Capstone Project on Bellabeat

Sung Park

11/29/23

Introduction

For my second capstone project with Bellabeat [1], I will analyze customer data from smart fitness devices, focusing on the FitBit Fitness Tracker—a benchmark for the company's entry into the smart fitness tracker market.

Bellabeat, known for its high-tech health-focused products for women, aims to venture into the smart fitness tracker business, positioning itself as a competitor to **FitBit Fitness Tracker**.

Despite being a smaller company, Bellabeat has already achieved success in its current business and has the potential to emerge as a global leader in the smart health device market.

This analysis will provide Bellabeat with valuable insights into how customers engage with the market's most popular devices, detailing the process of analyzing FitBit Fitness Tracker's datasets. The comprehensive approach of this analysis covers six critical steps: asking meaningful questions, preparing and processing data, conducting in-depth analysis, effectively sharing insights, and making informed decisions.

1. Ask

Guiding Questions and Analysis Strategy Overview

Question #1: What are some trends in smart fitness device usage?

- Strategy: Explore and analyze the trends in customers' smart fitness device usage, focusing on key patterns and behaviors.

Question #2: How could these trends apply to Bellabeat customers?

- Strategy: Examine the implications of identified trends on Bellabeat's customers, understanding how their behavior aligns with or deviates from broader industry trends.

Question #3: How could these trends help influence Bellabeat marketing strategy?

- Strategy: Investigate the potential impact of identified trends on Bellabeat's marketing strategy, considering opportunities for alignment or adaptation.

Conclusion: The synthesis of these analyses will inform a strategic optimization of Bellabeat's marketing approach, ensuring that it resonates effectively with the evolving landscape of smart fitness device usage.

Business Task: Optimize Bellabeat's Marketing Strategy

2. Prepare

Key tasks

2.1. Download data and identify how it's organized.

Downloaded **FitBit Fitness Tracker Data** [2] and imported it into RStudio.

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.2     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.2     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becon
```

```
daily_activity <- read.csv("dailyActivity_merged.csv")
daily_calories <- read.csv("dailyCalories_merged.csv")
daily_intensities <- read.csv("dailyIntensities_merged.csv")
daily_steps <- read.csv("dailySteps_merged.csv")
heartrate_seconds <- read.csv("heartrate_seconds_merged.csv")
hourly_calories <- read.csv("hourlyCalories_merged.csv")
hourly_intensities <- read.csv("hourlyIntensities_merged.csv")
```

```r
hourly_steps <- read.csv("hourlySteps_merged.csv")
minute_calories_narrow <- read.csv("minuteCaloriesNarrow_merged.csv")
minute_calories_wide <- read.csv("minuteCaloriesWide_merged.csv")
minute_intensities_narrow <- read.csv("minuteIntensitiesNarrow_merged.csv")
minute_intensities_wide <- read.csv("minuteIntensitiesWide_merged.csv")
minute_METs_narrow <- read.csv("minuteMETsNarrow_merged.csv")
minute_sleep <- read.csv("minuteSleep_merged.csv")
minute_steps_narrow <- read.csv("minuteStepsNarrow_merged.csv")
minute_steps_wide <- read.csv("minuteStepsWide_merged.csv")
sleep_day <- read.csv("sleepDay_merged.csv")
weight_loginfo <- read.csv("weightLogInfo_merged.csv")
```

Check the number of unique IDs in all datasets using **`n_distinct()`** in the **`tibble()`**, creating a summary table for each dataset.

```
# A tibble: 18 x 2
   Dataset                   Unique_IDs
   <chr>                          <int>
 1 daily_activity                    33
 2 daily_calories                    33
 3 daily_intensities                 33
 4 daily_steps                       33
 5 heartrate_seconds                 14
 6 hourly_calories                   33
 7 hourly_intensities                33
 8 hourly_steps                      33
 9 minute_calories_narrow            33
10 minute_calories_wide              33
11 minute_intensities_narrow         33
12 minute_intensities_wide           33
13 minute_METs_narrow                33
14 minute_sleep                      24
15 minute_steps_narrow               33
16 minute_steps_wide                 33
17 sleep_day                         24
18 weight_loginfo                     8
```

This tibble table shows that I have assembled data for a maximum of 33 individuals in these datasets.

Due to the limitation of time, I will concentrate on four datasets: daily_activity, heartbeat_seconds, hourly_calories, and minute_sleep.

With the daily_activity dataset, I will draw overall patterns for all 33 individuals. Using heartbeat_seconds, hourly_calories, and minute_sleep, I will follow a unique individual, combining three different pieces of information to observe how they affect each other.

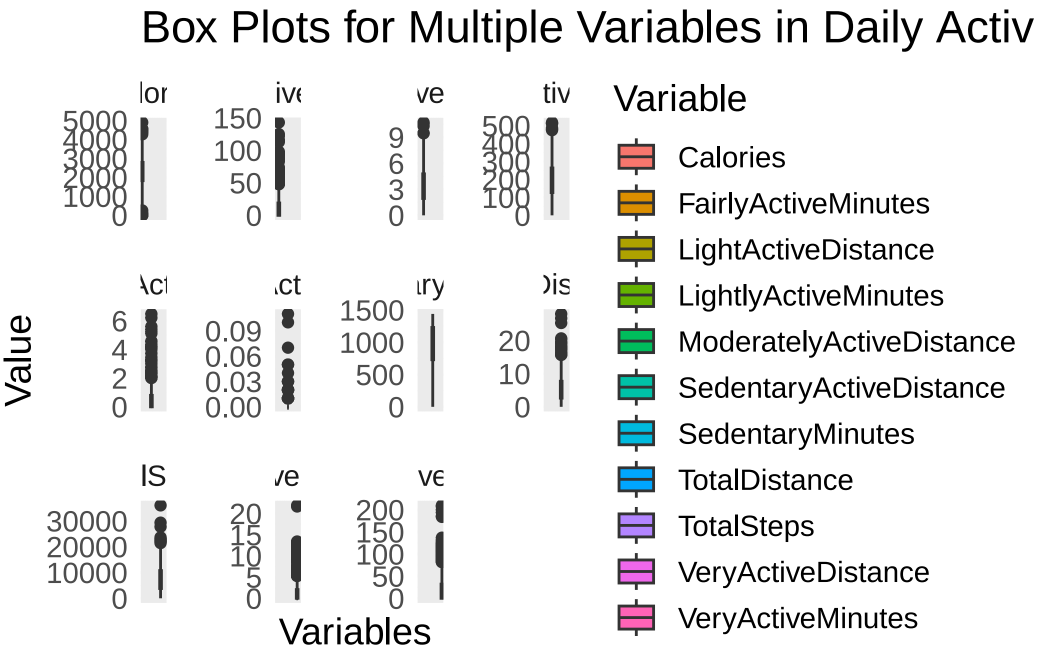
2.2. Sort and filter the data

In the following analyses, I frequently employ the **`glimpse`** function to gain insights into the structure and characteristics of various datasets.

2.2.1. A glimpse of the **daily_activity** dataset:

