

Introduction to downloading and working with NHANES 2003-2004 and 2005-2006 accelerometry data

Source: vignettes/NHANES_accelerometry_introduction.Rmd (https://github.com/andrew-leroux/rnhanesdata/blob/master/vignettes/NHANES_accelerometry_introduction.Rmd)

This vignette illustrates how to download and work NHANES 2003-2004 and 2005-2006 accelerometry data. We also illustrate how to access the prediction performance of objective physical activity measures and their ranking relative to other established predictors of 5-year all-cause mortality in the US. In this analysis, we use participants between 50 and 85 years old from the 2003-2004 and 2005-2006 samples of the National Health and Nutritional Examination Survey (NHANES, n=2978, number of deaths in 5 years=297) who wore a hip-worn accelerometer in the free living environment for up to 7 days.

Note: the html file displays only partial R code necessary to illustrate the functions and arguments used to execute the analysis. Full code is available in the corresponding .Rmd file and can be displayed by setting `echo = TRUE` in the R chunk option (e.g., `{r, echo = TRUE}`).

Prerequisites

The following packages will be used in this vignette to provide illustration of NHANES 5-year mortality prediction model.

```
pckgs <- c("knitr", "kableExtra",          ## packages used for creating Tables
           "devtools",                    ## package used to download R packages stored on GitHub
           "ggplot2", "gridExtra",        ## packages for plotting
           "corrplot",                    ## for correlation plot
           "reshape2",                    ## for transforming data long -> wide and wide -> long
           "dplyr",                        ## packages for merging/transforming data
           "survey",                       ## package used for analyzing complex survey data in R
           "mgcv", "refund"                ## packages used for smoothing/fpca
           )
sapply(pckgs, function(x) if(!require(x, character.only=TRUE, quietly=TRUE)) {
  install.packages(x)
  require(x, character.only=TRUE)
})
rm(list=c("pckgs"))

#rnhanes data package from github
if(!require("rnhanesdata")){
  install_github("andrew-leroux/rnhanesdata")
  require("rnhanesdata")
}

RNGkind(sample.kind="Rounding")
```

NHANES 2003-2004 and 2005-2006 5-year mortality model data selection

We start by downloading NHANES 2003-2004 and 2005-2006 cohorts' information, processing data and combining survey weights for the two cohorts using R package *rnhanesdata* (Leroux, 2018). Items 1 - 11 illustrate processing steps we performed to obtain the subset of NHANES data that meets this analysis criteria, and derive activity summaries. These steps produce an R object called "data_analysis", which is used in 5-year mortality prediction model.

Note: data dictionary and description of variable names is available in [NHANES_processed.pdf](#) document

1. Download and process NHANES lab measurements from the 2 cohorts: systolic blood pressure readings, total cholesterol, and HDL cholesterol. These data are saved in the temporary directory, which is then removed once the covariates information is processed. Blood pressure is recorded up to 4 times here for each participant. The data collection procedure and description of cholesterol and blood pressure variables is available at <https://wwwn.cdc.gov/nchs/nhanes/search/datapage.aspx?Component=Examination&CycleBeginYear=2003> (<https://wwwn.cdc.gov/nchs/nhanes/search/datapage.aspx?Component=Examination&CycleBeginYear=2003>) for the wave 2003-2004 and at <https://wwwn.cdc.gov/nchs/nhanes/search/datapage.aspx?Component=Examination&CycleBeginYear=2005> (<https://wwwn.cdc.gov/nchs/nhanes/search/datapage.aspx?Component=Examination&CycleBeginYear=2005>) for the wave 2005-2006.

```

# Create a (local) temporary directory
# where lab measurement (cholesterol, blood pressure) data will be downloaded from the CDC website
# and then loaded into R. These files need to be downloaded separately as
# the raw files associated with these lab measurements are not included in the rnhanesdata package.
dir_tmp = tempfile()
dir.create(dir_tmp)

if (!dir.exists(dir_tmp)){
  dir.create(dir_tmp, showWarnings = FALSE)
}
dl_file = function(url) {
  bn = basename(url)
  destfile = file.path(dir_tmp, bn)
  if (!file.exists(destfile)) {
    out = download.file(url, destfile = destfile, mode="wb")
  }
  stopifnot(file.exists(destfile))
}

## download the lab measurement data for the cohort 2003-2004
# Cholesterol - Total & HDL: LBXTC and LBXHDD
dl_file("https://wwwn.cdc.gov/Nchs/Nhanes/2003-2004/L13_C.XPT")
# Blood Pressure: BPXSY1 , BPXSY2, BPXSY3 and BPXSY4
dl_file("https://wwwn.cdc.gov/Nchs/Nhanes/2003-2004/BPX_C.XPT")

## download the lab measurement data for the cohort 2005-2006
# Total Cholesterol: LBXTC
dl_file("https://wwwn.cdc.gov/Nchs/Nhanes/2005-2006/TCHOL_D.XPT")
# HDL Cholesterol: LBDHDD
dl_file("https://wwwn.cdc.gov/Nchs/Nhanes/2005-2006/HDL_D.XPT")
# Blood Pressure, up to 4 measurements per person: BPXSY1 , BPXSY2, BPXSY3 and BPXSY4
dl_file("https://wwwn.cdc.gov/Nchs/Nhanes/2005-2006/BPX_D.XPT")

varnames <- c("LBXTC", "LBXHDD", "LBDHDD",          ## 1. cholesterol. Note LBXHDD and LBDHDD are the same variable,
              ## but different names for 2003-2004 and 2005-2006 cohorts
              "BPXSY1", "BPXSY2", "BPXSY3", "BPXSY4" ## 2. blood pressure measurements
              )

## load and merge the lab data
lab_data <- process_covar(varnames=varnames, localpath=dir_tmp)

## change column name for cholesterol variable that changed names
colnames(lab_data$Covariate_C)[colnames(lab_data$Covariate_C) == "LBXHDD"] <- "LBDHDD"

## combine waves
CVMarkers <- bind_rows(lab_data$Covariate_C, lab_data$Covariate_D)

rm(list=c("lab_data", "dir_tmp", "varnames"))

