long;
    set int3;
    array mins(0:59) intensity00-intensity59;

    do min = 0 to 59;
       intensity = mins(min);
       output;
    end;

    drop intensity00-intensity59;

run;

data ind;
    set int_long;
    /* This creates an indicator variable with values 1-24 where 1 indicates first
       hour of the defined day */
    temp1 = ActHour - &startHour. + 1;

    if temp1 <=0 then ind_dayhour = 24+temp1;
    else ind_dayhour = temp1;
    drop temp1;
run;

data ind2;
    set ind;
    if min = 0;
    drop min intensity;
run;
```

```
   proc sort data=ind2 ;
        by ind_dayhour actdate;
   run;
/* This step creates an indicator variable that counts the number of days included in the
dataset.  This indicator variable combined with the indicator for # of hours, will be used to
convert the dataset with into a wide */
   data ind3;
        set ind2;
        ind_daynum + 1;
        by ind_dayhour;
        if first.ind_dayhour then ind_daynum= 1;
   run;

        /*Merge back into the intensities dataset*/
   proc sort data=ind3;
        by ActDate ActHour;
   proc sort data=Int_long;
        by ActDate ActHour;
   run;

   Data Intensities ;
        merge ind3 Int_long;
        by ActDate ActHour;

        modvig = (intensity >=2);
        label modvig = "Moderate or Vigorous physical activity (1, intensity >=2; 0, otherwise)";
   run;

/*==============================================================================================*/
/*Convert to wide format: 1 observation per day with 60x24 columns each column                 */
/*IMPORTANT: Make sure to sort by day, then hour and minutes, so that rows are in correct order*/
/*==============================================================================================*/
   proc sort data=intensities;
        by ind_daynum ind_dayhour min;
   run;

   data modvig_wide;
        retain ActDate
               modvig_hr01_min00-modvig_hr01_min59 modvig_hr02_min00-modvig_hr02_min59
               modvig_hr03_min00-modvig_hr03_min59 modvig_hr04_min00-modvig_hr04_min59
               modvig_hr05_min00-modvig_hr05_min59 modvig_hr06_min00-modvig_hr06_min59
               modvig_hr07_min00-modvig_hr07_min59 modvig_hr08_min00-modvig_hr08_min59
               modvig_hr09_min00-modvig_hr09_min59 modvig_hr10_min00-modvig_hr10_min59
               modvig_hr11_min00-modvig_hr11_min59 modvig_hr12_min00-modvig_hr12_min59
               modvig_hr13_min00-modvig_hr13_min59 modvig_hr14_min00-modvig_hr14_min59
               modvig_hr15_min00-modvig_hr15_min59 modvig_hr16_min00-modvig_hr16_min59
               modvig_hr17_min00-modvig_hr17_min59 modvig_hr18_min00-modvig_hr18_min59
               modvig_hr19_min00-modvig_hr19_min59 modvig_hr20_min00-modvig_hr20_min59
               modvig_hr21_min00-modvig_hr21_min59 modvig_hr22_min00-modvig_hr22_min59
               modvig_hr23_min00-modvig_hr23_min59 modvig_hr24_min00-modvig_hr24_min59
        ;
        keep ActDate modvig_hr01_min00-modvig_hr01_min59 modvig_hr02_min00-modvig_hr02_min59
               modvig_hr03_min00-modvig_hr03_min59 modvig_hr04_min00-modvig_hr04_min59
               modvig_hr05_min00-modvig_hr05_min59 modvig_hr06_min00-modvig_hr06_min59
               modvig_hr07_min00-modvig_hr07_min59 modvig_hr08_min00-modvig_hr08_min59
               modvig_hr09_min00-modvig_hr09_min59 modvig_hr10_min00-modvig_hr10_min59
               modvig_hr11_min00-modvig_hr11_min59 modvig_hr12_min00-modvig_hr12_min59
               modvig_hr13_min00-modvig_hr13_min59 modvig_hr14_min00-modvig_hr14_min59
               modvig_hr15_min00-modvig_hr15_min59 modvig_hr16_min00-modvig_hr16_min59
               modvig_hr17_min00-modvig_hr17_min59 modvig_hr18_min00-modvig_hr18_min59
               modvig_hr19_min00-modvig_hr19_min59 modvig_hr20_min00-modvig_hr20_min59
               modvig_hr21_min00-modvig_hr21_min59 modvig_hr22_min00-modvig_hr22_min59
               modvig_hr23_min00-modvig_hr23_min59 modvig_hr24_min00-modvig_hr24_min59
        ;
        array mv [1:24,0:59] modvig_hr01_min00-modvig_hr01_min59
               modvig_hr02_min00-modvig_hr02_min59 modvig_hr03_min00-modvig_hr03_min59
               modvig_hr04_min00-modvig_hr04_min59 modvig_hr05_min00-modvig_hr05_min59
               modvig_hr06_min00-modvig_hr06_min59 modvig_hr07_min00-modvig_hr07_min59
               modvig_hr08_min00-modvig_hr08_min59 modvig_hr09_min00-modvig_hr09_min59
               modvig_hr10_min00-modvig_hr10_min59 modvig_hr11_min00-modvig_hr11_min59
               modvig_hr12_min00-modvig_hr12_min59 modvig_hr13_min00-modvig_hr13_min59
               modvig_hr14_min00-modvig_hr14_min59 modvig_hr15_min00-modvig_hr15_min59
               modvig_hr16_min00-modvig_hr16_min59 modvig_hr17_min00-modvig_hr17_min59
               modvig_hr18_min00-modvig_hr18_min59 modvig_hr19_min00-modvig_hr19_min59
               modvig_hr20_min00-modvig_hr20_min59 modvig_hr21_min00-modvig_hr21_min59
               modvig_hr22_min00-modvig_hr22_min59 modvig_hr23_min00-modvig_hr23_min59
               modvig_hr24_min00-modvig_hr24_min59
       ;
       set Intensities;
       by ind_daynum ;