```
Rows: 940
Columns: 15
$ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
$ ActivityDate             <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/~
$ TotalSteps               <int> 13162, 10735, 10460, 9762, 12669, 9705, 13019~
$ TotalDistance            <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
$ TrackerDistance          <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
$ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ VeryActiveDistance       <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5~
$ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3~
$ LightActiveDistance      <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0~
$ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ VeryActiveMinutes        <int> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~
$ FairlyActiveMinutes      <int> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~
$ LightlyActiveMinutes     <int> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~
$ SedentaryMinutes         <int> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~
$ Calories                 <int> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203~
```

I am creating a boxplot to visually represent the distribution of variables within the **daily_activity** dataset.

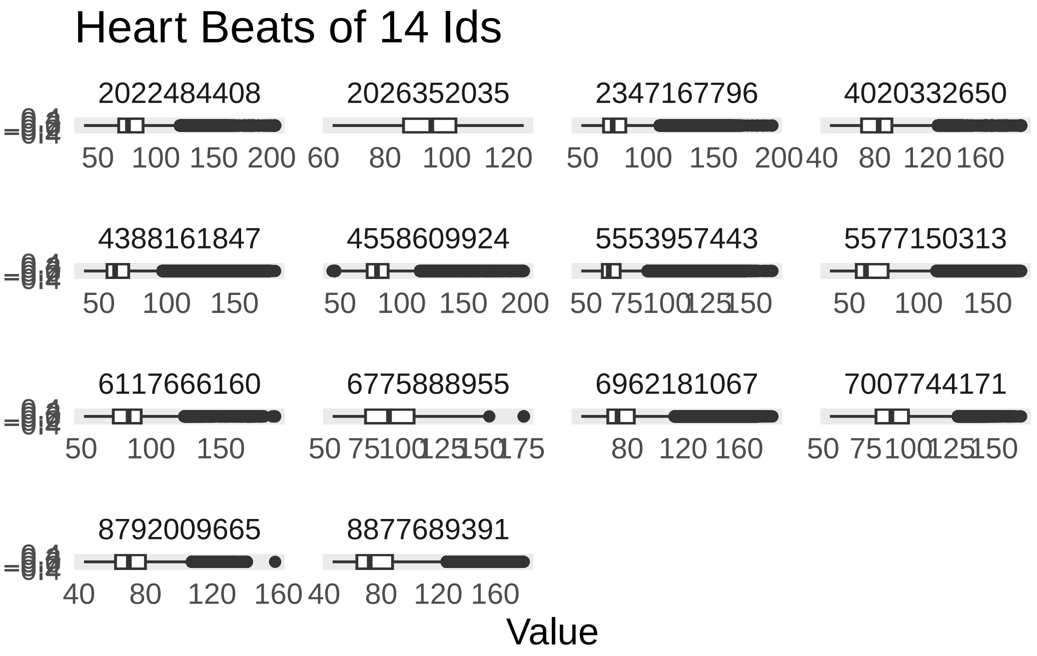


2.2.2. A glimpse of the **heartbeat_seconds** dataset:

```
Rows: 2,483,658
Columns: 3
$ Id    <dbl> 2022484408, 2022484408, 2022484408, 2022484408, 2022484408, 2022~
$ Time  <chr> "4/12/2016 7:21:00 AM", "4/12/2016 7:21:05 AM", "4/12/2016 7:21:~
$ Value <int> 97, 102, 105, 103, 101, 95, 91, 93, 94, 93, 92, 89, 83, 61, 60, ~
```

After utilizing the `glimpse` function to explore the structure of the **heartbeat_seconds** dataset, I find that it contains measurements of the heartbeats for fourteen individuals recorded at five-second intervals. This granular level of data allows for a detailed examination of heart rate dynamics over time.

I am generating a boxplot to visualize the distribution of heart rates recorded every five seconds for 14 participants in the **heartbeat_seconds** dataset. This graphical representation provides insights into the variability and central tendencies of heart rate measurements across the selected individuals.



2.2.3. A glimpse of **hourly__calories** dataset

```
Rows: 22,099
Columns: 3
$ Id           <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150396036~
$ ActivityHour <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM", "4/12/20~
$ Calories     <int> 81, 61, 59, 47, 48, 48, 48, 47, 68, 141, 99, 76, 73, 66, ~
```

2.2.4. A glimpse of the **minute__sleep** dataset:

```
Rows: 188,521
Columns: 4
$ Id    <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 1503960366, 1503~
$ date  <chr> "4/12/2016 2:47:30 AM", "4/12/2016 2:48:30 AM", "4/12/2016 2:49:~
$ value <int> 3, 2, 1, 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 2, 1, 1, 1, 1, 1, 1~
$ logId <dbl> 11380564589, 11380564589, 11380564589, 11380564589, 11380564589,~
```

3. Process

This table displays the classes of variables such as Time in the heartrate_second dataset, ActivityHour in the hourly_calories dataset, and date in the minute_sleep dataset. Understanding the classes of these variables is crucial for further analysis and interpretation of the data.

Variables	Current class	Desired class
Time in heartrate_second	character	POSIXct
ActivityHour in hourly_calories	character	POSIXct
date in minute_sleep	character	POSIXct

To accurately represent date and time information, it is essential to transform these three time-related variables – Time in the heartrate_second dataset, ActivityHour in the hourly_calories dataset, and date in the minute_sleep dataset – from character to the POSIXct class.

During the data cleaning process, entries with a value of zero in some Ids' activities were identified. In order to ensure data tidiness and accuracy, these entries will be appropriately treated as missing values (NAs).

4. Analyze

4.1. Analysis of Sleep hours

I am initiating the analysis of the minute_sleep dataset with the goal of extracting time information for when participants entered and woke up each day. This involves deciphering a list of long consecutive data to precisely identify specific bed entry and wake-up times for each participant.

To streamline this process and extract accurate time information, I am developing a custom function. This function takes the participant's ID as input and generates a comprehensive list of time data, indicating when they sleep and wake up each day. The meticulous design of this function ensures precise and reliable processing of data from the minute_sleep dataset, facilitating a detailed exploration of participants' sleep patterns.

With the completion of the function, the next step involves inputting all participant IDs to obtain the corresponding list of sleep time information for each individual. This comprehensive approach allows for a collective analysis of sleep patterns across all participants in the minute_sleep dataset.

