NHANES Data Tutorial

Erjia Cui

10/2021

Introduction

This tutorial illustrates how to start working with data collected from 2003-2004 ("C") and 2005-2006 ("D") waves of National Health and Nutrition Examination Survey (NHANES), one of the largest study conducted by CDC to assess the health and nutritional status of US population. Specifically, we focus on (1) activity count data collected from accelerometers of hip-worn devices (ActiGraph AM-7164), (2) mortality data collected from National Death Index and linked to NHANES, and (3) demographic, survey design, lifestyle, and comorbiditiy data. These data have been processed and stored in the `rnhanesdata` package. The processing pipeline can be found here.

In this tutorial, we assume that the `rnhanesdata` package has been successfully installed in your working environment. The package installation tutorial can be found here. The name of the processed data are listed as follows:

- Activity count data: PAXINTEN_C, PAXINTEN_D, FLAG_C, FLAG_D.
- Mortality data: Mortality_2015_C, Mortality_2015_D.
- Demographic, survey design, lifestyle, and comorbiditiy data: Covariate_C, Covariate_D.

For activity count data, "PAXINTEN_*" are matrices of count values and "FLAG_*" are the associated wear/non-wear flags. It is worth noting that the wear/non-wear status was classified based on the count values.

Load Data

```
## load package
library(rnhanesdata)
```

Since all processed data have been stored in the `rnhanesdata` package, all we need to do is to load this package. We can also download raw data of other categories from NHANES website and combine them with the processed data when needed.

Check the Data Storage Format

Data collected from C and D waves were processed into exactly the same format, which helps to combine these two waves in the following analysis. This decision was made because these two waves share very similar protocol.

Activity Count Data

```r
## check the storage format of activity count data
dim(PAXINTEN_C)
```

```
## [1] 50232  1445
```

```r
head(PAXINTEN_C[,1:10], n = 10)
```

```
##     SEQN PAXCAL PAXSTAT WEEKDAY SDDSRVYR MIN1 MIN2 MIN3 MIN4 MIN5
## 1  21005      1       1       1        3    0    0    0    0    0
## 2  21005      1       1       2        3    0    0    0    0    0
## 3  21005      1       1       3        3    0    0    0    0    0
## 4  21005      1       1       4        3    0    0    0    0    0
## 5  21005      1       1       5        3    0    0    0    0    0
## 6  21005      1       1       6        3    0    0    0    0    0
## 7  21005      1       1       7        3    0    0    0    0    0
## 8  21006      1       1       1        3   75  690  108   56   47
## 9  21006      1       1       2        3    0   10    0    0    0
## 10 21006      1       1       3        3    0    0    0    0    0
```

```r
table(PAXINTEN_C$WEEKDAY)
```

```
##
##    1    2    3    4    5    6    7
## 7176 7176 7176 7176 7176 7176 7176
```

```r
dim(PAXINTEN_D)
```

```
## [1] 52185  1445
```

```r
table(PAXINTEN_D$WEEKDAY)
```

```
##
##    1    2    3    4    5    6    7
## 7455 7455 7455 7455 7455 7455 7455
```

There were 7176 participants from C wave and 7455 participants from D wave who wore the device, leading to a total of 14631 study participants. For each eligible study participant, we have 7 consecutive days of activity count data, which can start from any day of the week. Each day is defined as from midnight to midnight. The meaning of each column is explained below:

- `SEQN`: Respondent sequence number (unique identifier).
- `PAXCAL`: Monitor calibration indicator. `PAXCAL = 1` for calibrated data.
- `PAXSTAT`: Data reliability status. `PAXSTAT = 1` for data deemed reliable.
- `WEEKDAY`: Day of the week. `WEEKDAY = 1` for Sunday, `WEEKDAY = 2` for Monday, etc. The start date of wearing the device is not necessarily Sunday.
- `SDDSRVYR`: Two-year data release cycle number. `SDDSRVYR = 3` for 2003-2004 ("C") wave, and `SDDSRVYR = 4` for 2005-2006 ("D") wave.
- `MIN*`: Activity count value at each minute of a day. `MIN1`: 12:00AM-12:01AM, etc.