       if first.ind_daynum then
          do i = 1 to 24;
             do j = 0 to 59;
                mv[i, j] = .;
             end;
          end;

       mv[ind_dayhour, min] = modvig;

       if &starthour. ne 0 then ActDate = Actdate-1;

       if last.ind_daynum then output;

   run;

/*=============================================================================================*/
/* This step creates variables that counts the # minutes of moderate or vigorous activity in the
   last 10 consecutive minutes. For example, mvsum_hr01_min10 is the sum from minute 1 to minute
   10 of hour 1.                                                                               */
/*=============================================================================================*/
   data modvig_sums;
       retain actdate mvsum_hr01_min00-mvsum_hr01_min59 mvsum_hr02_min00-mvsum_hr02_min59
               mvsum_hr03_min00-mvsum_hr03_min59 mvsum_hr04_min00-mvsum_hr04_min59
               mvsum_hr05_min00-mvsum_hr05_min59 mvsum_hr06_min00-mvsum_hr06_min59
               mvsum_hr07_min00-mvsum_hr07_min59 mvsum_hr08_min00-mvsum_hr08_min59
               mvsum_hr09_min00-mvsum_hr09_min59 mvsum_hr10_min00-mvsum_hr10_min59
               mvsum_hr11_min00-mvsum_hr11_min59 mvsum_hr12_min00-mvsum_hr12_min59
               mvsum_hr13_min00-mvsum_hr13_min59 mvsum_hr14_min00-mvsum_hr14_min59
               mvsum_hr15_min00-mvsum_hr15_min59 mvsum_hr16_min00-mvsum_hr16_min59
               mvsum_hr17_min00-mvsum_hr17_min59 mvsum_hr18_min00-mvsum_hr18_min59
               mvsum_hr19_min00-mvsum_hr19_min59 mvsum_hr20_min00-mvsum_hr20_min59
               mvsum_hr21_min00-mvsum_hr21_min59 mvsum_hr22_min00-mvsum_hr22_min59
               mvsum_hr23_min00-mvsum_hr23_min59 mvsum_hr24_min00-mvsum_hr24_min59 ;
       array mv [1440] modvig_hr01_min00-modvig_hr01_min59 modvig_hr02_min00-
               modvig_hr02_min59 modvig_hr03_min00-modvig_hr03_min59
               modvig_hr04_min00-modvig_hr04_min59 modvig_hr05_min00-modvig_hr05_min59
               modvig_hr06_min00-modvig_hr06_min59 modvig_hr07_min00-modvig_hr07_min59
               modvig_hr08_min00-modvig_hr08_min59 modvig_hr09_min00-modvig_hr09_min59
               modvig_hr10_min00-modvig_hr10_min59 modvig_hr11_min00-modvig_hr11_min59
               modvig_hr12_min00-modvig_hr12_min59 modvig_hr13_min00-modvig_hr13_min59
               modvig_hr14_min00-modvig_hr14_min59 modvig_hr15_min00-modvig_hr15_min59
               modvig_hr16_min00-modvig_hr16_min59 modvig_hr17_min00-modvig_hr17_min59
               modvig_hr18_min00-modvig_hr18_min59 modvig_hr19_min00-modvig_hr19_min59
               modvig_hr20_min00-modvig_hr20_min59 modvig_hr21_min00-modvig_hr21_min59
               modvig_hr22_min00-modvig_hr22_min59 modvig_hr23_min00-modvig_hr23_min59
               modvig_hr24_min00-modvig_hr24_min59   ;
       array mvsum [1431] mvsum_hr01_min09-mvsum_hr01_min59 mvsum_hr02_min00-mvsum_hr02_min59
               mvsum_hr03_min00-mvsum_hr03_min59 mvsum_hr04_min00-mvsum_hr04_min59
               mvsum_hr05_min00-mvsum_hr05_min59 mvsum_hr06_min00-mvsum_hr06_min59
               mvsum_hr07_min00-mvsum_hr07_min59 mvsum_hr08_min00-mvsum_hr08_min59
               mvsum_hr09_min00-mvsum_hr09_min59 mvsum_hr10_min00-mvsum_hr10_min59
               mvsum_hr11_min00-mvsum_hr11_min59 mvsum_hr12_min00-mvsum_hr12_min59
               mvsum_hr13_min00-mvsum_hr13_min59 mvsum_hr14_min00-mvsum_hr14_min59
               mvsum_hr15_min00-mvsum_hr15_min59 mvsum_hr16_min00-mvsum_hr16_min59
               mvsum_hr17_min00-mvsum_hr17_min59 mvsum_hr18_min00-mvsum_hr18_min59
               mvsum_hr19_min00-mvsum_hr19_min59 mvsum_hr20_min00-mvsum_hr20_min59
               mvsum_hr21_min00-mvsum_hr21_min59 mvsum_hr22_min00-mvsum_hr22_min59
               mvsum_hr23_min00-mvsum_hr23_min59 mvsum_hr24_min00-mvsum_hr24_min59 ;

