Processing Google Takeout Fitbit Data

Genevieve Roberts

2025-08-25

Load Packages and Setup Functions and Constants

```
library(here)
library(tidyverse)
library(scales)
```

```
source(here::here("fitbit/takeout_fitbit_processing_functions.R"))
```

```
#current path
data_path <- here::here("fitbit/sample_fitbit_takeout_data/9Aug25_groberts_fitbit_takeout/Fitbit")
print(data_path)
```

```
## [1] "/Users/gen-omix/Documents/umass/VIGOR-surveys/fitbit/sample_fitbit_takeout_data/9Aug25_groberts_
```

Explore some FitBit data

```
#define some constants for the nb
start_date="2025-07-07"
end_date="2025-08-09"
```

Heart Rate Variablility

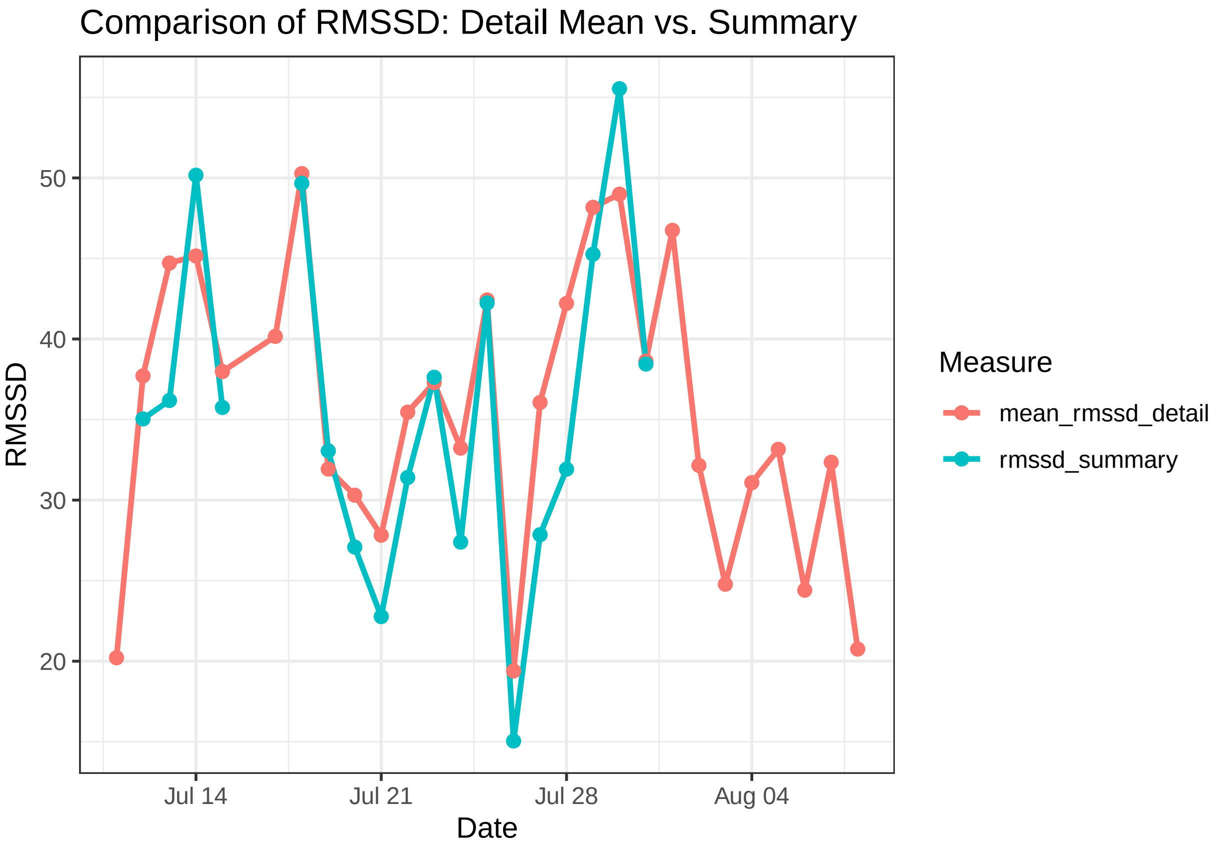
```
# Combined detailed + summary
hrv_data <- load_fitbit_hrv(start_date = start_date,
                            end_date = end_date,
                            root_dir = data_path,
                            summary_only = FALSE)

# Only summary data
hrv_data_only <- load_fitbit_hrv(start_date = start_date,
                                end_date = end_date,
                               root_dir = data_path,
                               summary_only = TRUE)

pander(sample_n(hrv_data, 5))
```