```

2. Load minute-level activity data and combine it with lab measurements, survey sampling data, and mortality data which are included in the *rnhanesdata* package.

```
## load the data
data("PAXINTEN_C");data("PAXINTEN_D") #processed physical activity data 2003-2004 and 2005-2006, respectively
data("Flags_C");data("Flags_D")
data("Mortality_2015_C");data("Mortality_2015_D")
data("Covariate_C");data("Covariate_D")

## re-code activity counts which are considered "non-wear" to be 0
## this doesn't impact many data points, most estimated non-wear times correspond to 0 counts
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 covariate, mortality, and accelerometry data
## note that both PAXINTEN_* and Covariate_* have a column
## called "SDDSRVYR" indicating which NHANES wave the data is associated with.
## To avoid duplicating this column in the merged data, we add this variable to the "by"
## argument in left_join()
AllAct_C <- left_join(PAXINTEN_C, Mortality_2015_C, by = "SEQN") %>%
  left_join(Covariate_C, by=c("SEQN", "SDDSRVYR"))
AllAct_D <- left_join(PAXINTEN_D, Mortality_2015_D, by = "SEQN") %>%
  left_join(Covariate_D, by=c("SEQN", "SDDSRVYR"))

AllFlags_C <- left_join(Flags_C, Mortality_2015_C, by = "SEQN") %>%
  left_join(Covariate_C, by=c("SEQN", "SDDSRVYR"))
AllFlags_D <- left_join(Flags_D, Mortality_2015_D, by = "SEQN") %>%
  left_join(Covariate_D, by=c("SEQN", "SDDSRVYR"))

## clean up the workspace for memory purposes
rm(list=c(paste0(c("PAXINTEN_", "Covariate_", "Mortality_2015_", "Flags_"),rep(LETTERS[3:4],each=4))))

## combine data for the two waves
AllAct <- bind_rows(AllAct_C,AllAct_D)
AllFlags <- bind_rows(AllFlags_C,AllFlags_D)

#merge with cardiovascular markers
AllAct <- left_join(AllAct, CVMarkers, by = "SEQN")
AllFlags <- left_join(AllFlags, CVMarkers, by = "SEQN")

## clean up the workspace again
rm(list=c("AllAct_C","AllAct_D","AllFlags_C","AllFlags_D","CVMarkers"))
```

Activity data has the following format

```
kable(head(AllAct)[,1:10])%>%
  kable_styling("striped", full_width = F)
```

SEQN	PAXCAL	PAXSTAT	WEEKDAY	SDDSRVYR	MIN1	MIN2	MIN3	MIN4	MIN5
21005	1	1	1	3	0	0	0	0	0
21005	1	1	2	3	0	0	0	0	0
21005	1	1	3	3	0	0	0	0	0
21005	1	1	4	3	0	0	0	0	0
21005	1	1	5	3	0	0	0	0	0
21005	1	1	6	3	0	0	0	0	0

Variable descriptions can be found by calling accelerometry data help files

```
?PAXINTEN_C
?PAXINTEN_D
```

Here,

SEQN: Respondent sequence number

WEEKDAY: Day of the week; WEEKDAY=1 for Sunday, 2 for Monday and so forth

PAXCAL: Denotes whether the monitor was in calibration when it was returned by the subject. The data for monitors that were out of calibration (PAXCAL=2) may be less reliable.

PAXSTAT: Component status code with PAXSTAT=1 for records with data that are deemed reliable. A PAXSTAT=2 was used to code records that had some questionable data; analysts may wish to examine these records more closely.

SDDSRVYR: Variable indicating which wave of the NHANES study this data is associated with. For example, SDDSRVYR = 3 corresponds to the 2003-2004 wave and SDDSRVYR = 4 corresponds to the 2005-2006 wave.

MIN1 - MIN1440: minute-level activity step count summary, where MIN1 corresponds to 12:00am, MIN2 to 12:01am and so on.

We can extract and plot participants 31128 and 31193 activity data

```
df.act <- subset(AllAct, SEQN %in% c(31128, 31193))

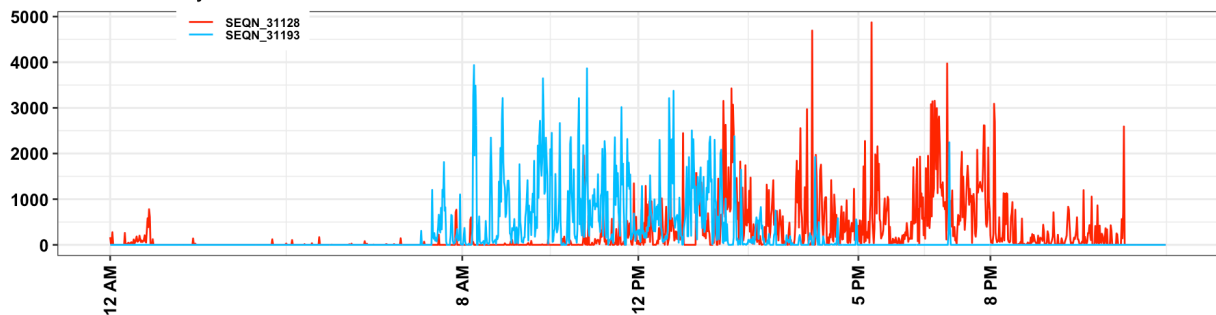
kable(df.act[,1:10],format = "html", caption = "First 5 minutes for two participants activity data",
      row.names = FALSE) %>%
  kable_styling("striped", full_width = F)
```

First 5 minutes for two participants activity data

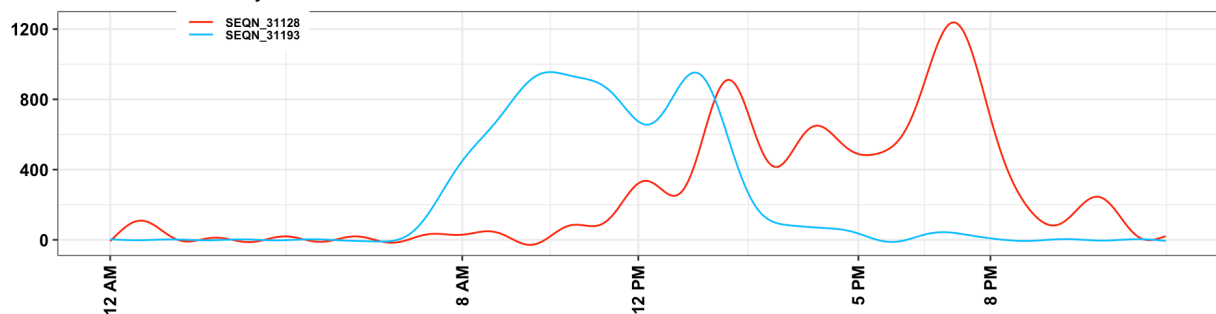
SEQN	PAXCAL	PAXSTAT	WEEKDAY	SDDSRVYR	MIN1	MIN2	MIN3	MIN4	MIN5
31128	1	1	1	4	166	27	0	276	0
31128	1	1	2	4	0	0	0	0	0
31128	1	1	3	4	0	0	0	0	0
31128	1	1	4	4	0	0	0	0	0
31128	1	1	5	4	0	0	0	0	0
31128	1	1	6	4	0	0	0	0	0
31128	1	1	7	4	0	0	0	0	0
31193	2	1	2	4	0	0	0	0	0
31193	2	1	3	4	335	2598	2185	1703	1378
31193	2	1	4	4	463	375	58	119	167
31193	2	1	5	4	0	0	0	0	0
31193	2	1	6	4	0	0	0	0	0
31193	2	1	7	4	0	0	0	0	0
31193	2	1	1	4	0	0	0	0	0

```
#smooth activity data
x <- 1:1440
#has to be converted to a numeric matrix type before smoothing is applied
tmp <- as.matrix(df.act[,grep("MIN", names(df.act))])
for(i in 1:nrow(df.act)){
  tmp[i,] <- gam(tmp[i,]~s(x,k=30))$fitted.values
}
```