Mortality Data

```
## check the storage format of mortality data
dim(Mortality_2015_C)
```

```
## [1] 10122     8
```

```
head(Mortality_2015_C)
```

```
##    SEQN eligstat mortstat permth_exm permth_int ucod_leading diabetes_mcod
## 1 21005        1        0        150        150         <NA>            NA
## 2 21006        2       NA         NA         NA         <NA>            NA
## 3 21007        2       NA         NA         NA         <NA>            NA
## 4 21008        2       NA         NA         NA         <NA>            NA
## 5 21009        1        0        135        135         <NA>            NA
## 6 21010        1        0        149        149         <NA>            NA
##   hyperten_mcod
## 1            NA
## 2            NA
## 3            NA
## 4            NA
## 5            NA
## 6            NA
```

```
table(Mortality_2015_C$mortstat)
```

```
## 
##    0    1 
## 4516 1094
```

```
dim(Mortality_2015_D)
```

```
## [1] 10348     8
```

```
table(Mortality_2015_D$mortstat)
```

```
## 
##    0    1 
## 4786  774
```

The number of rows of mortality data is the sample size of each wave. We have a total of 10122 participants enrolled in C wave and 10348 participants enrolled in D wave, respectively. Notice that not all study participants wore the device, which explains why we only have activity count data from around 71% of the population in each wave. In addition, not all study participants were linked to mortality data and the eligibility was labeled in the `eligstat` column. The meaning of each column is explained below:

- `SEQN`: Respondent sequence number (unique identifier).
- `eligstat`: Mortality linkage eligibility. `eligstat = 1` indicates that the survey participant was eligible for the mortality linkage.

- `mortstat`: Vital status code. `mortstat = 1` if assumed deceased and `mortstat = 0` if assumed alive.
- `permth_exm`: Person months of follow-up from MEC/Exam date.
- `permth_int`: Person months of follow-up from interview date.
- `ucod_leading`: Leading cause of death (code).
- `diabetes_mcod`: Diabetes flag from multiple cause of death.
- `hyperten_mcod`: Hypertension flag from multiple cause of death.

Covariate Data

```
## check the storage format of covariates
dim(Covariate_C)
```

```
## [1] 10122    23
```

```
head(Covariate_C)
```

```
##    SEQN SDDSRVYR SDMVPSU SDMVSTRA  WTINT2YR  WTMEC2YR RIDAGEMN RIDAGEEX
## 1 21005        3       2       39  5512.321  5824.782      232      233
## 2 21006        3       1       41  5422.140  5564.040      203      205
## 3 21007        3       2       35 39764.177 40591.066      172      172
## 4 21008        3       1       32  5599.499  5696.751      208      209
## 5 21009        3       2       31 97593.679 97731.727      671      672
## 6 21010        3       1       29 39599.363 43286.576      633      634
##   RIDAGEYR   BMI     BMI_cat  Race Gender Diabetes  CHF  CHD Cancer Stroke
## 1       19 50.85       Obese Black   Male       No <NA> <NA>   <NA>   <NA>
## 2       16 20.78      Normal Black Female       No <NA> <NA>   <NA>   <NA>
## 3       14 18.43 Underweight White Female       No <NA> <NA>   <NA>   <NA>
## 4       17 20.65      Normal Black   Male       No <NA> <NA>   <NA>   <NA>
## 5       55 31.26       Obese White   Male       No   No   No     No     No
## 6       52 25.49  Overweight White Female       No   No   No     No     No
##                     EducationAdult MobilityProblem   DrinkStatus
## 1                             <NA>            <NA>          <NA>
## 2                             <NA>            <NA>          <NA>
## 3                             <NA>            <NA>          <NA>
## 4                             <NA>            <NA>          <NA>
## 5 High school grad/GED or equivalent   No Difficulty   Non-Drinker
## 6          Some College or AA degree   No Difficulty Heavy Drinker
##   DrinksPerWeek SmokeCigs
## 1            NA      <NA>
## 2            NA      <NA>
## 3            NA      <NA>
## 4            NA      <NA>
## 5             0     Never
## 6            28   Current
```

```
dim(Covariate_D)
```

```
## [1] 10348    23
```

The number of rows of covariate data is the same as that of mortality data. For some variables we have missing values. The variables can be classified into several categories:

- Demographic data: SEQN, RIDAGEMN, RIDAGEEX, RIDAGEYR, BMI, BMI_cat, Race, Gender, EducationAdult, MobilityProblem.
- Survey design: SDDSRVYR, SDMVPSU, SDMVSTRA, WTINT2YR, WTMEC2YR.
- Comorbidity: Diabetes, CHF, CHD, Cancer, Stroke.
- Lifestyle: DrinkStatus, DrinksPerWeek, SmokeCigs.

