Accelerometry data processing script

Deborah Telford

2025-11-18

Overview

This script imports handwritten diary data and activPAL accelerometry files (15-second epochs), processes them, and produces a participant-level summary dataset:
`participant_complete.csv`

The processing steps include:

- Importing and cleaning diary files
- Importing and structuring raw activPAL files
- Intensity estimation (MVPA, VPA)
- Hourly and daily summaries
- Identification of valid measurement days
- Aggregation by participant and day type

---

1. Load packages, import diary data and define parameters

1.1. Load Packages and Import Diary Data

```r
library(tidyverse)
library(data.table) # for rbindlist function

# Import diary data including screen time, sleep latency, etc.
diary <- read_csv("Data/diary_fo.csv")
```

1.2 Define Threshold Parameters

```r
mpa_threshold = 25       # steps per 15s epoch (equivalent to 100/min)
vpa_threshold = 31       # steps per 15s epoch (equivalent to 125/min)
nonwear_threshold = 180  # minutes per day of non-wear time permitted
day_threshold = 3        # minimum valid days per participant
```

---

2. Import ActivPAL Data (15s epoch files, 1 file per participant)

```
# 1) Create list of files to import
myfiles <- list.files('Data/PAL/', full.names=TRUE, recursive = TRUE)

# 2) Import files into dataframe
epoch15raw <- rbindlist(lapply(myfiles, fread), idcol = "participantID")

# 3) Add participantID field using first 4 characters of filename
epoch15raw[, participantID := factor(participantID,
                                     labels = substr(basename(myfiles), start=1, stop=4))]
```

---

3. Process ActivPAL Data and Summarise by Hour

3.1 Add Date Fields and Adjust for Rounding Errors

```
epoch15 <- epoch15raw %>%
  mutate(time = as.numeric(time)) %>%
  mutate(date = as.POSIXct((time+0.000001)*3600*24, tz="GMT", origin="1899-12-30"),
         datenum = floor(time),
         hour = hour(date),
         minute = minute(date),
         roundingerror_s = (`Nonwear Time (s)` + `Upright Time (s)` +
                            `Sedentary Time (s)` + `Primary Lying Time (s)` +
                            `Secondary Lying Time (s)` - 15))
```

3.2 Intensity Estimates

```
epoch15 <- mutate(epoch15,
                  MVPAstep = if_else(StepCount >= mpa_threshold, `Stepping Time (s)`, as.integer(0)),
                  VPAstep = if_else(StepCount > vpa_threshold, `Stepping Time (s)`, as.integer(0)))
```

3.3 Calculate Sum of Vector Magnitudes (SVM) — for Reference

```
epoch15 <- epoch15 %>%
  rename(x_count="Sum(abs(dChannel1))",
         y_count="Sum(abs(dChannel2))",
         z_count="Sum(abs(dChannel3))") %>%
  mutate(svm = sqrt(x_count^2 + y_count^2 + z_count^2))
```

3.4 Create Hourly Summary

```
hour <- epoch15 %>% group_by(participantID, datenum, hour) %>%
  summarise(participantID = first(participantID),
            date = first(date),
            steps = sum(StepCount),
            nonwear = round(sum(`Nonwear Time (s)`)/60, 3),
            upright = round(sum(`Upright Time (s)`)/60, 3),
            sitting = round(sum(`Sedentary Time (s)`)/60, 3),
            lying = round(sum(`Secondary Lying Time (s)`)/60, 3),
            cycling = round(sum(`Cycling Time (s)`)/60, 3),
            stepping = round(sum(`Stepping Time (s)`)/60, 3),
            moving = cycling + stepping,
            standing = upright - stepping - cycling,
            sleep = round(sum(`Primary Lying Time (s)`)/60, 3),
            SB = round(sum(`Sedentary Time (s)` + `Secondary Lying Time (s)`)/60, 3),
            MVPAstep = round(sum(MVPAstep)/60, 3),
            VPAstep = round(sum(VPAstep)/60, 3),
            roundingerror = round(sum(roundingerror_s)/60, 3),
            total_min = round(upright + sleep + SB + nonwear - roundingerror, 2))

write_csv(hour, "Output/hour.csv")
```

---

4. Daily Summary

```
# Import hour summary if returning later
hour <- read_csv("Output/hour.csv")

# Aggregate by participant and date
day <- hour %>% group_by(participantID, datenum) %>%
  summarise(participantID = first(participantID),
            datenum = first(datenum),
            date = first(date),
            steps = sum(steps),
            nonwear = sum(nonwear),
            sitting = sum(sitting),
            lying = sum(lying),
            cycling = sum(cycling),
            stepping = sum(stepping),
            moving = sum(moving),
            standing = sum(standing),
            sleep = sum(sleep),
            SB = sum(SB),
            MVPAstep = sum(MVPAstep),
            MVPA = MVPAstep + cycling,
            LPAmove = stepping - MVPAstep,
            VPAstep = sum(VPAstep),
            roundingerror = sum(roundingerror),
            total_min = sum(total_min))
```