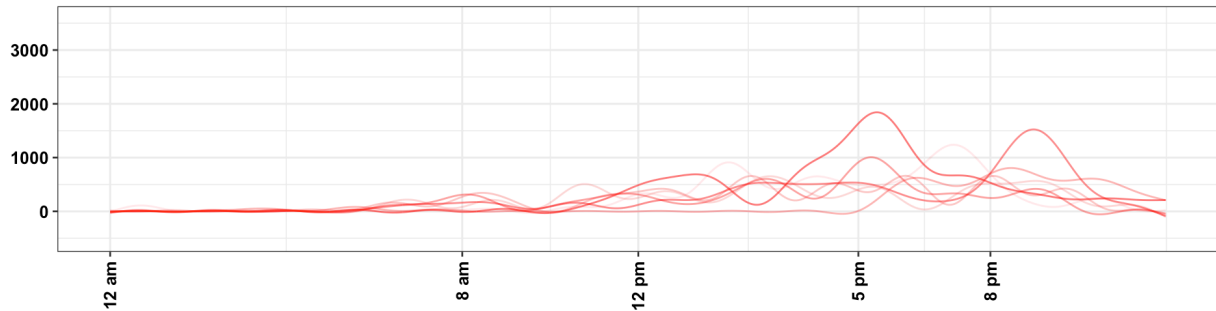
Raw data: Day 1



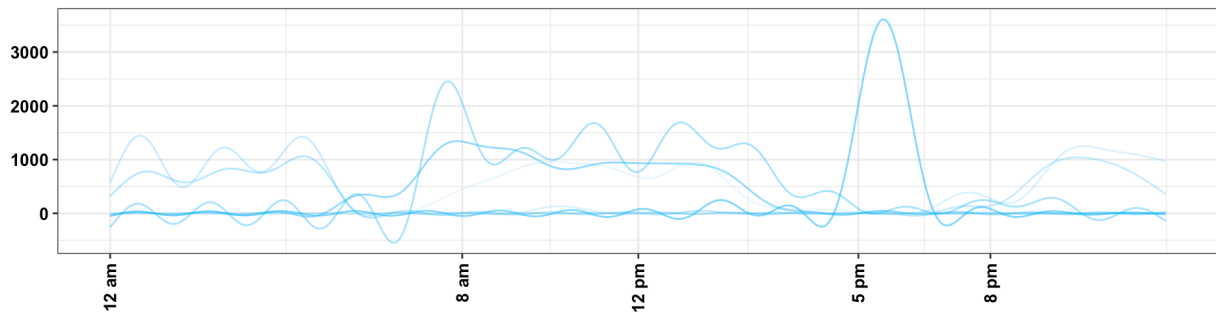
Smoothed data: Day 1



Smoothed daily activity profiles for subject 31128



Smoothed daily activity profiles for subject 31193



3. Create new factor covariates from NHANES questionnaire, which will be used in the prediction model. In addition, calculate average systolic blood pressure using the 4 measurements per participant.

```
## Code year 5 mortality, NAs for individuals with follow up less than 5 years and alive
AllAct$yr5_mort <- AllFlags$yr5_mort <-
  as.integer(ifelse(AllAct$permth_exm/12 <= 5 & AllAct$mortstat == 1, 1,
    ifelse(AllAct$permth_exm/12 < 5 & AllAct$mortstat == 0, NA, 0))
  )
## Create Age in years using the age at examination (i.e. when participants wore the device)
AllAct$Age <- AllFlags$Age <- AllAct$RIDAGEEX/12

## Re-level comorbidities to assign refused/don't know as not having the condition
## Note that in practice this does not affect many individuals, but it is an assumption we're making.
levels(AllAct$CHD)<- levels(AllFlags$CHD)<- list("No" = c("No","Refused","Don't know"), "Yes" = c("Yes"))
levels(AllAct$CHF)<- levels(AllFlags$CHF)<- list("No" = c("No","Refused","Don't know"), "Yes" = c("Yes"))
levels(AllAct$Stroke)<- levels(AllFlags$Stroke)<- list("No" = c("No","Refused","Don't know"), "Yes" = c("Yes"))
levels(AllAct$Cancer) <- levels(AllFlags$Cancer) <- list("No" = c("No","Refused","Don't know"), "Yes" = c("Yes"))
levels(AllAct$Diabetes) <- levels(AllFlags$Diabetes) <- list("No" = c("No","Borderline", "Refused","Don't know"), "Yes" = c("Yes"))

## Re-level education to have 3 levels and categorize don't know/refused to be missing
levels(AllAct$EducationAdult) <- levels(AllFlags$EducationAdult) <- list(
  "Less than high school" = c("Less than 9th grade", "9-11th grade"),
  "High school" = c("High school grad/GED or equivalent"),
  "More than high school" = c("Some College or AA degree", "College graduate or above"))

## Re-level alcohol consumption to include a level for "missing"
levels(AllAct$DrinkStatus) <- levels(AllFlags$DrinkStatus) <- c(levels(AllAct$DrinkStatus),
  "Missing alcohol")
AllAct$DrinkStatus[is.na(AllAct$DrinkStatus)] <-
  AllFlags$DrinkStatus[is.na(AllAct$DrinkStatus)] <- "Missing alcohol"

# systolic blood pressure calculation
AllAct$SYS <- AllFlags$SYS <- round(apply(AllAct[,c("BPXSY1", "BPXSY2", "BPXSY3", "BPXSY4")],
  1,mean, na.rm= T))

## Re-order columns so that activity and wear/non-wear flags are the last 1440 columns of our two
## data matrices. This is a personal preference and is absolutely not necessary.
act_cols <- which(colnames(AllAct) %in% paste0("MIN",1:1440))
oth_cols <- which(!colnames(AllAct) %in% paste0("MIN",1:1440))
AllAct <- AllAct[,c(oth_cols,act_cols)]
AllFlags <- AllFlags[,c(oth_cols,act_cols)]
rm(list=c("act_cols","oth_cols"))
```

4. Calculate daily activity summary measures:

- a. total activity count (TAC);
- b. total log activity count (TLAC);

- c. total accelerometer wear time (WT);
- d. total sedentary (< 100 counts) time (ST);
- e. total minutes of moderate/vigorous (<=2020 counts) physical activity (MVPA);
- f. sedentary/sleep/non-wear to active transition probability (SATP_{sl/nw});
- g. active to sedentary/sleep/non-wear transition probability (ASTP_{sl/nw}).