For some demographic and survey design variables with unintuitive capitalized names, we can find their actual meaning on the NHANES website. The meaning of the other columns is easier to understand. It is worth noting that NHANES study has many other types of data that are not limited to those shown above. Other types of data can be downloaded from the website and integrated with existing data using the same processing pipeline.

Data Cleaning

Since the purpose of this tutorial is to show **how to start** working with these large-scale, multilevel, high-dimensional, survey-weighted, and publicly available data, we only show necessary cleaning steps before combining these data into an analyzable format. For different projects, it is recommended to set your own data exclusion criteria and do further data cleaning accordingly.

```r
## load tidyverse package for data cleaning
library(tidyverse)

## change activity count value under non-wear flags to 0
PAXINTEN_C[,paste0("MIN",1:1440)] <- PAXINTEN_C[,paste0("MIN",1:1440)]*
  Flags_C[,paste0("MIN",1:1440)]
PAXINTEN_D[,paste0("MIN",1:1440)] <- PAXINTEN_D[,paste0("MIN",1:1440)]*
  Flags_D[,paste0("MIN",1:1440)]

## merge mortality and covariate data
mort_cov_C <- inner_join(Mortality_2015_C, Covariate_C, by = "SEQN")
mort_cov_D <- inner_join(Mortality_2015_D, Covariate_D, by = "SEQN")

## combine data collected from two waves
mort_cov <- bind_rows(mort_cov_C, mort_cov_D)
act_cnt <- bind_rows(PAXINTEN_C, PAXINTEN_D)
wear_flag <- bind_rows(Flags_C, Flags_D)
rm(mort_cov_C, mort_cov_D)

## create Age (in years) using the age at examination
mort_cov$Age <- mort_cov$RIDAGEEX/12
```

After these cleaning steps, there remains two major differences between the activity count data and the rest: (1) each row of activity count data is one participant-day, while each row of the other data is one participant; (2) each row of activity count data has minute-level values, which is high-dimensional. While such multilevel, high-dimensional structure itself leads to many interesting statistical research questions, in some other cases people prefer to have simple, participant-level summary measures to work with.

To solve challenge (2), we follow the literature and create several physical activity summary variables. To solve challenge (1), we compress the activity count data to participant-level by taking the average of (i) activity count value at each minute of a day, and (ii) summary variables, across "good days". A day is defined as an good day if: (i) it has estimated wear time of over 10 hours, (ii) the data are calibrated, and (iii) the data are deemed reliable by NHANES. We also only compress activity count data for participants with at least 3 good days. Notice that the purpose of such compression is to integrate activity count data with other participant-level data so that we have a single data frame with non-redundant information. In other words, not all of the following steps are necessary if we are only interested in the multilevel data. However, it is still recommended to use only "good days" in the multilevel data analysis.