```r
library(purrr)
```

```
unique_ids <- unique(minute_sleep$Id)
sleep_result_list <- map(unique_ids, small_sleep_sequence)
sleep_result_df <- bind_rows(sleep_result_list)
```

Thanks to the powerful operation of the **map** function in the **purrr** library, I was able to efficiently obtain the sleep hours of all participants. Demonstrating the speed and effectiveness of this approach, I am printing a preview of the results, showcasing the first 10 lines for initial insights.

```
           Id          start_time            end_time sleep_hours
1  1503960366 2016-04-12 02:47:30 2016-04-12 08:32:30    05:45:00
2  1503960366 2016-04-13 03:08:30 2016-04-13 08:20:30    05:12:00
3  1503960366 2016-04-13 20:10:00 2016-04-13 21:43:00    01:33:00
4  1503960366 2016-04-15 02:59:00 2016-04-15 10:20:00    07:21:00
5  1503960366 2016-04-16 02:11:00 2016-04-16 06:59:00    04:48:00
6  1503960366 2016-04-16 07:02:00 2016-04-16 08:19:00    01:17:00
7  1503960366 2016-04-16 23:27:00 2016-04-17 11:18:00    11:51:00
8  1503960366 2016-04-19 02:06:30 2016-04-19 07:25:30    05:19:00
9  1503960366 2016-04-20 02:01:00 2016-04-20 08:17:00    06:16:00
10 1503960366 2016-04-21 02:32:30 2016-04-21 08:35:30    06:03:00
```

Before summarizing the **minute_sleep** dataset, I observed that the original IDs, consisting of 10-digit long integers, might pose challenges in plotting when the x-axis is crowded. To address this, I have assigned shorter one-letter IDs alongside the original ones.

```
library(dplyr)
library(lubridate)

sleep_result_df <- sleep_result_df |>
  mutate(sleep_hours_seconds = as.numeric(hms(sleep_hours)))


sleep_summary <- sleep_result_df |>
  group_by(Id) |>
  summarise(
    n = n(),
    mean_sleep_seconds = mean(sleep_hours_seconds, na.rm = TRUE),
    median_sleep_seconds = median(sleep_hours_seconds, na.rm = TRUE),
    min_sleep_seconds = min(sleep_hours_seconds, na.rm = TRUE),
    max_sleep_seconds = max(sleep_hours_seconds, na.rm = TRUE)
  )
```

```
unique_ids <- unique(sleep_summary$Id)
custom_labels <- LETTERS[seq_along(unique_ids)]

sleep_result_df$CustomId <- factor(sleep_result_df$Id, levels = unique_ids, labels = custo

sleep_summary$CustomId <- factor(sleep_summary$Id, levels = unique_ids, labels = custom_la

sleep_summary <- sleep_summary |>
  select(Id, CustomId, n, mean_sleep_seconds, median_sleep_seconds,
         min_sleep_seconds, max_sleep_seconds )
```

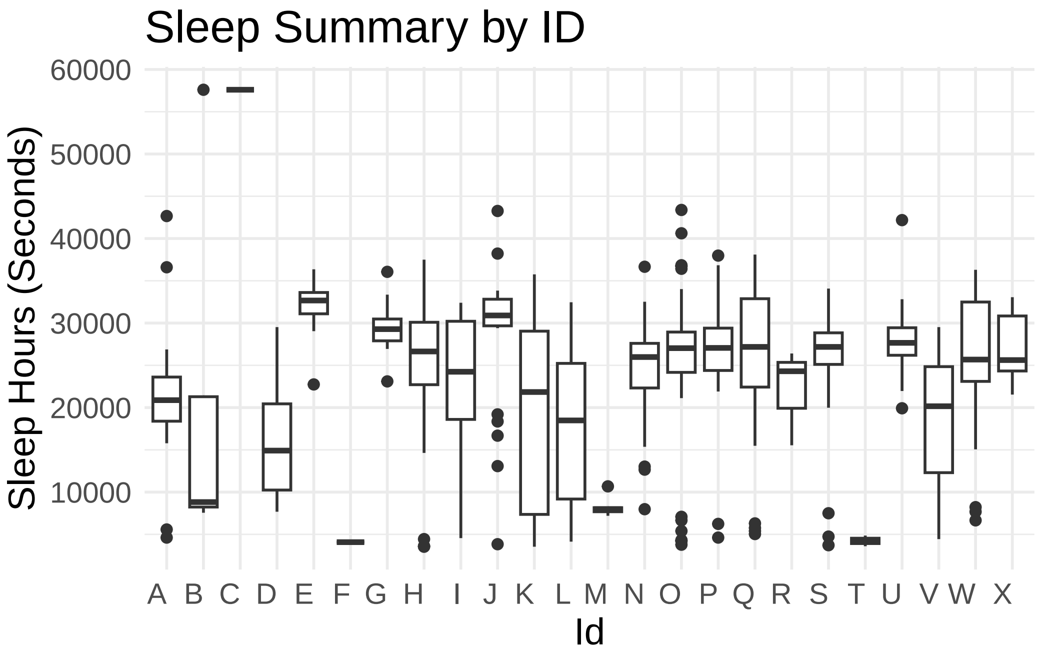
Now, with the data appropriately prepared, here is the summary.

```
# A tibble: 24 x 7
           Id CustomId     n mean_sleep_seconds median_sleep_seconds
        <dbl> <fct>    <int>              <dbl>                <dbl>
 1 1503960366 A           27             21229.                20880
 2 1644430081 B            4             20700                  8820
 3 1844505072 C            3             57600                 57600
 4 1927972279 D            8             16358.                14910
 5 2026352035 E           28             32199.                32670
 6 2320127002 F            1              4080                  4080
 7 2347167796 G           15             29420                 29280
 8 3977333714 H           32             24150                 26640
 9 4020332650 I            8             22725                 24240
10 4319703577 J           27             28942.                30900
# i 14 more rows
# i 2 more variables: min_sleep_seconds <dbl>, max_sleep_seconds <dbl>
```

Stable entry patterns for sleep time information, spanning over more than 10 days, are evident in IDs A, E, G, H, and J, while other IDs (B, C, D, F, and I) exhibit fewer entries.

To visually capture the distribution of sleep patterns across all participants, I have created a boxplot as a comprehensive summary of the **minute_sleep** dataset.

