HRV analysis

Janice Tjeng

2017-10-31

# Load packages

library(RMySQL)  
library(tidyverse)  
library(lubridate)  
library(stringr)

# Read data from E4, firstbeat, & MSband

db <- dbConnect(MySQL(), user="deepresearcher", password="ctpc177!", dbname="deephealth2", host="deephealthlab.org")  
e4 <- dbSendQuery(db,"SELECT \* FROM viewl\_e4\_rr")  
e4dat <- fetch(e4,n=-1)   
fb <- dbSendQuery(db, "SELECT \* FROM viewl\_firstbeat\_rr")  
fbdat <- fetch(fb, n=-1)  
mb <- dbSendQuery(db, "SELECT \* FROM viewl\_msband\_rr")  
mbdat <- fetch(mb, n=-1)

# Read timing data

timing <- read\_csv("timing\_data.csv")  
df <- timing %>%  
 select (Event, LWP2\_0019, Activity) %>%  
 rename("Time"="LWP2\_0019")  
n <- nrow(df)  
dte <- mdy(as.character(df[1,"Time"]))  
start <- as.character(df[2,"Time"])  
end <- as.character(df[n,"Time"])

# Select rows for user 19 from E4

user19\_l <- e4dat %>%  
 filter(user=="lwp2\_0019", device\_location=="Left") %>%   
 mutate(readable\_timestamp=as.POSIXct(readable\_timestamp),  
 e4\_rr=e4\_rr\*1000)   
filter\_time\_day1\_e4 <- user19\_l %>%  
 filter(strftime(readable\_timestamp, "%H:%M:%S") >= start & strftime(readable\_timestamp, "%H:%M:%S") <= end, date(readable\_timestamp)==dte)

# Select rows for user 19 from ECG

user19\_fb <- fbdat %>%  
 filter(user=="lwp2\_0019") %>%   
 mutate(readable\_timestamp=as.POSIXct(readable\_timestamp))   
filter\_time\_day1\_fb <- user19\_fb %>%  
 filter(strftime(readable\_timestamp, "%H:%M:%S") >= start & strftime(readable\_timestamp, "%H:%M:%S") <= end, date(readable\_timestamp)==dte)

# Select rows for user 19 from MSband

user19\_l\_mb <- mbdat %>%  
 filter(user=="lwp2\_0019", device\_location=="Left") %>%   
 mutate(readable\_timestamp=as.POSIXct(readable\_timestamp),  
 mb\_rr=(mb\_rr\*1000))   
filter\_time\_day1\_mb <- user19\_l\_mb %>%  
 filter(strftime(readable\_timestamp, "%H:%M:%S") >= start & strftime(readable\_timestamp, "%H:%M:%S") <= end, date(readable\_timestamp)==dte)

# Clean data

## Detect outlier, short RR-interval, based on z-score standardization.

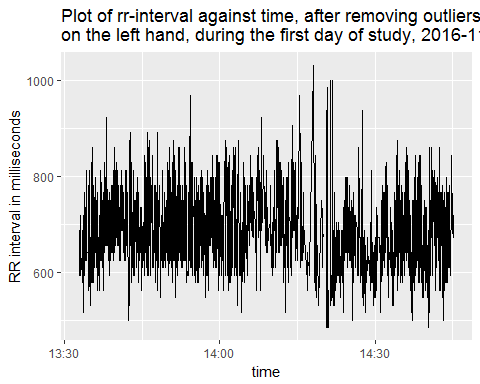
# Standardize function, standard deviation based on median absolute deviation, a robust estimate of standard deviation  
standardize <- function(rr){  
 return ((rr-mean(rr))/mad(rr))  
}  
  
# Function to identify outlier  
outlier\_short <- function(rr){  
 z <- standardize(rr)  
 ind <- which(z<(-3)) # -3: limit in sd for outlier detection of short rr interval  
 return (ind)  
}  
  
# Function to remove outlier, then add to subsequent beat  
rm\_short <- function(x,y){ # x is the index of outliers, y is the column of rr-interval  
 for (i in 1:length(x)){  
 if (y[x[i]]<length(y)){  
 y[x[i]+1] <- y[x[i]+1] + y[x[i]]  
 y[x[i]] <- NA  
 }  
 else{ # if rr-interval is at the end of the array, just remove it without adding it to the next beat  
 y[x[i]] <- NA  
 }  
 }  
 return (y)  
}  
  
# Combine all functions  
rm\_short\_comp <- function (rr){  
 indx <- outlier\_short(rr)  
 clean <- rm\_short(indx,rr)  
 return (clean)  
}