In addition, compute total log activity count summary measures (TLAC_1, TLAC_2, ..., TLAC_12) in each 2-hr window, i.e. 12AM-2AM, 2AM-4AM, 4AM-6AM, etc.

Note, there is one individual with 501 minutes recorded as NA. These missing data occur on the last day they wore the device for the last 501 minutes of the day. We impute these missing data with 0.

Description of these activity related measures is available at: <https://www.biorxiv.org/content/10.1101/182337v1> (<https://www.biorxiv.org/content/10.1101/182337v1>)

```
## Assign just the activity and wear/non-wear flag data to matrices.
## This makes computing the features faster but is technically required.
act_mat <- as.matrix(AllAct[,paste0("MIN",1:1440)])
flag_mat <- as.matrix(AllFlags[,paste0("MIN",1:1440)])

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

#total activity count (TAC)
AllAct$TAC <- AllFlags$TAC <- rowSums(act_mat)
#total log activity count (TLAC)
AllAct$TLAC <- AllFlags$TLAC <- rowSums(log(1+act_mat))
#total accelerometer wear time (WT)
AllAct$WT <- AllFlags$WT <- rowSums(flag_mat)
#total sedentary time (ST)
AllAct$ST <- AllFlags$ST <- rowSums(act_mat < 100)
#total time spent in moderate to vigorous physical activity (MVPA)
AllAct$MVPA <- AllFlags$MVPA <- rowSums(act_mat >= 2020)

## calculate fragmentation measures
bout_mat <- apply(act_mat >= 100, 1, function(x){
  mat <- rle(x)
  sed <- mat$lengths[which(mat$values == FALSE)]
  act <- mat$length[mat$values == TRUE]

  sed <- ifelse(length(sed) == 0, NA, mean(sed))
  act <- ifelse(length(act) == 0, NA, mean(act))
  c(sed,act)
})

AllAct$SBout <- AllFlags$SBout <- bout_mat[1,]
AllAct$ABout <- AllFlags$ABout <- bout_mat[2,]
AllAct$SATP <- AllFlags$SATP <- 1/AllAct$SBout
AllAct$ASTP <- AllFlags$ASTP <- 1/AllAct$ABout

# compute total log activity count in each 2-hr window,
# 2 hour (120 minutes) binning window
tlen <- 120
nt <- floor(1440/tlen)
# create a list of indices for binning into 2-hour windows
inx_col_ls <- split(1:1440, rep(1:nt,each=tlen))
Act_2hr <- sapply(inx_col_ls, function(x) rowSums(log(1+act_mat[,x,drop=FALSE])))
colnames(Act_2hr) <- paste0("TLAC_",c(1:12))

AllAct <- cbind(AllAct, Act_2hr)
AllFlags <- cbind(AllFlags, Act_2hr)

rm(list=c("tlen","nt","inx_col_ls","Act_2hr","act_mat","flag_mat","bout_mat"))
```

5. From the total of

14631 exclude participants who were:

- i. younger than 50 years old, or 85 and older at the time they wore the accelerometer (10859 participants);
- ii. missing BMI or education predictor variables (41 participants);
- iii. had fewer than 3 days of data with at least 10 hours of estimated wear time (517 participants);
- iv. missing mortality information (21 participants);
- v. alive with follow up less than 5 years (0 participants);
- vi. missing systolic blood pressure, total cholesterol or HDL cholesterol measurements (293 participants).

The number of participants with missing data for all pairwise combinations of variables with missing data:

Missing data patterns

	BMI	Education	Bad Accel Data	Mortality	Follow-up	Lab
BMI	35	0	8	1	0	7
Education	0	6	2	0	0	1
Bad Accel Data	8	2	517	4	0	57
Mortality	1	0	4	21	0	1
Follow-up	0	0	0	0	0	0
Lab	7	1	57	1	0	293

6. Create our dataset for analysis, "data_analysis", with one row per subject containing only those subjects who meet our inclusion criteria.

```
## Create our dataset for analysis with one row per subject
## containing only those subjects who meet our inclusion criteria.
data_analysis <- subset(table_dat, Exclude == 0)
## get adjusted survey weights using the reweight_accel function
data_analysis <- reweight_accel(data_analysis)

## Get activity/flag data for only those included participants AND days.
## Since we've already removed the "bad" days from Act_Analysis and Act_Flags,
## we need only subset based on subject ID now
Act_Analysis <- subset(Act_Analysis, SEQN %in% data_analysis$SEQN)
Flags_Analysis <- subset(Flags_Analysis, SEQN %in% data_analysis$SEQN)
```

Thus, after exclusion criteria were applied, the data set contained 2978 subjects. Among these participants, 86 had missing alcohol information; for these participants we introduced the category "Missing Alcohol" and retained them in the dataset.

Maximum follow-up time in this cohort

```
max(data_analysis$permth_exm/12)
```

```
## [1] 13.08333
```

