Excite-O-Meter Pilot Data - Analysis

Anna Francova

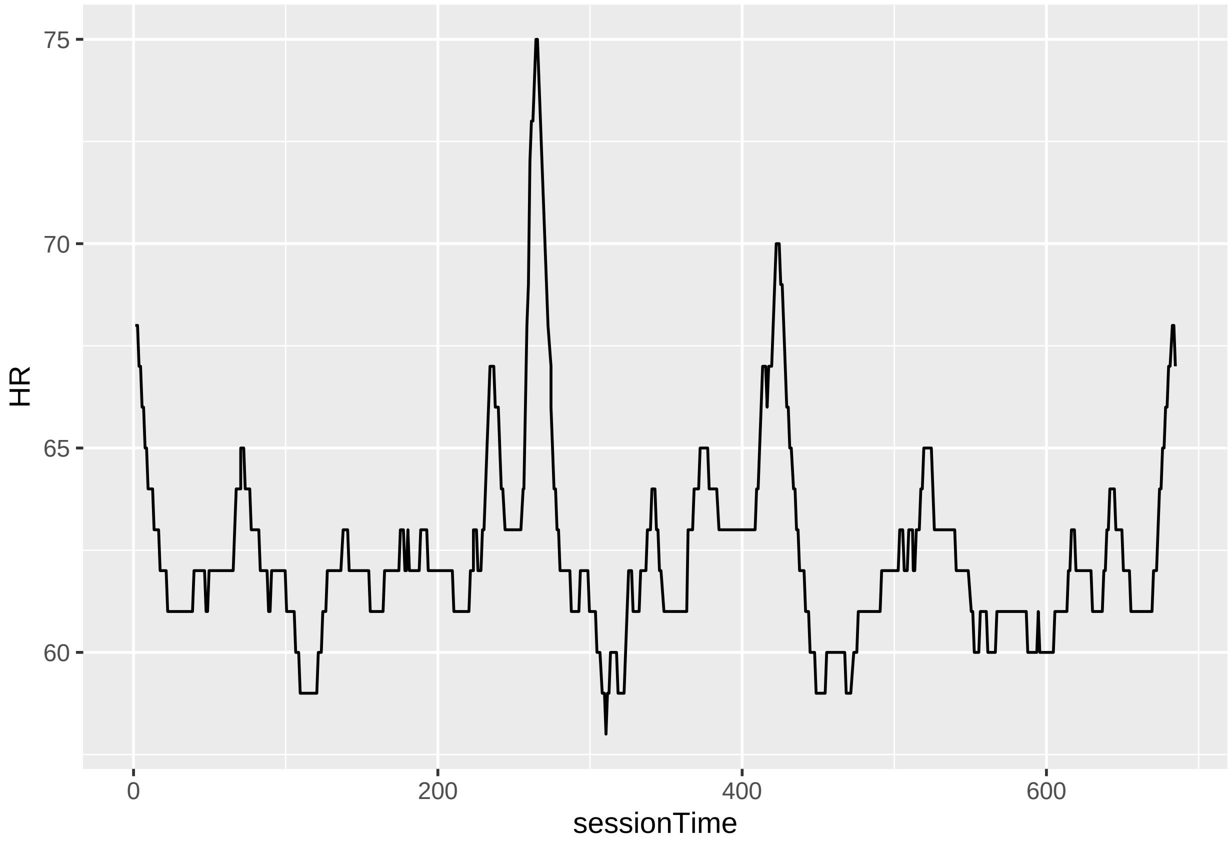
25. 9. 2020

Loading the data

Load participant's Polar H10 raw data - RRIs, RMSDDs, SDNNs, HR and create a basic HR line plot.

```r
RRI_raw <- read.csv("C:/Users/anyaf/Desktop/EoM_data/P3/rri.csv")
RMSDD_raw <- read.csv("C:/Users/anyaf/Desktop/EoM_data/P3/RMSSD.csv")
SDNN_raw <- read.csv("C:/Users/anyaf/Desktop/EoM_data/P3/SDNN.csv")
HR_raw <- read.csv("C:/Users/anyaf/Desktop/EoM_data/P3/hr.csv")

#install.packages("ggplot2")
library(ggplot2)
ggplot(data=HR_raw, aes(x=sessionTime, y=HR, group=1)) +
  geom_line()
```