Table 1: Table continues below

timestamp_detail	rmssd_detail	coverage	low_frequency	high_frequency	file_date	timestamp_summary	rmssd_summary	nremhr	entropy
2025-08-03 04:50:00	15.76	1.002	182	53.02	2025-08-03	NA	NA	NA	NA
2025-08-07 02:15:00	18.16	1.004	498.4	113.6	2025-08-07	NA	NA	NA	NA
2025-07-19 04:30:00	31.78	0.926	1264	202.1	2025-07-19	2025-07-19	33.05	72.56	2.614
2025-08-05 06:45:00	28.65	0.976	1505	479.7	2025-08-05	NA	NA	NA	NA
2025-08-03 04:05:00	22.71	1.005	144.6	62.6	2025-08-03	NA	NA	NA	NA

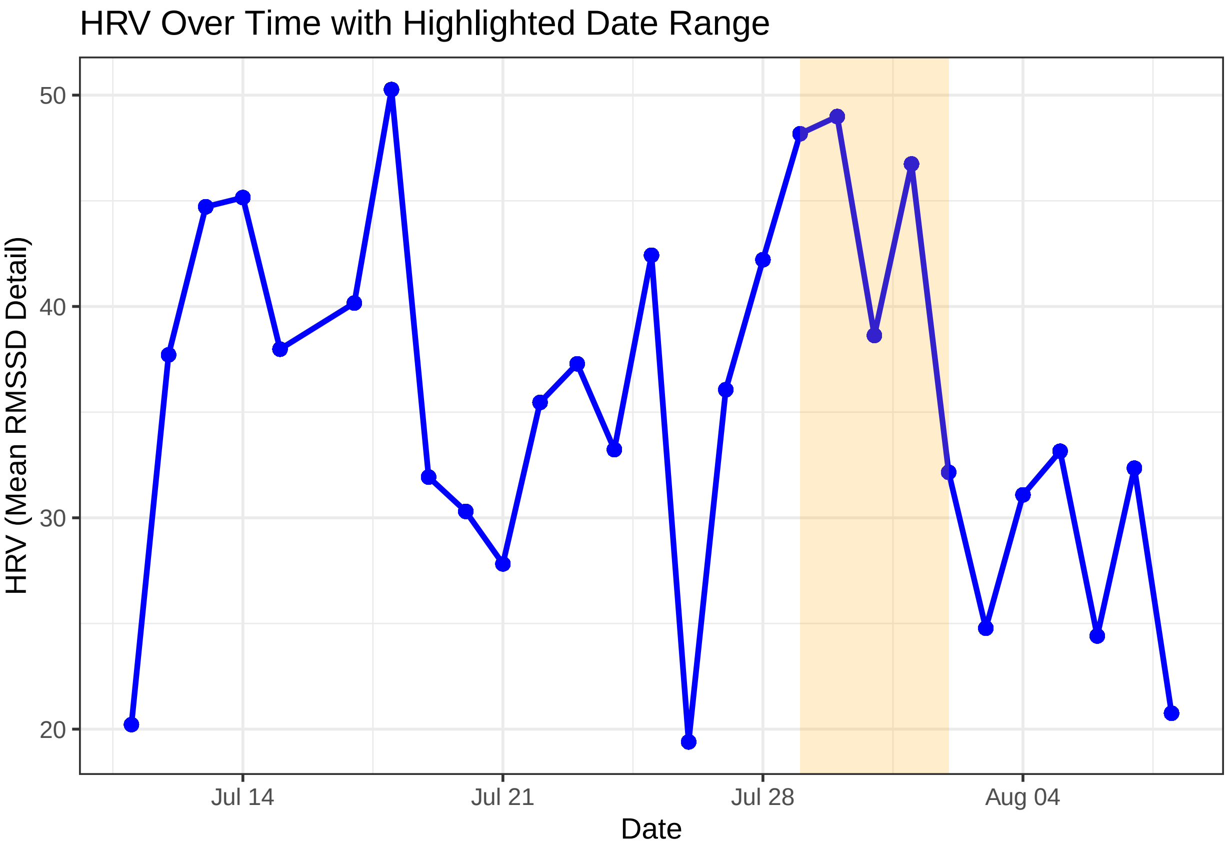
Here, I want to know if the `rmssd_summary` from the summary HRV files is simply the mean across all of the "detail" datapoints for that day. The plot below suggests they are not the same, but they are closely related.

```r
check_if_mean_equals_summary <- hrv_data %>%
  group_by(file_date) %>%
  summarize(
    mean_rmssd_detail = mean(rmssd_detail, na.rm = TRUE),
    rmssd_summary = first(rmssd_summary)   # summary has one value per date
  ) %>%
  ungroup()

# Prepare data in long format for plotting
mean_detail_compare_plot_df <- check_if_mean_equals_summary %>%
  pivot_longer(cols = c(mean_rmssd_detail, rmssd_summary),
               names_to = "Type",
               values_to = "RMSSD")

ggplot(mean_detail_compare_plot_df, aes(x = file_date, y = RMSSD, color = Type)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(
    title = "Comparison of RMSSD: Detail Mean vs. Summary",
    x = "Date",
    y = "RMSSD",
    color = "Measure"
  ) +
  theme_bw()
```