7. Calculate subject specific averages of the accelerometry features using only the "good" days of data.

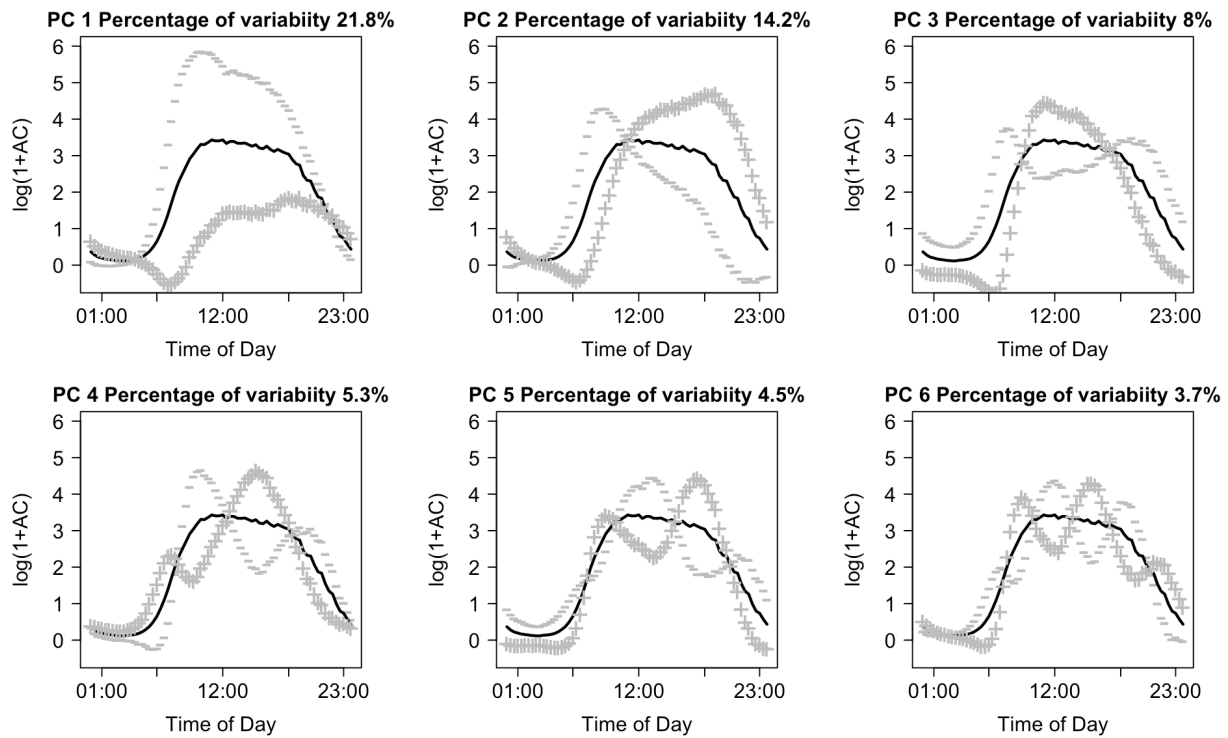
```
## calculate subject specific averages of the accelerometry features
## using only the "good" days of data
act_var_nms <- c("TAC", "TLAC", "WT", "ST", "MVPA", "ABout", "SBout", "SATP", "ASTP", paste0("TLAC_", 1:12))
for(i in act_var_nms){
  data_analysis[[i]] <- vapply(data_analysis$SEQN, function(x) mean(Act_Analysis[[i]][Act_Analysis$SEQN==x]), numeric(1))
}

## clean up the workspace
rm(list=c("AllAct", "AllFlags", "i", "criteria_vec", "nms_rm", "tab_miss", "act_var_nms"))
```

8. Get log(count + 1) activity data in a matrix format and conduct functional Principal Component analysis (fPCA) using the `fpca.face()` function with 50 knots in the *refund* package in R.

```
Act <- as.matrix(log(1+Act_Analysis[,paste0("MIN", 1:1440)]))
Act[is.na(Act)] <- 0
fpca_fit <- fpca.face(Act, knots=50)
```

9. Retain the first 6 PCs, which explain approximately 57% variability in the daily activity data (the space spanned by the rows of the matrix of non-smoothed log transformed activity counts) and obtain the score for each day on each PC and calculate the mean and standard deviation of these scores for each subject across days.



10. Identify the PC-specific mean and standard deviation that are significantly associated with 5-year mortality using backward selection logistic model fitted with `svyglm()` function in R. In this step, the demographic, behavioral and comorbidity variables were included in all models and the selection was conducted on the means and standard deviations of the scores on the 6 PCs (a total of 12 variables) using the complex survey AIC criteria (Lumley and Scott 2015).

Note, there are 86 participants with missing alcohol information, which was coded as an additional level of the `DrinkStatus` variable. Results are presented in Table

```
## Create a svydesign() object for
## estimating complex survey generalized linear models.
## Here we use the adjusted (re-weighted) 4-year normalized survey weights.
data_analysis_svy <- svydesign(id= ~SDMVPSU, strata = ~SDMVSTRA,
                             weights = ~wtmec4yr_adj_norm, data = data_analysis, nest = TRUE)

ind_vars <- c("mi1", "mi2", "mi3", "mi4", "mi5", "mi6",
             "si1", "si2", "si3", "si4", "si5", "si6")
inc_vars <- c("Age", "SmokeCigs", "DrinkStatus", "BMI_cat",
             "Diabetes", "CHF", "CHD", "Stroke",
             "Cancer", "MobilityProblem", "LBXTC", "LBDHDD", "SYS")
exc_vars <- ind_vars
epic_vec <- var_vec <- model_vec <- rep(NA, length(ind_vars))
for(i in 1:length(ind_vars)){
  epic_ij <- rep(NA, length(exc_vars))

  for(k in 1:length(exc_vars)){
    form <- paste0(c(inc_vars, exc_vars[-k]), collapse="+")
    fit_tmp <- svyglm(as.formula(paste("yr5_mort ~", form)), design=data_analysis_svy, family=quasibinomial())

    epic_ij[k] <- survey::extractAIC.svyglm(fit_tmp, k=4)[2]

    rm(list=c("fit_tmp", "form"))
  }

  k_cur <- which(epic_ij == min(epic_ij))
  model_vec[i] <- paste0(c(inc_vars, exc_vars[-k_cur]), collapse="+")
  exc_vars <- exc_vars[-k_cur]
  epic_vec[i] <- epic_ij[k_cur]
  rm(list=c("k_cur", "k", "epic_ij"))
}

## get the final model as the first model where AIC increases after removing a variable
## identified mi1, si6
backward_model <- model_vec[which(diff(epic_vec) > 0) + 1][1]
fit_logistic_pca <- svyglm(as.formula(paste("yr5_mort ~", backward_model)), design=data_analysis_svy,
                          family=quasibinomial())
```

```
ci.pca <- confint(fit_logistic_pca, level = 0.95, method = "likelihood")

#exponentiate to convert estimated coefficients into odds ratio
df2 <- data.frame(exp(df[,which(colnames(df) == "Estimate")]), exp(ci.pca))
colnames(df2) <- paste("OR", c("Estimate", "Lower CL", "Upper CL"))
df <- cbind(df, round(df2,3))

kable(df, caption = "Final backward selection model estimated coefficients")%>%
  kable_styling("striped", full_width = F) #p-values
```

Final backward selection model estimated coefficients

	Estimate	Pr(> t)	OR Estimate	OR Lower CL	OR Upper CL
Intercept	-7.590	0.000	0.001	0.000	0.004
Age	0.080	0.000	1.083	1.059	1.108
Former Smoker	0.487	0.021	1.627	1.073	2.485
Current Smoker	0.941	0.000	2.563	1.797	3.648
Non-Drinker	0.567	0.000	1.763	1.279	2.440
Heavy Drinker	1.053	0.002	2.866	1.423	5.576
Missing Alcohol	0.816	0.065	2.261	0.875	5.197
Underweight	0.597	0.384	1.817	0.400	6.520
Overweight	-0.554	0.005	0.575	0.386	0.854
Obese	-0.509	0.000	0.601	0.478	0.755
Diabetes: yes	0.286	0.161	1.331	0.879	1.989
CHF: yes	0.833	0.001	2.300	1.356	3.841
CHD: yes	0.113	0.671	1.120	0.649	1.879
Stroke: yes	0.287	0.299	1.332	0.756	2.279
Cancer: yes	0.273	0.100	1.314	0.939	1.823
Mobility Problem	0.620	0.001	1.859	1.270	2.719
Total Cholesterol	-0.003	0.243	0.997	0.992	1.002
HDL Cholesterol	-0.001	0.840	0.999	0.989	1.008
Systolic Blood Pressure	0.000	0.949	1.000	0.993	1.008
mi1	0.014	0.008	1.014	1.004	1.026
si6	-0.077	0.000	0.926	0.888	0.965

11. Replace the significant PC based measures with corresponding surrogate variables. This step is done to improve the interpretability and comparability (across studies) of results. For those unfamiliar with principal component analysis, reading the section "Intuition behind fPCA" below may be helpful in understanding this step further.