```
ggplot(sleep_result_df, aes(x = factor(CustomId), y = sleep_hours_seconds)) +
  geom_boxplot() +
  labs(title = "Sleep Summary by ID", x = "Id", y = "Sleep Hours (Seconds)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0, hjust = 1))  +
   theme(text = element_text(size = 14))
```



The length of the box in a boxplot is representative of the interquartile range (IQR), serving as an indicator of the spread or dispersion of the middle 50% of the data.

While the box length offers insights into the spread of the middle 50% of the data, considerations of outliers and the length of the whiskers further contribute to a comprehensive understanding of the dataset's distribution.

Taking into account the length of the box, the whiskers, and the presence of outliers, I have identified ID G as exhibiting low variability. This suggests that values for this participant tend to be closer to the mean or median, making the data more predictable, and thereby enhancing the reliability of statistical models.

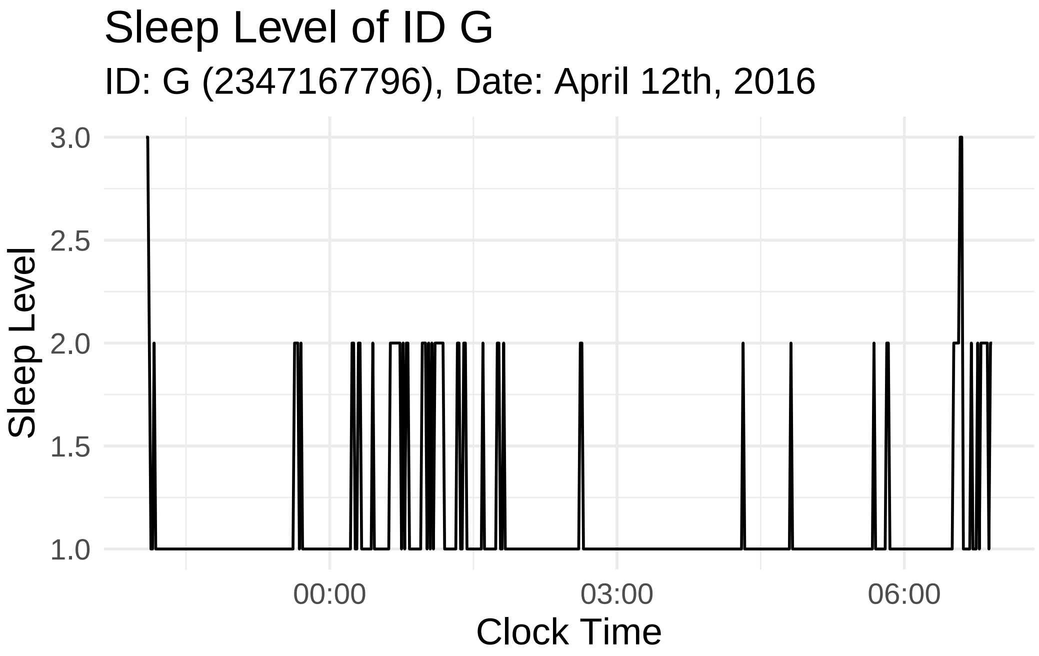
For the subsequent analyses, I will focus on examining the heart rate, sleep level, and calories burned for ID G (numerical ID 2347167796), leveraging the consistent and less variable nature of the data for more robust insights.

4.2. Analysis of Id G's sleep_level

Utilizing my **`small_sleep_sequence`** function, I extracted all sleep_hour data entries. Through this process, I discovered that the first entry of ID G's sleep hour record was from April 12, 2016, 10:05 PM, extending until the following morning at 06:55 AM.

Leveraging this detailed sleep hours information, I proceeded to create a plot illustrating the sleep level of ID G, offering a visual representation of their sleep patterns during this specific time frame.

```r
sleep_data <- minute_sleep[minute_sleep$Id == '2347167796', ]
sleep_data$Time <- as.POSIXct(sleep_data$date, format = "%m/%d/%Y %I:%M:%S %p")

start_time <- as.POSIXct("2016-04-12 22:05:00", format = "%Y-%m-%d %H:%M:%S")
end_time <- as.POSIXct("2016-04-13 06:55:00", format = "%Y-%m-%d %H:%M:%S")

sleep_at <- value <- numeric(0)
sleep_at <- vector("numeric")
value <- vector("numeric")
j <- 1
for (i in 1:nrow(sleep_data)) {
  row = sleep_data[i,]