       set modvig_wide;
```

```
    MVPA_10m_any = 0;

    /*Count the number moderate/vigorous minutes within 10 consecutive minutes */
    do i= 1 to 1431;
       mvsum[i] = sum( mv[i], mv[i+1], mv[i+2], mv[i+3], mv[i+4], mv[i+5], mv[i+6], mv[i+7],
                   mv[i+8], mv[i+9] );
    /*Create the MVPA binary variable  */
    if mvsum[i] >= 10 then MVPA_10m_any = 1;
    end;

    /*Create placeholder variables for the first 9 minutes of the day*/
    mvsum_hr01_min00 = modvig_hr01_min00;
    mvsum_hr01_min01 = sum(modvig_hr01_min00, modvig_hr01_min01);
    mvsum_hr01_min02 = sum(modvig_hr01_min00, modvig_hr01_min01, modvig_hr01_min02);
    mvsum_hr01_min03 = sum(modvig_hr01_min00, modvig_hr01_min01, modvig_hr01_min02,
       modvig_hr01_min03);
    mvsum_hr01_min04 = sum(modvig_hr01_min00, modvig_hr01_min01, modvig_hr01_min02,
       modvig_hr01_min03, modvig_hr01_min04);
    mvsum_hr01_min05 = sum(modvig_hr01_min00, modvig_hr01_min01, modvig_hr01_min02,
       modvig_hr01_min03, modvig_hr01_min04, modvig_hr01_min05);
    mvsum_hr01_min06 = sum(modvig_hr01_min00, modvig_hr01_min01, modvig_hr01_min02,
       modvig_hr01_min03, modvig_hr01_min04, modvig_hr01_min05, modvig_hr01_min06);
    mvsum_hr01_min07 = sum(modvig_hr01_min00, modvig_hr01_min01, modvig_hr01_min02,
       modvig_hr01_min03, modvig_hr01_min04, modvig_hr01_min05, modvig_hr01_min06,
       modvig_hr01_min07);
    mvsum_hr01_min08 = sum(modvig_hr01_min00, modvig_hr01_min01, modvig_hr01_min02,
       modvig_hr01_min03, modvig_hr01_min04, modvig_hr01_min05, modvig_hr01_min06,
       modvig_hr01_min07, modvig_hr01_min08);
run;