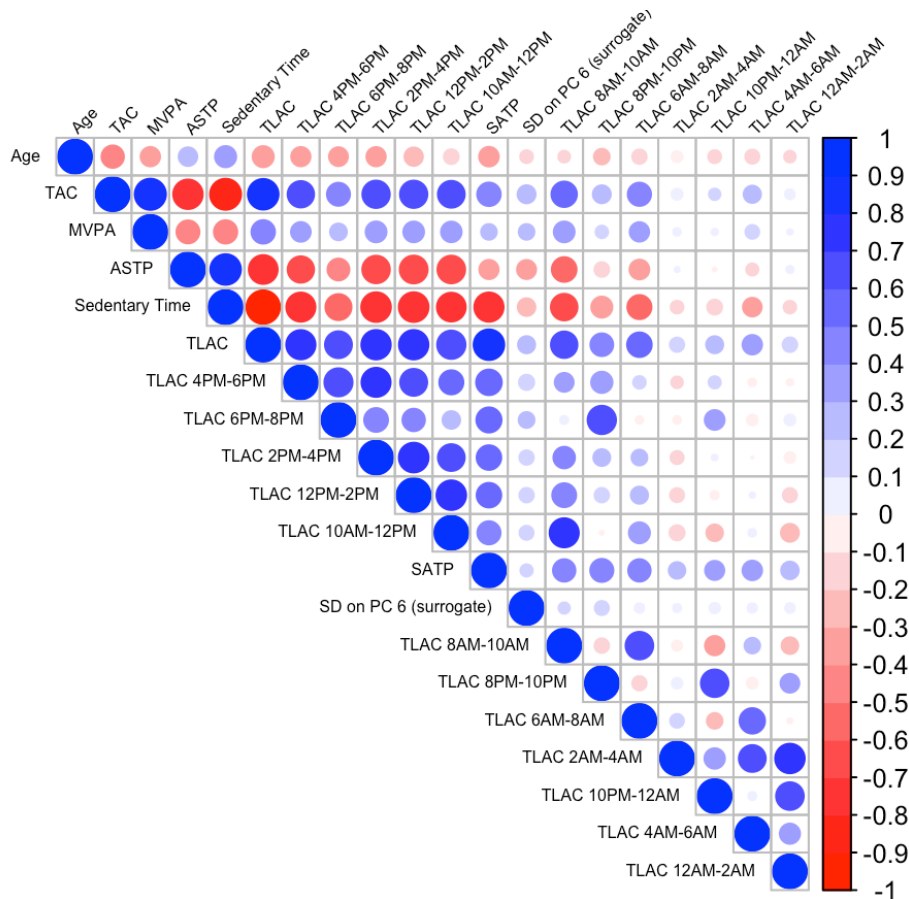
Identifying potential surrogate measures is based on (subjective) interpretations involving the shapes of each principal component. Fundamentally, the idea is to use visual inspection of the principal components to identify the "dominant" features of each component. Then, we return to the original data, and calculate a statistic which we believe captures this dominant feature. For example, looking at the upper-left panel of the plot above in Step 10, we see that days which load negatively on the first the first principal component tend to be extremely active, while those who load positively tend to be very inactive. As a result, one reasonable "guess" at a surrogate measure which is highly associated with average first component is simply the total log transformed activity count (TLAC) for that day. If that is true, we would also expect that the average PC1 score within subjects is highly correlated with their average TLAC across days. In our data, average TLAC and average PC1 score are highly (negatively) correlated ($\hat{\rho} = -0.87$), which is expected based on the sign of the first PC.

This procedure would then be repeated for each feature identified as potentially predictive. In our application, we are also interested in the standard deviation of PC6. Looking at the bottom-right panel of the plot above in Step 10, we see that there are 6 periods where the contrast is highest between days with positive and negative loadings (i.e. the difference between the + and - curves are largest). One reasonable guess for a statistic which is highly correlated with PC6 score is the difference in average activity during the specific time periods where positive/negative loadings are high/low, respectively. For example, days that load highly on PC6 should, on average, have higher activity during the mid morning (8AM-10AM), late afternoon (3PM-5PM) and late evening (10PM-12AM) and lower activity during the early morning (5AM-7AM), late morning/early afternoon (11AM-1PM), and early evening (6PM-8PM). Since we're interested in the standard deviation of PC6 score, we calculate the standard deviation of average log-transformed activity counts during these periods as a surrogate measure for s_{i6} . In our analysis, we used all 6 of these time periods and obtained an observed correlation of $\hat{\rho} = 0.87$, though multiple choices can be explored to see which statistic has the highest correlation with s_{i6} .

Correlation between activity derived predictors and age

Our analysis includes 20 variables derived directly from accelerometry measurements. These variables may be highly correlated with one another and with age. The correlation plot between age and all activity derived variables is presented below. Age has high negative correlations with TAC, TLAC and positive correlation with sedentary time. TAC is highly correlated with most activity derived measures, including MVPA, SATP, Sedentary time, TLAC, and TLAC 4PM – 6PM, 6PM – 8PM, 2PM – 4PM, 12PM – 2PM, 10PM – 12 PM, 8AM – 10AM, 6AM – 8AM. ASTP, sedentary time, TLAC and SATP are among top activity derived measures that are highly correlated with multiple other variables. The surrogate for the standard deviation on the 6th PC has low correlation with other activity derived variables, which explains its contribution to the increase in AUC in the forward selection model beyond the conventional activity measures.

Due to high correlation among different activity summaries, many of these variables contain similar information and often replace each other in the final mortality prediction model. Thus, in practice, we suggest examining a smaller subset of activity derived variables including: TLAC, sedentary and wear time, MVPA, SATP, ASTP, and the surrogates the standard deviation of the sixth PC (SD on PC 6).