  if(row$Time >= start_time & row$Time <= end_time){
    sleep_at[j] = row$Time
    value[j] <- row$value
    j <- j + 1
  }
}
sleep_level <- data.frame(
  sleep_at <- as.POSIXct(sleep_at, origin = "1970-01-01"),
  value <- value
)
plot <- ggplot(sleep_level)
plot + geom_line(aes(x = sleep_at, y = value)) +
  labs(
    title = "Sleep Level of ID G",
    subtitle = "ID: G (2347167796), Date: April 12th, 2016 ",
    x = "Clock Time",
    y = "Sleep Level") +
  theme_minimal() +
  theme(text = element_text(size = 14))
```



- Here's a common interpretation of the sleep level in health monitoring apps:

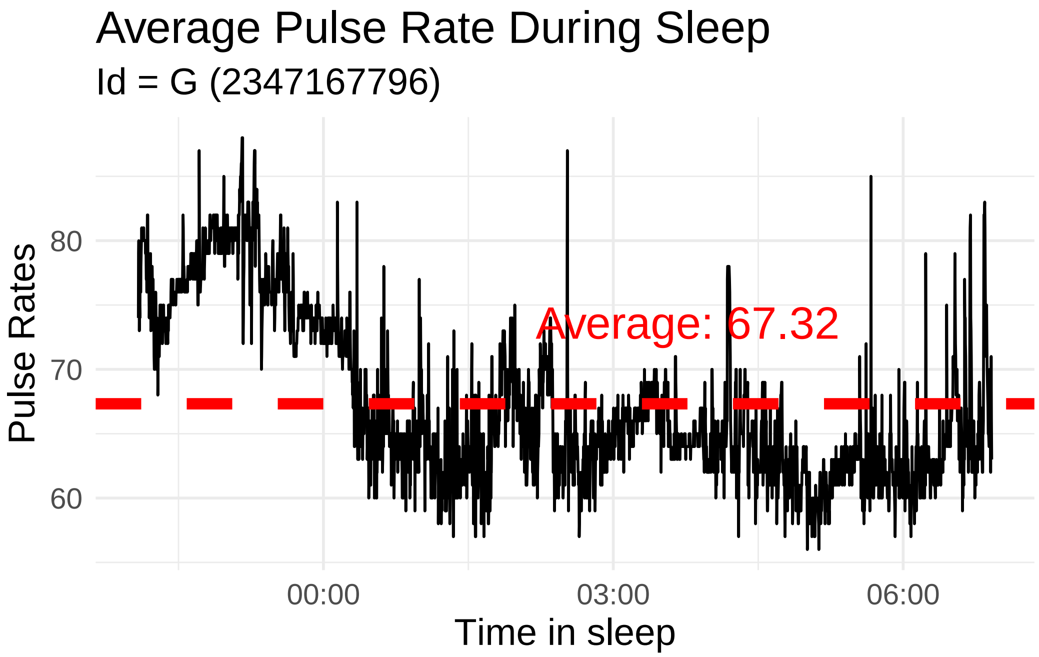
  1. **Sleep Level 1:** Typically signifies light sleep, a transitional phase between wakefulness and deeper sleep. Easier to wake up, associated with the early sleep cycle.
  2. **Sleep Level 2:** Represents a deeper stage with a slower heart rate, reduced body temperature, and increased muscle relaxation. Slightly harder to wake up than Sleep Level 1.
  3. **Sleep Level 3:** Associated with the deepest and most restorative sleep, known as slow-wave sleep (SWS) or deep sleep. Crucial for physical and mental restoration, involving essential repair and rejuvenation processes.

4.3. Analysis of Id G's Heart Rate

4.3.1. During Sleep

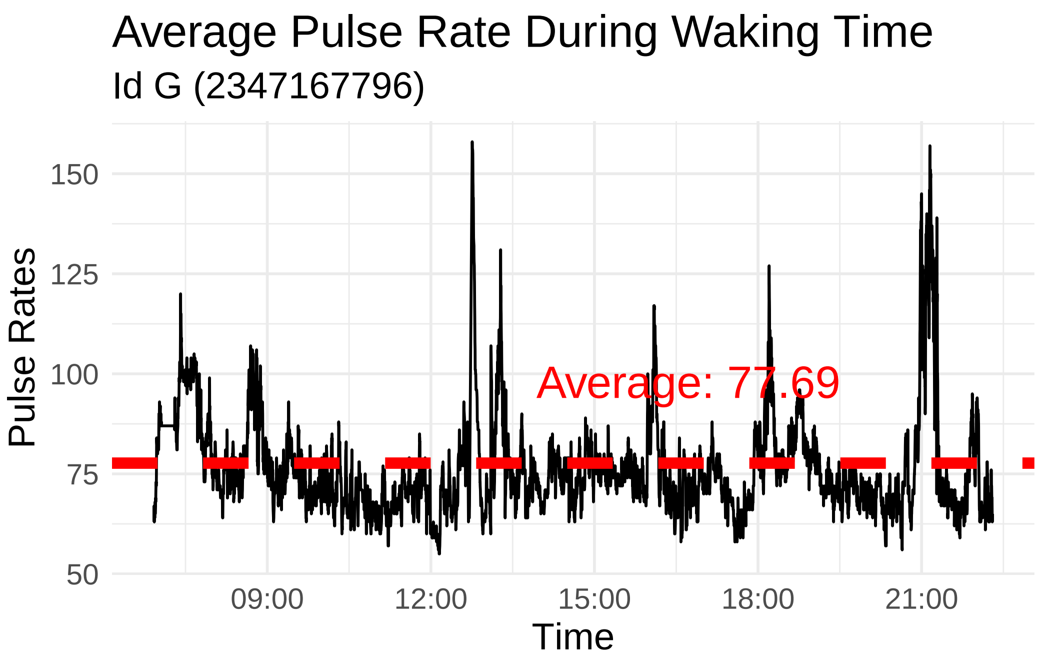
```
heartbeat_data <- heartrate_seconds[heartrate_seconds$Id == '2347167796', ]
heartbeat_data$Time <- as.POSIXct(heartbeat_data$Time, format = "%m/%d/%Y %I:%M:%S %p")

cnt <- 0
p_cnt <- 0
j <- 0
```

```r
pulse_at <- value <- numeric(0)
pulse_at <- vector("numeric")
value <- vector("numeric")

for (i in (1:nrow(heartbeat_data))) {
  row <- heartbeat_data[i,]
  start_sleep <- as.POSIXct("2016-04-12 22:05:00", format = "%Y-%m-%d %H:%M:%S")
  end_sleep <- as.POSIXct("2016-04-13 06:55:00 ", format = "%Y-%m-%d %H:%M:%S")

  if (row$Time >= start_sleep & row$Time <= end_sleep) {
    cnt = cnt + 1
    j <- j +1
    p_cnt = p_cnt + 1

    pulse_at[j] <- row$Time
    value[j] <- row$Value

#    if(p_cnt == 100) {
#      cat(" cnt and row$Time and value ", cnt, format(row$Time, "%Y#-%m-%d %H:%M:%S"), ro
#      p_cnt = 0
#    }
  }
}
sleep_heartrate <- data.frame(
    pulse_at = as.POSIXct(pulse_at, origin = "1970-01-01"),
    value = value
  )

average_value <- mean(sleep_heartrate$value)
plot <- ggplot(sleep_heartrate)
plot + geom_line(aes(x = pulse_at, y = value)) +
  geom_hline(yintercept = average_value, linetype = "dashed", linewidth = 2, color = "red"

  annotate("text", x = max(sleep_heartrate$pulse_at), y = average_value,
           label = paste("Average:", round(average_value, 2)),
           hjust = 1.5, vjust = -2, color = "red", size = 6) +
    labs(
    title = "Average Pulse Rate During Sleep",
    subtitle = "Id = G (2347167796)",
    x = "Time in sleep",
    y = "Pulse Rates") +
  theme_minimal() +
  theme(text = element_text(size = 14))
```



4.3.2. During Waking Hours