Load the JSON sessionData and save timestamps data.

```r
#install.packages(rjson)
library("rjson")
sessionData <- fromJSON(file = "C:/Users/anyaf/Desktop/EoM_data/P3/sessionData.json")
timestamps <- data.frame(Reduce(rbind, sessionData[["manualMarkers"]]))
```

Dividing the data

Divide the **HR** data according to the timestamps (~ parts of the experiment).

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
HR_divided <- HR_raw %>%
  mutate(period = case_when(
    sessionTime > timestamps$timestamp[9] ~ "rest5",
    sessionTime > timestamps$timestamp[8] ~ "video2",
    sessionTime > timestamps$timestamp[7] ~ "rest4",
    sessionTime > timestamps$timestamp[6] ~ "video0",
    sessionTime > timestamps$timestamp[5] ~ "rest3",
    sessionTime > timestamps$timestamp[4] ~ "video3",
    sessionTime > timestamps$timestamp[3] ~ "rest2",
    sessionTime > timestamps$timestamp[2] ~ "video1",
    sessionTime > 0 ~ "baseline"

))
```

Divide the **RRI** data according to the timestamps (~ parts of the experiment).

```r
library(dplyr)


RRI_divided <- RRI_raw %>%
  mutate(period = case_when(
    sessionTime > timestamps$timestamp[9] ~ "rest5",
    sessionTime > timestamps$timestamp[8] ~ "video2",
    sessionTime > timestamps$timestamp[7] ~ "rest4",
    sessionTime > timestamps$timestamp[6] ~ "video0",
    sessionTime > timestamps$timestamp[5] ~ "rest3",
    sessionTime > timestamps$timestamp[4] ~ "video3",
    sessionTime > timestamps$timestamp[3] ~ "rest2",
    sessionTime > timestamps$timestamp[2] ~ "video1",
    sessionTime > 0 ~ "baseline"

))
```

Divide the **RMSDD** data according to the timestamps (~ parts of the experiment).

```
library(dplyr)


RMSDD_divided <- RMSDD_raw %>%
  mutate(period = case_when(
    sessionTime > timestamps$timestamp[9] ~ "rest5",
    sessionTime > timestamps$timestamp[8] ~ "video2",
    sessionTime > timestamps$timestamp[7] ~ "rest4",
    sessionTime > timestamps$timestamp[6] ~ "video0",
    sessionTime > timestamps$timestamp[5] ~ "rest3",
    sessionTime > timestamps$timestamp[4] ~ "video3",
    sessionTime > timestamps$timestamp[3] ~ "rest2",
    sessionTime > timestamps$timestamp[2] ~ "video1",
    sessionTime > 0 ~ "baseline"

))
```

Divide the **SDNN** data according to the timestamps (~ parts of the experiment).

```
library(dplyr)


SDNN_divided <- SDNN_raw %>%
  mutate(period = case_when(
    sessionTime > timestamps$timestamp[9] ~ "rest5",
    sessionTime > timestamps$timestamp[8] ~ "video2",
    sessionTime > timestamps$timestamp[7] ~ "rest4",
    sessionTime > timestamps$timestamp[6] ~ "video0",
    sessionTime > timestamps$timestamp[5] ~ "rest3",
    sessionTime > timestamps$timestamp[4] ~ "video3",
    sessionTime > timestamps$timestamp[3] ~ "rest2",
    sessionTime > timestamps$timestamp[2] ~ "video1",
    sessionTime > 0 ~ "baseline"

))
```

Merge the values of low (video 0 + 1) and high arousing (video 2 + 3) videos.

Descriptive Statistics

Calculate means of all values (RRI, RMSDD, SDNN and HR) for all 9 conditions.