---

5. Identify Valid Days and Merge with Diary Data

```r
# Remove incomplete days
day <- day %>% filter(total_min > 1439)

# Join with diary data and filter out invalid days
day <- full_join(day, diary, by = c("participantID","datenum")) %>%
  filter(!(is.na(daytype) & nonwear > nonwear_threshold)) %>%
  filter(!is.na(daytype))

# Add valid_day and sleep_net
day <- day %>% mutate(
  valid_day = if_else(nonwear >= nonwear_threshold |
                      nonweardiary >= nonwear_threshold/60 |
                      abs(roundingerror) > 1 |
                      daytype == "X" |
                      is.na(nonwear), 0, 1),
  sleep_net = (sleep - latency)/60,
  daytype2 = if_else((!is.na(nonweardiary) & nonweardiary >= nonwear_threshold/60) |
                     (!is.na(nonwear) & nonwear >= nonwear_threshold), "NW",
                     if_else(is.na(nonwear), "P",
                             if_else(roundingerror < -10, "E", daytype)))
)
```

---

6. Summarise Day Types and Participant-Level Metrics

```r
# Daytype summary
daytype_summary <- day %>% group_by(participantID) %>%
  summarise(X = sum(daytype2=="X", na.rm=TRUE),
            S = sum(daytype2=="S", na.rm=TRUE),
            W = sum(daytype2=="W", na.rm=TRUE),
            N = sum(daytype2=="N", na.rm=TRUE),
            D = sum(daytype2=="D", na.rm=TRUE),
            P = sum(daytype2=="P", na.rm=TRUE),
            E = sum(daytype2=="E", na.rm=TRUE),
            NW = sum(daytype2=="NW", na.rm=TRUE),
            V = sum(valid_day, na.rm=TRUE),
            gaming = round(mean(gaming),2),
            tv = round(mean(tv),2),
            phone = round(mean(phone),2),
            computer = round(mean(computer),2),
            homework = round(mean(homework),2),
            screen = round(mean(screen),2),
            bedtime_var = round((as.numeric(max(bedtime)) - as.numeric(min(bedtime)))*24,2),
            waketime_var = round((as.numeric(max(waketime)) - as.numeric(min(waketime)))*24,2),
            VPA_days = sum(VPA_diary=="yes"),
            BM_days = sum(BM_diary=="yes"))

write_csv(daytype_summary, "Output/daytype_fo.csv")
```

---

7. Create Final Participant Dataset

```r
participant <- day %>% filter(valid_day==1) %>%
  group_by(participantID) %>%
  summarise(steps = mean(steps),
            nonwear = round(mean(nonwear),2),
            sitting = round(mean(sitting),2),
            lying = round(mean(lying),2),
            cycling = round(mean(cycling),2),
            stepping = round(mean(stepping),2),
            moving = round(mean(moving),2),
            standing = round(mean(standing),2),
            sleep = round(mean(sleep),2),
            SB = round(mean(SB),2),
            MVPAstep = round(mean(MVPAstep),2),
            MVPA = round(mean(MVPA),2),
            LPAmove = round(mean(LPAmove),2),
            VPAstep = round(mean(VPAstep),2),
            latency = round(mean(latency),2),
            sleep_net = round(mean(sleep_net),2),
            valid_days = sum(valid_day),
            school_days = sum(daytype2=="S", na.rm=TRUE))

participant <- participant %>% mutate(validAP = (valid_days>=3 & !is.na(nonwear))*1)

participant_complete <- full_join(participant, daytype_summary, by="participantID") %>%
  filter(validAP==1) %>%
  select(-X,-S,-W,-N,-D,-P,-E,-V,-NW,-validAP)

write_csv(participant_complete, "Output/participant_complete_fo.csv")
```

---

8. Summarise by Day Type (School / Non-School)

```r
participant_daytype <- day %>%
  filter(valid_day==1) %>%
  mutate(daytype=if_else(daytype=="S","S","N")) %>%
  group_by(participantID, daytype) %>%
  summarise(steps=mean(steps),
            nonwear=mean(nonwear),
            sitting=mean(sitting),
            lying=mean(lying),
            cycling=mean(cycling),
            stepping=mean(stepping),
            moving=mean(moving),
            standing=mean(standing),
            sleep=mean(sleep),
            SB=mean(SB),
            MVPA=mean(MVPA),
            LPAmove=mean(LPAmove),
            VPAstep=mean(VPAstep),
            valid_days=sum(valid_day),
            gaming = mean(gaming),
            tv = mean(tv),
            phone = mean (phone),
            computer = mean(computer),
            homework = mean(homework),
            screen=mean(screen),
            sleep_net=mean(sleep_net))

daytype <- participant_daytype %>%
  group_by(daytype) %>%
  summarise(steps=mean(steps),
            nonwear=mean(nonwear),
            sitting=mean(sitting),
            lying=mean(lying),
            cycling=mean(cycling),
            stepping=mean(stepping),
            moving=mean(moving),
            standing=mean(standing),
            sleep=mean(sleep),
            SB=mean(SB),
            MVPA=mean(MVPA),
            LPAmove=mean(LPAmove),
            VPAstep=mean(VPAstep),
            valid_days=sum(valid_days),
            gaming = mean(gaming),
            tv = mean(tv),
            phone = mean (phone),
            computer = mean(computer),
            homework = mean(homework),
            screen=mean(screen),
            sleep_net=mean(sleep_net))
```

End of Script