```
cnt <- 0
p_cnt <- 0
j <- 0
pulse_at <- value <- numeric(0)
pulse_at <- vector("numeric")
value <- vector("numeric")

for (i in (1:nrow(heartbeat_data))) {
  row <- heartbeat_data[i,]
  start_sleep <- as.POSIXct("2016-04-12 22:05:00", format = "%Y-%m-%d %H:%M:%S")
  end_sleep <- as.POSIXct("2016-04-13 06:55:00", format = "%Y-%m-%d %H:%M:%S")
  end_of_day <- as.POSIXct("2016-04-13 22:18:00", format = "%Y-%m-%d %H:%M:%S")

  if (row$Time > end_sleep & row$Time <= end_of_day) {
    cnt = cnt + 1
    j <- j +1
    p_cnt = p_cnt + 1

    pulse_at[j] <- row$Time
    value[j] <- row$Value

#    if(p_cnt == 100) {
#      cat(" cnt and row$Time and value ", cnt, format(row$Time, "%Y-%m-%d %H:%M:%S"), row
#      p_cnt = 0
#    }
  }
}
active_heartrate <- data.frame(
  pulse_at = as.POSIXct(pulse_at, origin = "1970-01-01"),
  value = value
)
average_value <- mean(active_heartrate$value)
plot <- ggplot(active_heartrate)
plot + geom_line(aes(x = pulse_at, y = value)) +
  geom_hline(yintercept = average_value, linetype = "dashed", linewidth = 2, color = "red"
  annotate("text", x = max(active_heartrate$pulse_at), y = average_value,
           label = paste("Average:", round(average_value, 2)),
           hjust = 1.5, vjust = -2, color = "red", size = 6) +
  labs(
    title = "Average Pulse Rate During Waking Time",
    subtitle = "Id G (2347167796)",
    x = "Time",
    y = "Pulse Rates") +
  theme_minimal() +
  theme(text = element_text(size = 14))
```

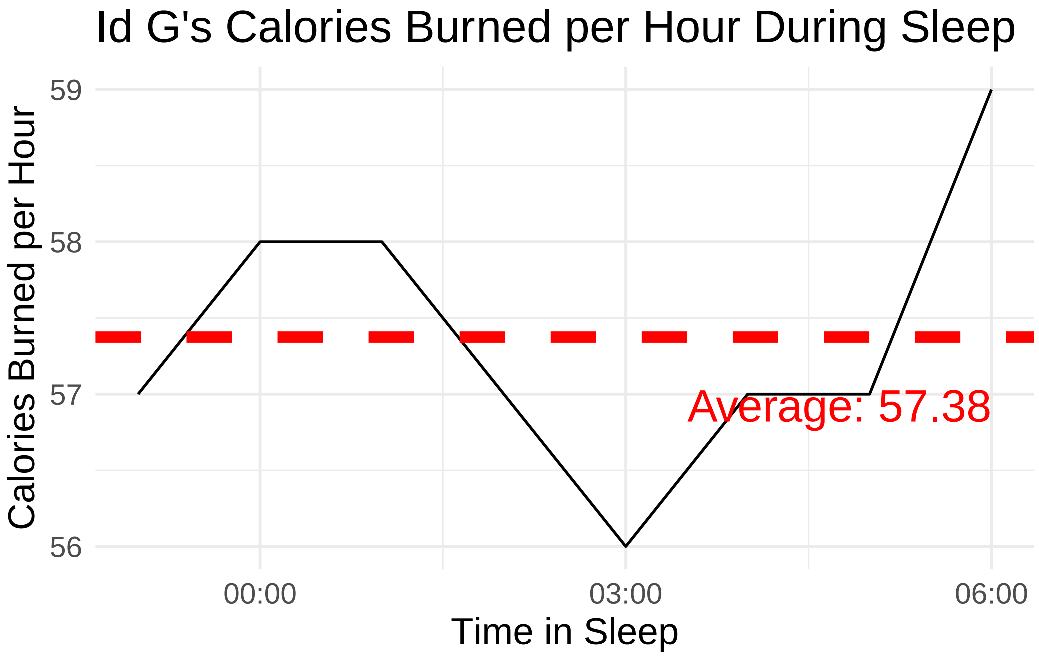


4.4. Analysis of ID G's Calories Burned

4.4.1. During Sleep

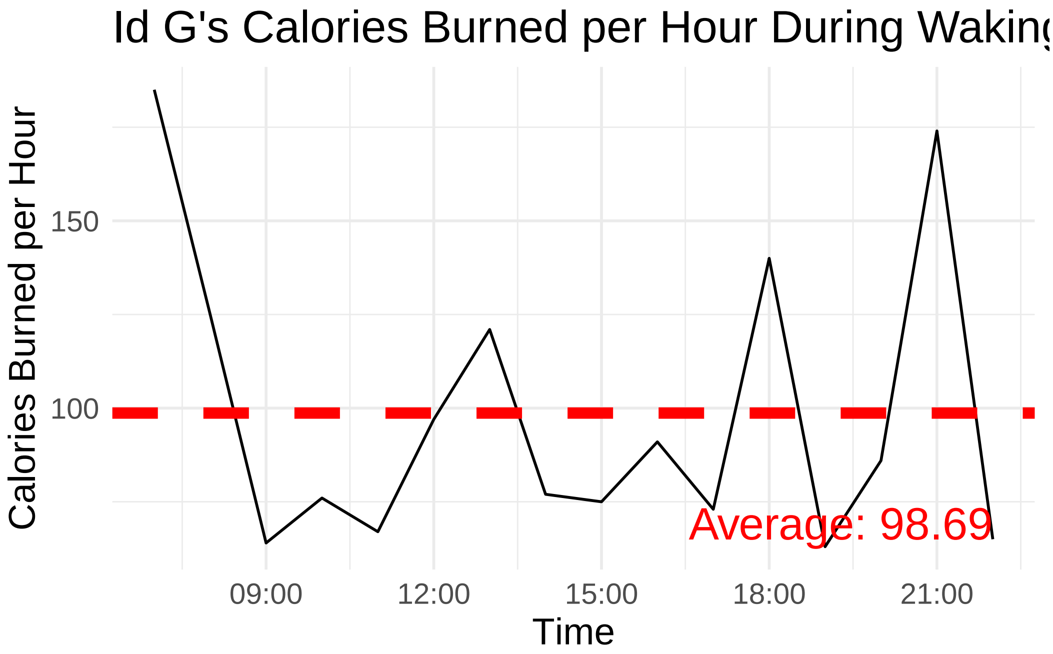
```
calorie_hourly <- hourly_calories[hourly_calories$Id == '2347167796', ]
calorie_hourly$Time <- as.POSIXct(calorie_hourly$ActivityHour, format = "%m/%d/%Y %I:%M:%S