## Impute long rr-interval (Still working to optimize function)

# Identiy long rr-interval  
impute\_long <- function(rr){  
 dmax <- (2\*mad(rr)) + mean(rr)  
 dp <- numeric()  
 dp1 <- numeric()  
 ind\_long <- which(rr>dmax)  
 for (i in 1:length(ind\_long)){  
 dp <- sum((rr[ind\_long][i]-dmax), dp)  
 }  
 dp <- dp/length(ind\_long)  
 rr[ind\_long][i] <- rr[ind\_long][i]-dp  
 for (i in 1:length(rr[-ind\_long])){  
 dp1 <- sum((rr[-ind\_long][i]-dmax), dp1)  
 }  
 dp1 <- dp1/length(rr[-ind\_long])  
 rr[-ind\_long][i] <- rr[-ind\_long][i]-dp1  
 return (rr)  
}

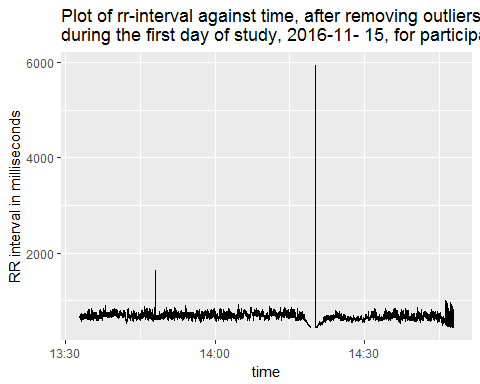
**E4**

filter\_time\_day1\_e4.clean <- filter\_time\_day1\_e4 %>%  
 mutate(e4\_rr=rm\_short\_comp(e4\_rr))  
  
# Plot  
ggplot(filter\_time\_day1\_e4.clean, aes(x=readable\_timestamp, y=e4\_rr)) + geom\_line() +  
 ggtitle("Plot of rr-interval against time, after removing outliers, measured by E4 \non the left hand, during the first day of study, 2016-11- 15, for participant 19.") +  
 ylab("RR interval in milliseconds") +   
 xlab("time")

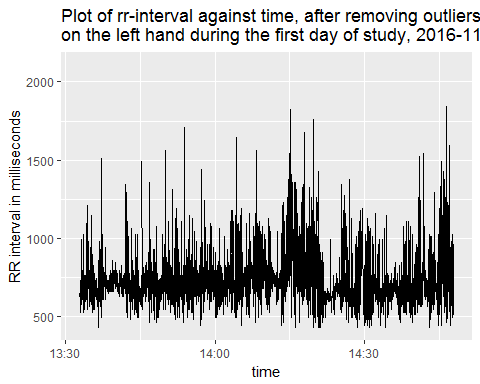


**ECG**

filter\_time\_day1\_fb.clean <- filter\_time\_day1\_fb %>%  
 mutate(fb\_rr=rm\_short\_comp(fb\_rr))  
  
# Plot  
ggplot(filter\_time\_day1\_fb.clean, aes(x=readable\_timestamp, y=fb\_rr)) + geom\_line() +  
 ggtitle("Plot of rr-interval against time, after removing outliers, measured by ECG \nduring the first day of study, 2016-11- 15, for participant 19.") +  
 ylab("RR interval in milliseconds") +   
 xlab("time")

 **MSband**

filter\_time\_day1\_mb.clean <- filter\_time\_day1\_mb %>%  
 mutate(mb\_rr=rm\_short\_comp(mb\_rr))  
  
# Plot  
ggplot(filter\_time\_day1\_mb.clean, aes(x=readable\_timestamp, y=mb\_rr)) + geom\_line() +  
 ggtitle("Plot of rr-interval against time, after removing outliers, measured by MSband \non the left hand during the first day of study, 2016-11- 15, for participant 19.") +  
 ylab("RR interval in milliseconds") +   
 xlab("time")



# RMSSD equation

# x should be the column containing rr-interval, units in millisecond  
HRV <- function(x){  
 total <- 0  
 for (i in 1:(length(x)-1)){  
 total <- sum(total, (x[i+1] -x[i])^2, na.rm=T)  
 }  
 return (sqrt(total/(length(x)-1)))  
}