The plot below shows how we can highlight a period of interest

```r
# Example dates to highlight
highlight_start <- as.Date("2025-07-29")
highlight_end <- as.Date("2025-08-02")

#add the mean rmssd from the detailed HRV information
hrv_data <- hrv_data %>%
  group_by(file_date) %>%
  mutate(mean_rmssd_detail = mean(rmssd_detail)) %>%
  ungroup()

ggplot(hrv_data, aes(x = file_date, y = mean_rmssd_detail)) +
  geom_line(color = "blue", size = 1) +
  geom_point(color = "blue", size = 2) +

  # Highlight date range with a transparent rectangle
  annotate(
    "rect",
    xmin = highlight_start, xmax = highlight_end,
    ymin = -Inf, ymax = Inf,
    alpha = 0.2, fill = "orange"
  ) +

  labs(
    title = "HRV Over Time with Highlighted Date Range",
    x = "Date",
    y = "HRV (Mean RMSSD Detail)"
  ) +

  theme_bw()
```


HRV Over Time with Highlighted Date Range
HRV (Mean RMSSD Detail)
50
40
30
20
Jul 14
Jul 21
Jul 28
Aug 04
Date

Add Resting Heart Rate Data

```r
rhr_data <- load_fitbit_resting_hr(start_date = start_date,
                                   end_date = end_date,
                                   root_dir = data_path)

# Combine by day (default)
combined <- combine_fitbit_data(hrv_data, rhr_data)
pander(sample_n(combined, 5))
```

Table 3: Table continues below

file_date	timestamp_detail	rmssd_detail	coverage	low_frequency
2025-07-13	2025-07-13 04:40:00	57.24	0.888	1915
2025-08-04	2025-08-04 22:50:00	33.25	0.803	498.7
2025-07-22	2025-07-22 23:20:00	26.63	0.929	209.9
2025-07-22	2025-07-22 00:40:00	35.34	1	850.4
2025-08-05	2025-08-05 04:45:00	28.32	0.951	682.9