start_time <- as.POSIXct("2016-04-12 22:05:00", format = "%Y-%m-%d %H:%M:%S")
end_time <- as.POSIXct("2016-04-13 06:55:00", format = "%Y-%m-%d %H:%M:%S")
end_of_day <- as.POSIXct("2016-04-13 22:18:00", format = "%Y-%m-%d %H:%M:%S")

calorie_at_sleep <- value_sleep <- calorie_at_daytime <- value_daytime <- numeric(0)
calorie_at_sleep <- vector("numeric")
value_sleep <- vector("numeric")
calorie_at_daytime <- vector("numeric")
value_daytime <- vector("numeric")

j <- 1
k <- 1
for(i in 1:nrow(calorie_hourly)) {
  row <- calorie_hourly[i,]
  if(row$Time >= start_time & row$Time <= end_time) {
```

```r
    calorie_at_sleep[j] <- row$Time
    value_sleep[j] <- row$Calories
    j <- j + 1
  }
  if(row$Time > end_time & row$Time <= end_of_day) {
    calorie_at_daytime[k] <- row$Time
    value_daytime[k] <- row$Calories
    k <- k + 1
  }
}
calorie_hourly_sleep <- data.frame(
  calorie_at_sleep <- as.POSIXct(calorie_at_sleep, origin = "1970-01-01"),
  value_sleep <- value_sleep
)

calorie_hourly_daytime <- data.frame(
  calorie_at_daytime <- as.POSIXct(calorie_at_daytime, origin = "1970-01-01"),
  value_daytime <- value_daytime
)
average_value <- mean(calorie_hourly_sleep$value)
plot <- ggplot(calorie_hourly_sleep)
plot + geom_line(aes(x = calorie_at_sleep, y = value_sleep)) +
  geom_hline(yintercept = average_value, linetype = "dashed", linewidth = 2, color = "red"
  annotate("text", x = max(calorie_hourly_sleep$calorie_at_sleep), y = average_value + 0.3
           label = paste("Average:", round(average_value, 2)),
           hjust = 1, vjust = 4, color = "red", size = 6) +
  labs(
    title = "Id G's Calories Burned per Hour During Sleep",
    x = "Time in Sleep",
    y = "Calories Burned per Hour") +
  theme_minimal() +
  theme(text = element_text(size = 14))
```



4.4.2. During Waking Hours

```
average_value_daytime <- mean(calorie_hourly_daytime$value)
plot <- ggplot(calorie_hourly_daytime)
plot + geom_line(aes(x = calorie_at_daytime, y = value_daytime)) +
  geom_hline(yintercept = average_value_daytime, linetype = "dashed", linewidth = 2, color
  annotate("text", x = max(calorie_hourly_daytime$calorie_at_daytime), y = average_value_d
           label = paste("Average:", round(average_value_daytime, 2)),
           hjust = 1, vjust = 4, color = "red", size = 6) +
  labs(
    title = "Id G's Calories Burned per Hour During Waking Hours",
    x = "Time",
    y = "Calories Burned per Hour") +
  theme_minimal() +
  theme(text = element_text(size = 14))
```



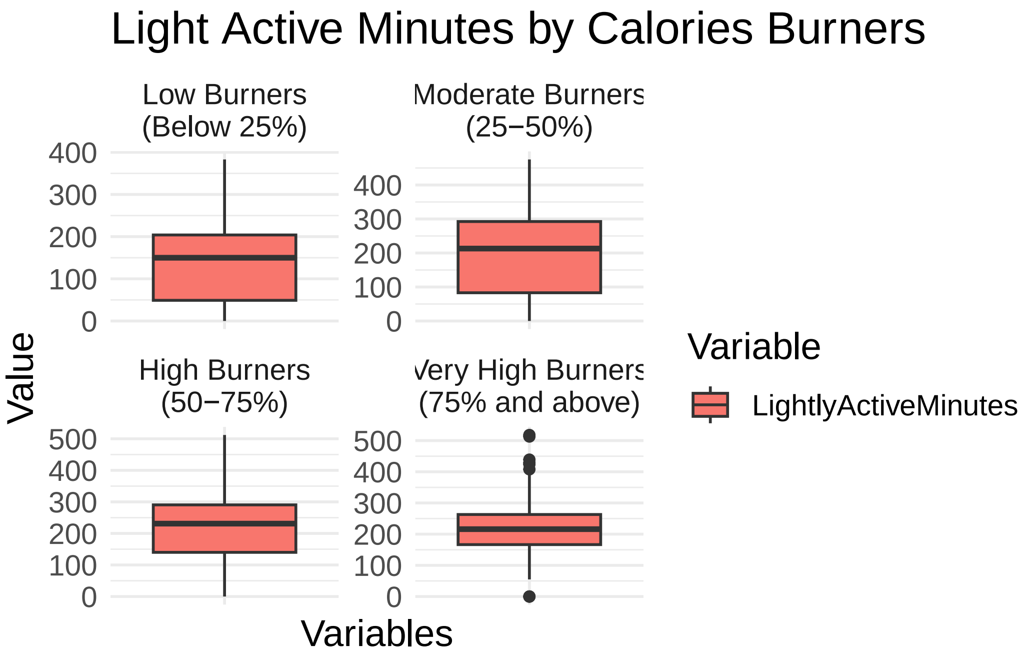
4.5. Analysis of daily activity

The daily activities of all participants, including walking distances and the duration of intensively active hours, are recorded, providing valuable insights into the calories burned during these activities. Therefore, this dataset serves as an excellent resource to identify and analyze the effectiveness of different types of activities in calorie consumption.

I categorized participants into four groups based on the amount of daily calories burned, creating distinct tiers. These tiers are defined as follows: the Top Tier (Very High Burners - 75% above), second tier (High Burners - 50 - 75%), third tier (Moderate Burners - 25 - 50%), and fourth tier (Low Burners - below 25%).

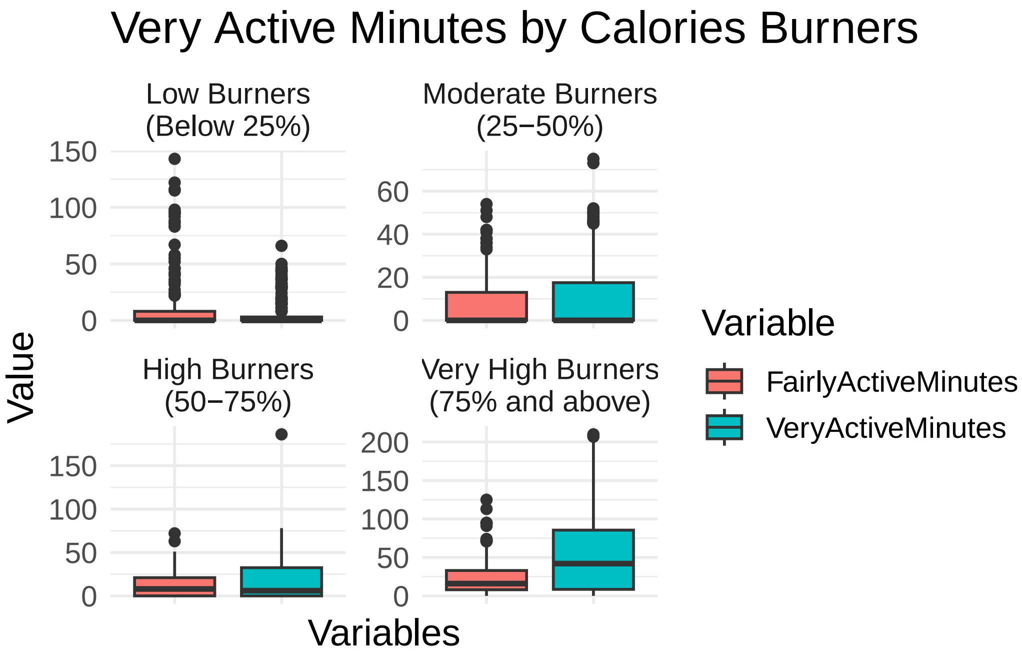
4.5.1. Light Active Minutes by Calories Burners