Further Cleaning of Activity Count Data

```
## extract count values and flags as matrices
cnt_mat <- as.matrix(act_cnt[,paste0("MIN",1:1440)])
flag_mat <- as.matrix(wear_flag[,paste0("MIN",1:1440)])

## replace NAs with 0s
cnt_mat[is.na(cnt_mat)] <- 0
flag_mat[is.na(flag_mat)] <- 0

## calculate activity count summary measures
### total activity count (TAC)
act_cnt$TAC <- rowSums(cnt_mat)
### total log activity count (TLAC)
act_cnt$TLAC <- rowSums(log(1+cnt_mat))
### total wear time (WT)
act_cnt$WT <- rowSums(flag_mat)
### total sedentary time (ST)
act_cnt$ST <- rowSums(cnt_mat < 100) ## threshold set based on the literature
### total moderate to vigorous physical activity time (MVPA)
act_cnt$MVPA <- rowSums(cnt_mat >= 2020) ## threshold set based on the literature

## create "good day" indicator
act_cnt$goodday <- ifelse(act_cnt$PAXCAL == 1 & act_cnt$PAXSTAT == 1 &
                            act_cnt$WT >= 600, 1, 0)

## store the minute-level activity count data as a column of the data frame
act_cnt$AC <- I(cnt_mat)

## clean the multilevel activity count data
act_cnt_ml <- act_cnt %>% filter(goodday == 1) %>%
  select("SEQN", "SDDSRVYR", "WEEKDAY",
         "AC", "TAC", "TLAC", "ST", "MVPA", "WT",
         "PAXCAL", "PAXSTAT", "goodday")

## add number of good days for each participant
act_cnt_ml <- left_join(act_cnt_ml, act_cnt_ml %>% count(SEQN) %>%
                        mutate(n_good_days = n) %>% select(SEQN, n_good_days),
                        by = "SEQN")
dim(act_cnt_ml)
```

```
## [1] 65777    13
```

```
str(act_cnt_ml)
```

```
## 'data.frame':    65777 obs. of  13 variables:
##  $ SEQN       : int  21005 21005 21005 21006 21006 21006 21006 21007 21007 21007 ...
```

```
##  $ SDDSRVYR   : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ WEEKDAY    : int  4 6 7 1 2 6 7 1 2 3 ...
##  $ AC         : 'AsIs' num [1:65777, 1:1440] 0 0 0 75 0 273 138 0 0 0 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:1440] "MIN1" "MIN2" "MIN3" "MIN4" ...
##  $ TAC        : num  851531 249123 469072 217784 69560 ...
##  $ TLAC       : num  3836 2350 3401 3049 1112 ...
##  $ ST         : num  981 1153 1083 1081 1309 ...
##  $ MVPA       : num  195 31 96 10 1 24 1 56 48 36 ...
##  $ WT         : num  873 681 875 961 712 615 727 737 910 941 ...
##  $ PAXCAL     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ PAXSTAT    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ goodday    : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ n_good_days: int  3 3 3 4 4 4 4 7 7 7 ...
```

```
rm(act_cnt, wear_flag, cnt_mat, flag_mat)
```

The data frame `act_cnt_ml` contains cleaned multilevel activity count data in NHANES. This is the data we want to use for **multilevel** statistical modeling.

Compression of Activity Count Data

```
## select variables of activity count data to compress
act_cnt_ml2sl <- act_cnt_ml %>% filter(n_good_days >= 3) %>%
  select(SEQN, AC, TAC, TLAC, ST, MVPA, WT, n_good_days)

## compress activity count data into participant level
act_cnt_sl <- aggregate(act_cnt_ml2sl[,3:ncol(act_cnt_ml2sl)],
                        list(SEQN = act_cnt_ml2sl$SEQN), mean)
### for the count value matrix stored in "AC" column, we have to do aggregation manually
inx_row <- split(1:nrow(act_cnt_ml2sl), f = factor(act_cnt_ml2sl$SEQN))
act_cnt_sl$AC <- I(t(vapply(inx_row, function(x)
                   colMeans(act_cnt_ml2sl$AC[x,,drop=FALSE],na.rm=TRUE),
                   numeric(ncol(act_cnt_ml2sl$AC)))))
dim(act_cnt_sl)
```

```
## [1] 11201     8
```

```
str(act_cnt_sl)
```

```
## 'data.frame':    11201 obs. of  8 variables:
##  $ SEQN       : int  21005 21006 21007 21008 21009 21010 21012 21015 21017 21019 ...
##  $ TAC        : num  523242 117241 361203 254120 409353 ...
##  $ TLAC       : num  3196 1635 3600 2113 3522 ...
##  $ ST         : num  1072 1253 995 1196 998 ...
##  $ MVPA       : num  107.3 9 37 30.5 48.3 ...
##  $ WT         : num  810 754 932 804 900 ...
##  $ n_good_days: num  3 4 7 4 7 7 7 7 6 6 ...
##  $ AC         : 'AsIs' num [1:11201, 1:1440] 0 121.5 5 97.5 0 ...
```

```
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:11201] "21005" "21006" "21007" "21008" ...
##   .. ..$ : chr [1:1440] "MIN1" "MIN2" "MIN3" "MIN4" ...
```

```
rm(act_cnt_ml2sl, inx_row)
```