Fitting 5-year complex survey weighted mortality prediction model

The first step is to construct a survey object and specify corresponding survey weights.

```
data_analysis_svy_adj <- svydesign(id= ~SDMVPSU,
                                strata = ~SDMVSTRA,
                                weights = ~wtmec4yr_adj_norm,
                                data = data_analysis, nest = TRUE)
data_analysis_svy_adj
```

```
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (60) clusters.
## svydesign(id = ~SDMVPSU, strata = ~SDMVSTRA, weights = ~wtmec4yr_adj_norm,
##          data = data_analysis, nest = TRUE)
```

```
length(unique(data_analysis$SDMVPSU))
```

```
## [1] 2
```

```
length(unique(data_analysis$SDMVSTRA))
```

```
## [1] 30
```

The survey weighted logistic regression model can be fitted using `svyglm()` function in package `survey`.

```

final_formula <- "TAC+Age+SmokeCigs+CHF+
  DrinkStatus+ASTP+MobilityProblem+Gender+
  sPC6+Diabetes+EducationAdult+TLAC_1+Stroke"
fit_final <- svyglm(as.formula(paste("yr5_mort ~",
  final_formula)),
  design=data_analysis_svy_adj,
  family=quasibinomial())

```

Syntax for extracting model coefficients from the survey object is similar to that of `lm()` and `glm()` functions in R.

```

df <- round(summary(fit_final)$coefficients, 3)
kable(df)%>%
  kable_styling("striped", full_width = F, font_size = 8)

```

Table presents estimated final model coefficients with corresponding standard errors and significance values, according to the complex survey design of NHANES via the `svyglm()` function. The number of variables used in the model is selected according to across validated AUC criteria. The "adjusted" weights we use for regression analyses are "wtmec4yr_adj_norm". These weights are calculated using the `reweight_accel()` function (see `?reweight_accel` for help) which re-weights observed participants using age, gender, and ethnicity strata.

Finally, we construct confidence intervals for estimated regression coefficients and exponentiate these values to convert to OR confidence intervals.

```

ci.final <- confint(fit_final, level = 0.95, method = "likelihood")

#exponentiate to convert estimated coefficients into odds ratio
#exponentiate to convert estimated coefficients into odds ratio
df2 <- data.frame(exp(df[,which(colnames(df) == "Estimate")]), exp(ci.final))
colnames(df2) <- paste("OR", c("Estimate", "Lower CL", "Upper CL"))
df <- cbind(df, round(df2,3))

## selected using the adjusted survey weight model
kable(df,caption = "Estimated final model coefficients with corresponding standard errors and significance values") %>%
  kable_styling("striped", full_width = F)

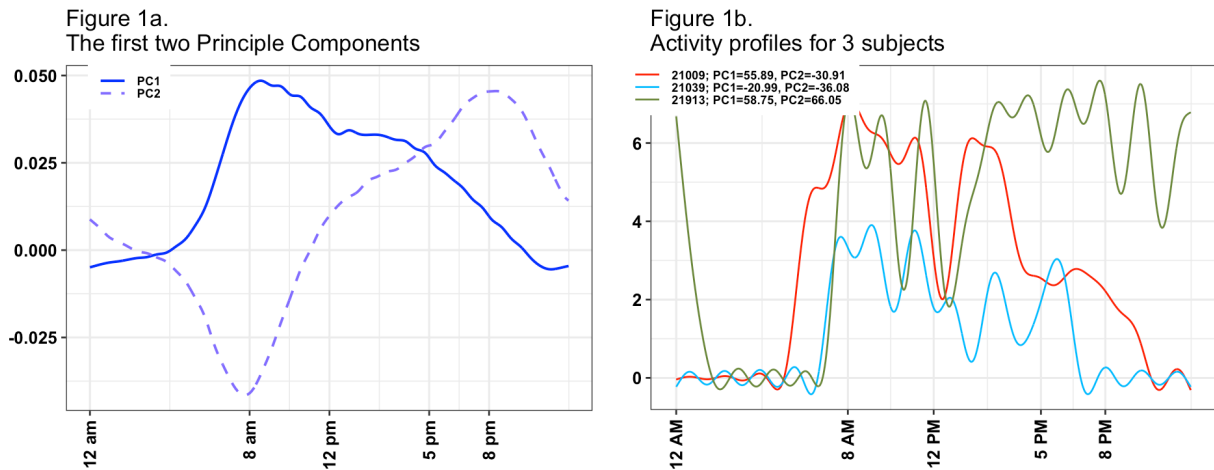
```

Estimated final model coefficients with corresponding standard errors and significance values

	Estimate	Pr(> t)	OR Estimate	OR Lower CL	OR Upper CL
Intercept	-9.511	0.000	0.000	0.000	0.001
TAC	0.000	0.982	1.000	1.000	1.000
Age	0.083	0.000	1.087	1.063	1.112
Former Smoker	0.332	0.176	1.394	0.835	2.345
Current Smoker	0.797	0.002	2.219	1.412	3.478
CHF: yes	0.777	0.013	2.175	1.177	3.930
Non-Drinker	0.565	0.010	1.759	1.165	2.677
Heavy Drinker	0.963	0.018	2.620	1.148	5.673
Missing Alcohol	0.747	0.106	2.111	0.752	5.193
ASTP	4.305	0.016	74.069	2.328	2380.239
Mobility Problem	0.546	0.028	1.726	1.057	2.816
Gender: female	-0.648	0.007	0.523	0.332	0.817
SD on PC 6 (surrogate)	-1.077	0.002	0.341	0.180	0.635
Diabetes: yes	0.216	0.310	1.241	0.780	1.937
High school Education	-0.008	0.973	0.992	0.582	1.694
More than high school Education	-0.231	0.309	0.794	0.489	1.294
TLAC 12AM-2AM	0.002	0.099	1.002	0.999	1.005
StrokeYes	0.193	0.505	1.213	0.636	2.227

Intuition behind fPCA

A major problem with PC analysis is that it is not always intuitive and requires a degree of familiarity with matrix algebra and complex trajectories (functional data analysis in statistics speak). While some of these problems are unavoidable given the complexity of the data, we will now provide the needed intuition for understanding both the PCs and the implication of our findings on the original data scale (daily minute-level activity profiles). We will start with explaining the first 2PCs, which are shown in the left-top panel of Figure 1.



The first PC (solid blue line) captures 22% of the overall variability in the observed daily profiles of activity. It has a distinct shape, with values starting negative between 12AM and 5AM then becoming strongly positive with a peak around 8AM and slowly decreasing but staying positive until 9:30PM, and then becoming negative after 9:30PM. This is exactly what we expected to see. For each subject, we have 3 - 7 days of valid activity data. Individuals with a positive score on this component (a.k.a., positively loaded on the first PC) on a given day will tend to have less activity during the night hours and more activity during the day hours than the average activity across all subject-days. The biggest difference between such a subject's day and the average daily activity across all subjects is centered on the morning hours (8AM-9AM). In contrast, the second PC (dashed line) captures 14% of the overall variability in the observed daily activity profiles. It starts positive between 12AM and 2AM, then becomes negative between 2AM and 11AM, with a negative peak at 8AM, increases between 11 AM and 8PM, and decreases while staying positive after 8PM. Participants days with positive scores on this component will be more active in the evening than the average individuals daily activity.