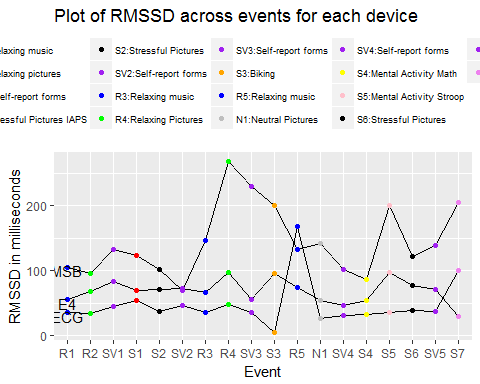
# Select each period and calculate RMSSD for E4, fb, and MSband

for (i in 3:(n-1)){  
 # select rr-interval for E4  
 rr\_int\_e4 <- filter\_time\_day1\_e4.clean %>%  
 filter(strftime(readable\_timestamp, "%H:%M")>=df$Time[i] & strftime(readable\_timestamp, "%H:%M") <df$Time[i+1])  
 # select rr-interval for firstbeat  
 rr\_int\_fb <- filter\_time\_day1\_fb.clean %>%  
 filter(strftime(readable\_timestamp, "%H:%M")>=df$Time[i] & strftime(readable\_timestamp, "%H:%M") < df$Time[i+1])  
 # select rr-interval for MSband  
 rr\_int\_mb <- filter\_time\_day1\_mb.clean %>%  
 filter(strftime(readable\_timestamp, "%H:%M")>=df$Time[i] & strftime(readable\_timestamp, "%H:%M") < df$Time[i+1])  
 # calculate RMSSD for each e4 interval  
 df$RMSSD\_e4[i] <- HRV(rr\_int\_e4$e4\_rr)  
 # calculate RMSSD for each firstbeat interval  
 df$RMSSD\_fb[i] <- HRV(rr\_int\_fb$fb\_rr)  
 # calculate RMSSD for each firstbeat interval  
 df$RMSSD\_mb[i] <- HRV(rr\_int\_mb$mb\_rr)  
}  
df$RMSSD\_e4[n] <- NA  
df$RMSSD\_fb[n] <- NA  
df$RMSSD\_mb[n] <- NA  
  
df #output

## # A tibble: 21 x 6  
## Event Time Activity RMSSD\_e4 RMSSD\_fb  
## <chr> <chr> <chr> <dbl> <dbl>  
## 1 Date 11/15/2016 <NA> NA NA  
## 2 Session 1 Start 13:32:52 <NA> NA NA  
## 3 R1 13:35 Relaxing music 54.81554 34.71556  
## 4 R2 13:44 Relaxing pictures 67.80185 34.25887  
## 5 SV1 13:46 Self-report forms 83.08122 43.96133  
## 6 S1 13:48 Stressful Pictures IAPS 69.27776 54.00338  
## 7 S2 14:00 Stressful Pictures 70.53724 36.60368  
## 8 SV2 14:02 Self-report forms 71.75808 45.70727  
## 9 R3 14:03 Relaxing music 66.28695 35.46144  
## 10 R4 14:14 Relaxing Pictures 97.18029 47.60630  
## # ... with 11 more rows, and 1 more variables: RMSSD\_mb <dbl>

# Plot HRV over different periods for each device

hrv\_df <- df %>%  
 slice(3:20) %>%  
 gather(`RMSSD\_e4`, `RMSSD\_fb`, `RMSSD\_mb`, key="Device", value="RMSSD") %>%  
 mutate(Device=str\_replace(Device, "^.\*\_", ""),  
 Event=factor(Event, levels=df$Event[3:20]))  
  
clr <- c("blue", "green", "purple", "red", "black", "purple", "blue", "green", "purple", "orange", "blue", "gray", "purple", "yellow", "pink", "black", "purple", "violet")  
ggplot(hrv\_df, aes(x=Event, y=RMSSD)) +  
 geom\_path(aes(group=Device)) +  
 geom\_point(aes(colour=Event)) +  
 #facet\_wrap(~Device) +  
 #coord\_flip() +   
 scale\_colour\_manual(values=clr,   
 labels=paste(hrv\_df$Event,hrv\_df$Activity, sep=":"),  
 name=NULL) +  
 annotate(geom="text", x=1, y=50, label="E4") +  
 annotate(geom="text", x=1, y=30, label="ECG") +  
 annotate(geom="text", x=1, y=100, label="MSB") +  
 ggtitle("Plot of RMSSD across events for each device") +  
 ylab("RMSSD in milliseconds") +  
 theme(legend.position="top",   
 legend.text=element\_text(size=7))



# Plot HRV across time window of 3min + 30s each, for E4. Working on function to increment by 30s.