The data frame act_cnt_sl contains participant-level activity count data. We next merge activity count data with mortality and covariate data.

```
## merge activity count data and other data
data_analysis <- left_join(mort_cov, act_cnt_sl, by = "SEQN")
dim(data_analysis)
```

```
## [1] 20470    38
```

```
str(data_analysis)
```

```
## 'data.frame':    20470 obs. of  38 variables:
##  $ SEQN           : int  21005 21006 21007 21008 21009 21010 21011 21012 21013 21014 ...
##  $ eligstat       : int  1 2 2 2 1 1 2 1 2 2 ...
##  $ mortstat       : int  0 NA NA NA 0 0 NA 1 NA NA ...
##  $ permth_exm     : int  150 NA NA NA 135 149 NA 127 NA NA ...
##  $ permth_int     : int  150 NA NA NA 135 149 NA 128 NA NA ...
##  $ ucod_leading   : chr  NA NA NA NA ...
##  $ diabetes_mcod  : int  NA NA NA NA NA NA NA 0 NA NA ...
##  $ hyperten_mcod  : int  NA NA NA NA NA NA NA 0 NA NA ...
##  $ SDDSRVYR       : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ SDMVPSU        : num  2 1 2 1 2 1 2 2 1 1 ...
##  $ SDMVSTRA       : num  39 41 35 32 31 29 40 33 37 33 ...
##  $ WTINT2YR       : num  5512 5422 39764 5599 97594 ...
##  $ WTMEC2YR       : num  5825 5564 40591 5697 97732 ...
##  $ RIDAGEMN       : num  232 203 172 208 671 633 3 765 163 42 ...
##  $ RIDAGEEX       : num  233 205 172 209 672 634 4 766 164 42 ...
##  $ RIDAGEYR       : num  19 16 14 17 55 52 0 63 13 3 ...
##  $ BMI            : num  50.9 20.8 18.4 20.6 31.3 ...
##  $ BMI_cat        : Factor w/ 4 levels "Normal","Underweight",..: 4 1 2 1 4 3 NA 1 1 2 ...
##  $ Race           : Factor w/ 5 levels "White","Mexican American",..: 4 4 1 4 1 1 2 4 4 4 ...
##  $ Gender         : Factor w/ 2 levels "Male","Female": 1 2 2 1 1 2 1 1 2 1 ...
##  $ Diabetes       : Factor w/ 5 levels "No","Yes","Borderline",..: 1 1 1 1 1 1 NA 1 1 1 ...
##  $ CHF            : Factor w/ 4 levels "No","Yes","Refused",..: NA NA NA NA 1 1 NA 1 NA NA ...
##  $ CHD            : Factor w/ 4 levels "No","Yes","Refused",..: NA NA NA NA 1 1 NA 1 NA NA ...
##  $ Cancer         : Factor w/ 4 levels "No","Yes","Refused",..: NA NA NA NA 1 1 NA 1 NA NA ...
##  $ Stroke         : Factor w/ 4 levels "Yes","No","Refused",..: NA NA NA NA 2 2 NA 2 NA NA ...
##  $ EducationAdult : Factor w/ 7 levels "Less than 9th grade",..: NA NA NA NA 3 4 NA 3 NA NA ...
##  $ MobilityProblem: Factor w/ 2 levels "No Difficulty",..: NA NA NA NA 1 1 NA 2 NA NA ...
##  $ DrinkStatus    : Factor w/ 3 levels "Moderate Drinker",..: NA NA NA NA 2 3 NA NA NA NA ...
##  $ DrinksPerWeek  : num  NA NA NA NA 0 28 NA NA NA NA ...
##  $ SmokeCigs      : Factor w/ 3 levels "Never","Former",..: NA NA NA NA 1 3 NA 3 NA NA ...
##  $ Age            : num  19.4 17.1 14.3 17.4 56 ...
##  $ TAC            : num  523242 117241 361203 254120 409353 ...
##  $ TLAC           : num  3196 1635 3600 2113 3522 ...
##  $ ST             : num  1072 1253 995 1196 998 ...
```

```
## $ MVPA          : num  107.3 9 37 30.5 48.3 ...
## $ WT            : num  810 754 932 804 900 ...
## $ n_good_days   : num  3 4 7 4 7 7 NA 7 NA NA ...
## $ AC            : 'AsIs' num [1:20470, 1:1440] 0 121.5 5 97.5 0 ...
##  ..- attr(*, "dimnames")=List of 2
##  .. ..$ : chr [1:20470] "21005" "21006" "21007" "21008" ...
##  .. ..$ : chr [1:1440] "MIN1" "MIN2" "MIN3" "MIN4" ...
```

The data frame `data_analysis` contains cleaned activity count data, mortality data, and covariate data of **all study participants** in the NHANES 2003-2004 ("C") and 2005-2006 ("D") waves. Each row represents one study participant. The proportion of missing data varies by type. For different research questions, it is recommended to set corresponding exclusion criteria.

The NHANES data is now ready to use. Enjoy!