Now, let us investigate in detail the connection between PCs, scores and individual daily trajectories. The right panel in Figure 1 displays the one day of activity profiles for 3 subjects. Here, we plot the activity data smoothed for each subject and day using thin-plate penalized spline with 30 knots as implemented in the **gam()** function in the **mgcv** package in R. The individual days for subjects 21009 (red line) and 21913 (green line) have overall high activity, which is reflected by the high positive loadings on PC 1 (55.89 and 58.75, respectively). In contrast, the individual day's activity for subject 21039 (blue line) has lower levels, which is reflected by the negative score on the first PC (-21.12). Subjects 21009 (red line) and 21039 (blue line) are mostly active between 7AM and 9PM and their corresponding scores of the second principal component are negative (-30.91 and -36.08, respectively). Subject 21913 (green line) is however, unusually highly active during night hours (8PM - 2AM) with a highly positive score on the second PC (66.05).

The last, but not least important interpretation is of means of scores versus standard deviation of scores. In Figure 1, we have displayed three days, one for each subject. However, each subject has multiple days and each day will get a score and a pattern. For example, on days when subject 21009 (red line) is less active the score on PC1 will be lower, even if the general pattern stays the same. Thus, for every subject and PC we obtain a vector of scores; for 21009 (red line) we obtain (38.39, 55.89, 35.55, 60.61, 40.91, 48.67, -21.25) on the first PC, where we showed only the trajectory corresponding to the 3rd day. What we calculate is the mean of these scores, 36.97, and the standard deviation, 27.27. The mean of the scores is relatively easy to interpret, as it represents whether the average of the 7 days is higher or lower on PC1. The standard deviation captures the day-to-day variability of the individual. In this case the mean and standard deviation for subject 21009 (red line) were 36.97 and 27.27, respectively. In contrast, for subject 21039 (blue line), the mean and standard deviation were -19.16 and 5.02, respectively, much smaller than for subject 21009 (red line). This means that both the overall mean and the daily variability around this larger mean are larger for subject 21009 than for subject 21039 (blue line). This is depicted in Figure 2, where we show smoothed activity data for all days for subjects 21009 (Figure 2a red lines) and 21039 (Figure 2b blue lines). Note that the red lines tend to be higher the blue lines and that the blue lines are less variable around their means. We conclude that, in general, average PC1 scores will tend to distinguish between lower and higher activity individuals whose are, on average, more active over the course of days with available activity data. In contrast, average PC2 scores will distinguish between individuals who, on average, have high activity in the morning and low in the evening/night and individuals who, on average, have lower activity intensity in the morning and higher during the evening/nighttime.

Of course, things are more complicated once we start interpreting every component. Instead, in Table 1 we will provide just the interpretation of those components and summaries that were found to be predictive of the outcome.

Figure 2a.
Daily activity profiles for subject 21009

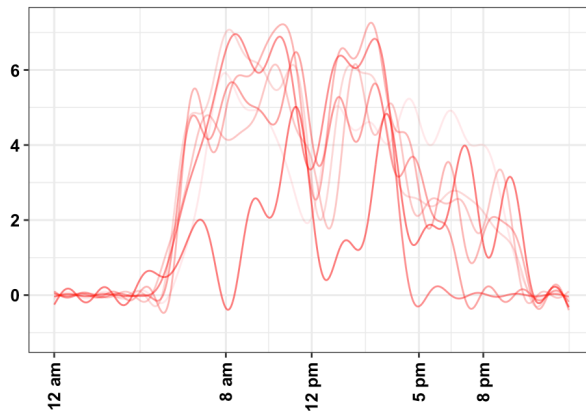


Figure 2b.
Daily activity profiles for subject 21039

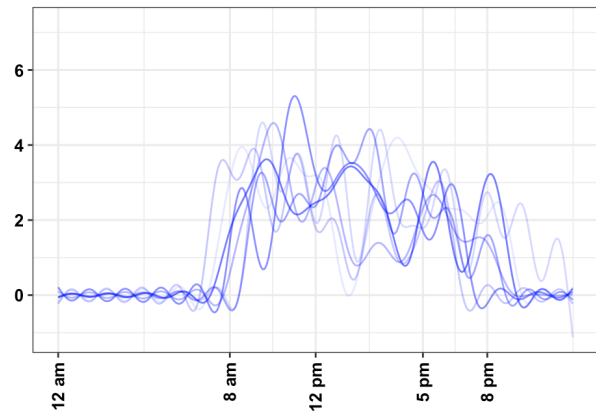


Table 3: Interpretation of the results of FPCA

Result	Interpretation	Surrogates
(-) association mi1	Individuals with higher levels of overall activity during the day, and those who have higher early afternoon activity relative to early AM are associated with later mortality	1. Average TLAC
(+) association si6	Individuals who are more variable in the start time of their daily activity are associated with earlier mortality.	1. Standard deviation of ratio of mid-day to morning/afternoon activity 2. Standard deviation of the difference in average activity during peaks/troughs highlighted by PC6

References

1. T Lumley, and A Scott; AIC and BIC for modeling with complex survey data, *Journal of Survey Statistics and Methodology*, Volume 3, Issue 1, 1 March 2015, Pages 1-18, <https://doi.org/10.1093/jssam/smu021> (<https://doi.org/10.1093/jssam/smu021>)
2. A Leroux, E Smirnova, Q Cao, and C Crainiceanu. rnhanesdata: NHANES Accelerometry Data Pipeline. URL:<https://github.com/andrew-leroux/rnhanesdata> (<https://github.com/andrew-leroux/rnhanesdata>). R package version 1.0.
3. E Smirnova, A Leroux, Q Cao, L Tabacu, V Zipunnikov, C Crainiceanu, and J Urbanek. The Predictive Performance of Objective Measures of Physical Activity Derived From Accelerometry Data for 5-Year All-Cause Mortality in Older Adults: National Health and and Nutritional Examination Survey 2003–2006, *The Journals of Gerontology: Series A*, glz193, <https://doi.org/10.1093/gerona/glz193> (<https://doi.org/10.1093/gerona/glz193>).
4. A Leroux, J Di, E Smirnova, E McGuffey, Q Cao, E Bayatmokhtari, L Tabacu, V Zipunnikov, J Urbanek, C Crainiceanu. Organizing and Analyzing the Activity Data in NHANES, *Statistics in Biosciences*, 1-26, 2019.