    /*This step counts the number of 10 minute bouts of MVPA */
Data modvig_counts;
    array mvsum[1440] mvsum_hr01_min00-mvsum_hr01_min59 mvsum_hr02_min00-mvsum_hr02_min59
            mvsum_hr03_min00-mvsum_hr03_min59 mvsum_hr04_min00-mvsum_hr04_min59
            mvsum_hr05_min00-mvsum_hr05_min59 mvsum_hr06_min00-mvsum_hr06_min59
            mvsum_hr07_min00-mvsum_hr07_min59 mvsum_hr08_min00-mvsum_hr08_min59
            mvsum_hr09_min00-mvsum_hr09_min59 mvsum_hr10_min00-mvsum_hr10_min59
            mvsum_hr11_min00-mvsum_hr11_min59 mvsum_hr12_min00-mvsum_hr12_min59
            mvsum_hr13_min00-mvsum_hr13_min59 mvsum_hr14_min00-mvsum_hr14_min59
            mvsum_hr15_min00-mvsum_hr15_min59 mvsum_hr16_min00-mvsum_hr16_min59
            mvsum_hr17_min00-mvsum_hr17_min59 mvsum_hr18_min00-mvsum_hr18_min59
            mvsum_hr19_min00-mvsum_hr19_min59 mvsum_hr20_min00-mvsum_hr20_min59
            mvsum_hr21_min00-mvsum_hr21_min59 mvsum_hr22_min00-mvsum_hr22_min59
            mvsum_hr23_min00-mvsum_hr23_min59 mvsum_hr24_min00-mvsum_hr24_min59 ;
    set modvig_sums (keep = ActDate MVPA_10m_any mvsum_hr01_min00-mvsum_hr01_min59
            mvsum_hr02_min00-mvsum_hr02_min59 mvsum_hr03_min00-mvsum_hr03_min59
            mvsum_hr04_min00-mvsum_hr04_min59 mvsum_hr05_min00-mvsum_hr05_min59
            mvsum_hr06_min00-mvsum_hr06_min59 mvsum_hr07_min00-mvsum_hr07_min59
            mvsum_hr08_min00-mvsum_hr08_min59 mvsum_hr09_min00-mvsum_hr09_min59
            mvsum_hr10_min00-mvsum_hr10_min59 mvsum_hr11_min00-mvsum_hr11_min59
            mvsum_hr12_min00-mvsum_hr12_min59 mvsum_hr13_min00-mvsum_hr13_min59
            mvsum_hr14_min00-mvsum_hr14_min59 mvsum_hr15_min00-mvsum_hr15_min59
            mvsum_hr16_min00-mvsum_hr16_min59 mvsum_hr17_min00-mvsum_hr17_min59
            mvsum_hr18_min00-mvsum_hr18_min59 mvsum_hr19_min00-mvsum_hr19_min59
            mvsum_hr20_min00-mvsum_hr20_min59 mvsum_hr21_min00-mvsum_hr21_min59
            mvsum_hr22_min00-mvsum_hr22_min59 mvsum_hr23_min00-mvsum_hr23_min59
            mvsum_hr24_min00-mvsum_hr24_min59    );
    /*This counts the # of non-overlapping, SUCCESSIVE bouts of 10 MVPA minutes*/
            i=1;
            j=1;
            mvpa_count = 0;
            do until (i=1440);
               if mvsum(i) = 10 and j >=10 then do;
                    mvpa_count +1;
                    j=1;
                     end;
            else do;
               j+1;
               end;

            i+1;
```

```
               end;
   run;
/*The final step creates dataset containing only the MVPA variables*/
   Data MVPA_10min;
       set modvig_counts (keep = actdate MVPA_10m_any mvpa_count);
   run;
%mend;

/*Call the macro and assign 3:00 am as the start of the activity day*/
%mvpa(starthour=3);
```

CONCLUSION

The availability of fashionable and inexpensive wearable fitness trackers has provided researchers and data analysts with new opportunities and challenges to tap into. At CBCH, we created a SAS macro that followed a 16 step procedure to calculate the number of 10-min. MVPA bouts per day, an analyzable measure of physical activity from minute-by-minute raw data that is pertinent to determining if an individual is meeting current physical activity guidelines (e.g., 150 min. of MVPA/week performed in bouts of at least 10 min.). Our code incorporates a logical sequence of steps that can be easily modified to adjust the calculation of MVPA bouts to a different length of time. Additionally, the code highlights the usefulness of SAS arrays in creating and manipulating thousands of variables just with a few lines of code.

REFERENCES

Fitbit Inc. 2016. “Activity & Exercise Logs”. Accessed March 1, 2016. https://dev.fitbit.com/docs/activity/#get-activity-intraday-time-series.

Small Steps Labs, LLC. 2016. “Fitabase”. Accessed March 1, 2016. https://www.fitabase.com.

Li, Arthur X. 2013. “Sharpening Your Skills in Reshaping data: PROC TRANSPOSE vs. Array Processing”. *Proceedings of the SAS Global Forum 2013,* San Francisco, CA. Available at: http://support.sas.com/resources/papers/proceedings13/082-2013.pdf.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Faith E.Parsons
Center for Behavioral Cardiovascular Health
Columbia University Medical Center
E-mail: fep2110@cumc.columbia.edu
http://linkedin.com/in/faithparsons

Keith M. Diaz, PhD
Center for Behavioral Cardiovascular Health
Columbia University Medical Center
E-mail: kd2442@cumc.columbia.edu

Jacob E. Julian
Center for Behavioral Cardiovascular Health
Columbia University Medical Center
E-mail: jej2140@cumc.columbia.edu