High Frequency Checks (HFCs)

Keegan Brown, Elena Spielmann, Nate Spilka

Reading in and cleaning data

The following code chunk reads and cleans activity and heart rate Fitbit data. Data can be found here.

```
library(tidyverse)
library(lubridate)

# load data
map(
  list.files(
    'data',
    full.names = TRUE),
  ~ read_csv(.) %>%
    janitor::clean_names()) %>%
  set_names('daily_activity_data', 'hr_data') %>%
  list2env(.GlobalEnv)
```

```
<environment: R_GlobalEnv>
```

```
# clean up activity dates
activity_data <- daily_activity_data %>%
  mutate(
    date = mdy(activity_date)) %>%
  select(-activity_date)

# clean up hr dates
hr_data_2 <- hr_data %>%
  mutate(
    date_time = mdy_hms(time),
```

```
    date = date(date_time),
    hms = hms::as_hms(date_time),
    hr = value) %>%
  select(id, date, hms, hr)

# merging datasets
fitbit_data <-
  left_join(
    hr_data_2,
    activity_data,
    by = c('id' ,'date'))
```

Data flagging

The following code chunk flags observations and participants with noteworthy time spent sedentary (i.e., minutes spent sedentary exceeds the minutes in a day) and heart rates.

```
# hr lower limit
low_hr <- 39

# hr upper limit
high_hr <- 210

# minutes in a day
max_sed_min <- 1440

# updating the dataset to ensure the wearables are correctly recording hr
fitbit_data_flagged <-
  fitbit_data %>%
  # flags each observation where the hr exceeds 210 or is below 39
  mutate(
    obs_hr_flag = if_else(
      hr > high_hr | hr < low_hr,
      1,
      0)) %>%
  group_by(id) %>%
  # flags each participant where their hr has ever exceeded 210 or was below 39
  mutate(
    participant_hr_flag = if_else(
      any(hr > high_hr | hr < low_hr),
```

```
      1,
      0)) %>%
  ungroup() %>%
  group_by(id, date) %>%
  # flags observations where sedentary minutes exceeded a day (in minutes)
  mutate(
    sedantary_flag = if_else(
      sedentary_minutes > max_sed_min,
      1,
      0))
```

Sedentary Minutes:

One of the potential issues with collecting high-frequency data from wearable devices is the possibility of incomplete or erroneous data due to device malfunctions or user errors. One example of a high frequency check would be checking that the daily number of sedentary minutes does not exceed the total number of minutes in a day. Without this check, it is possible that the data could be skewed, leading to inaccurate or unreliable results.

To implement this check, we would check if the daily number of sedentary minutes exceeds the total number of minutes in a day (1440). If it does, then we can flag the data point as an outlier or erroneous data. By implementing this check, we can ensure that the data accurately reflects the users' sedentary behavior, which can have important implications for health and wellness.

Here, we can see the code and output for sedentary minutes that exceed 1440 from our dataset:

```
# here are the IDs and days that are flagged for sedentary minutes
fitbit_data_flagged %>%
  filter(sedantary_flag == 1) %>%
  distinct(id)
```

```
# A tibble: 0 x 2
# Groups:   id, date [0]
# ... with 2 variables: id <dbl>, date <date>
```

Heart Rate:

Another potential issue with high frequency data is the presence of outliers or errors that can affect the accuracy and reliability of the data. In the case of maximum heart rate, it is important to ensure that it does not exceed a physiologically plausible threshold, such as 210 beats per minute, or dip below 39 beats per minute (i.e., for our sample). This check helps identify outliers or errors in the data that could skew the results, as well as ensures the data accurately reflects the user's heart rate patterns. To implement this check, we would simply need to identify the maximum and minimum heart rate for each user and check if it exceeds the physiological threshold of 210 beats per minute or falls under 39 beats per minute. If it does, we can flag the data point as an outlier or erroneous data. We could also investigate whether the high heart rate is due to a device malfunction or user error, and take appropriate action to correct any issues. By implementing this check, we can improve the overall quality of the data, which can lead to more accurate and reliable insights into users' cardiovascular health.

For both of these high frequency checks, we can flag the data points that fail the check and remove them from the dataset, or we can keep them in the dataset and mark them as outliers for further investigation. We can also set up automated alerts to notify us when a data point fails the check, so that we can take immediate action to investigate and resolve any issues. Overall, implementing these checks can help to ensure the accuracy and reliability of the high-frequency data and improve the quality of insights we can derive from it.

Here, we can see the code and output fro heart rate from our dataset:

```r
# here are the IDs and days that are flagged for hr
fitbit_data_flagged %>%
  filter(obs_hr_flag == 1) %>%
  distinct(id)
```

```
# A tibble: 2 x 2
# Groups:   id, date [2]
          id date
       <dbl> <date>
1 2022484408 2016-04-27
2 5577150313 2016-05-04
```