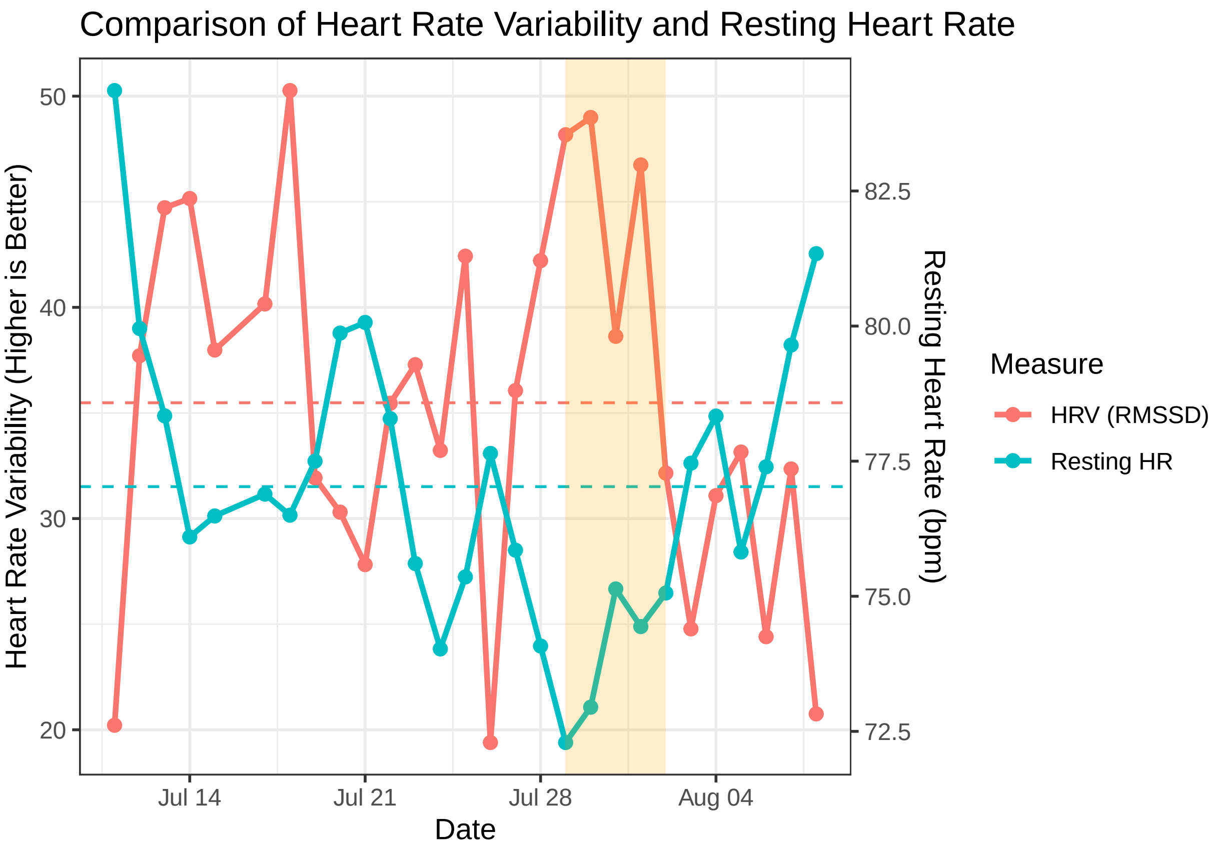
Table 4: Table continues below

high_frequency	timestamp_summary	rmssd_summary	nremhr	entropy
882	2025-07-13	36.18	73.27	2.75
408.2	NA	NA	NA	NA
251.3	2025-07-22	31.4	68.38	2.606
478.6	2025-07-22	31.4	68.38	2.606
295.8	NA	NA	NA	NA

mean_rmssd_detail	resting_hr
44.72	78.34
31.08	78.33
35.46	78.29
35.46	78.29
33.15	75.82

The plot below compares heart rate variability to resting heart rate. You can see there is a rough inverse correlation.

```r
# Data prep
compare_rhr_hrv <- combined %>%
  group_by(file_date) %>%
  summarize(
    mean_rmssd_detail = mean(rmssd_detail, na.rm = TRUE),
    resting_hr = first(resting_hr)
  ) %>%
  ungroup()

# observed HR range
min_hr <- min(compare_rhr_hrv$resting_hr, na.rm = TRUE)
max_hr <- max(compare_rhr_hrv$resting_hr, na.rm = TRUE)

# scaling factor using observed ranges
scale_factor <- (max(compare_rhr_hrv$mean_rmssd_detail, na.rm = TRUE) -
                 min(compare_rhr_hrv$mean_rmssd_detail, na.rm = TRUE)) /
                (max_hr - min_hr)

# means
mean_hrv <- mean(compare_rhr_hrv$mean_rmssd_detail, na.rm = TRUE)
mean_rhr <- mean(compare_rhr_hrv$resting_hr, na.rm = TRUE)

# get default ggplot colors (2-category hue palette)
plot_colors <- hue_pal()(2)

# Assign colors to measures consistently
measure_colors <- c("HRV (RMSSD)" = plot_colors[1],
                    "Resting HR"  = plot_colors[2])

ggplot(compare_rhr_hrv, aes(x = file_date)) +
  geom_line(aes(y = mean_rmssd_detail, color = "HRV (RMSSD)"), size = 1) +
  geom_point(aes(y = mean_rmssd_detail, color = "HRV (RMSSD)"), size = 2) +
  geom_line(aes(
    y = (resting_hr - min_hr) * scale_factor +
        min(compare_rhr_hrv$mean_rmssd_detail, na.rm = TRUE),
    color = "Resting HR"
  ), size = 1) +
  geom_point(aes(
    y = (resting_hr - min_hr) * scale_factor +
        min(compare_rhr_hrv$mean_rmssd_detail, na.rm = TRUE),
    color = "Resting HR"
  ), size = 2) +

  # mean lines matching legend colors
  geom_hline(yintercept = mean_hrv, linetype = "dashed", color = measure_colors["HRV (RMSSD)"]) +
  geom_hline(
    yintercept = (mean_rhr - min_hr) * scale_factor +
                 min(compare_rhr_hrv$mean_rmssd_detail, na.rm = TRUE),
    linetype = "dashed", color = measure_colors["Resting HR"]
  ) +
  scale_color_manual(values = measure_colors) +
  scale_y_continuous(
    name = "Heart Rate Variability (Higher is Better)",
    sec.axis = sec_axis(
      ~ (. - min(compare_rhr_hrv$mean_rmssd_detail, na.rm = TRUE)) / scale_factor + min_hr,
      name = "Resting Heart Rate (bpm)"
    )
  ) +
  labs(
    title = "Comparison of Heart Rate Variability and Resting Heart Rate",
    x = "Date",
    color = "Measure"
  ) +
  annotate(
    "rect",
    xmin = highlight_start, xmax = highlight_end,
    ymin = -Inf, ymax = Inf,
    alpha = 0.2, fill = "orange"
  ) +
  theme_bw()
```


Comparison of Heart Rate Variability and Resting Heart Rate
Heart Rate Variability (Higher is Better)
Resting Heart Rate (bpm)
Jul 14
Jul 21
Jul 28
Aug 04
Date
Measure
HRV (RMSSD)
Resting HR