```
quartiles <- quantile(daily_activity$Calories, c(0.25, 0.5, 0.75))

daily_activity$calories <- cut(daily_activity$Calories, breaks = c(-Inf, quartiles, Inf),
            labels = c("Low Burners\n(Below 25%)", "Moderate Burners\n(25-50%)",
                       "High Burners\n(50-75%)", "Very High Burners\n(75% and above)"), inc

calories_lightly_active_minutes <- daily_activity |>
  pivot_longer(cols = c(#TotalSteps, TotalDistance, VeryActiveDistance,
                        #ModeratelyActiveDistance, LightActiveDistance,
                        #SedentaryActiveDistance,
                         LightlyActiveMinutes
                        # FairlyActiveMinutes, VeryActiveMinutes
                        #FairlyActiveMinutes, LightlyActiveMinutes,
                        #SedentaryMinutes
                        ),
               names_to = "Variable", values_to = "Value")
ggplot(calories_lightly_active_minutes, aes(x = Variable, y = Value, fill = Variable)) +
  geom_boxplot() +
  facet_wrap(~ calories, scales = "free_y") +
  theme_minimal() +
  labs(title = "Light Active Minutes by Calories Burners", x = "Variables", y = "Value") +
  theme(axis.text.x = element_blank()) +
  theme(text = element_text(size = 14))
```



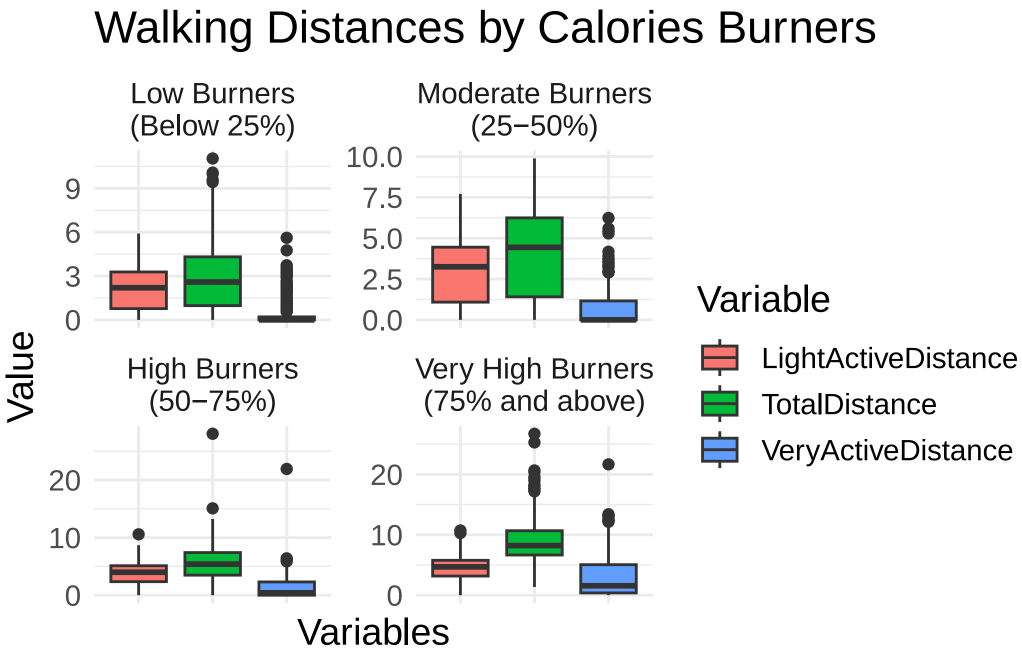
4.5.2. Very Active Minutes by Calories Burners

```r
calories_fairly_very_active_minutes <- daily_activity |>
  pivot_longer(cols = c(
                        #TotalSteps, TotalDistance, VeryActiveDistance,
                        #ModeratelyActiveDistance, LightActiveDistance,
                        #SedentaryActiveDistance,
                        #LightlyActiveMinutes
                        FairlyActiveMinutes, VeryActiveMinutes
                        #FairlyActiveMinutes, LightlyActiveMinutes,
                        #SedentaryMinutes
                       ),
  names_to = "Variable", values_to = "Value")
ggplot(calories_fairly_very_active_minutes, aes(x = Variable, y = Value, fill = Variable))
  geom_boxplot() +
  facet_wrap(~ calories, scales = "free_y") +
  theme_minimal() +
  labs(title = "Very Active Minutes by Calories Burners", x = "Variables", y = "Value") +
  theme(axis.text.x = element_blank()) +
  theme(text = element_text(size = 14))
```



4.5.3. Walking Distances Covered by Calorie Burners

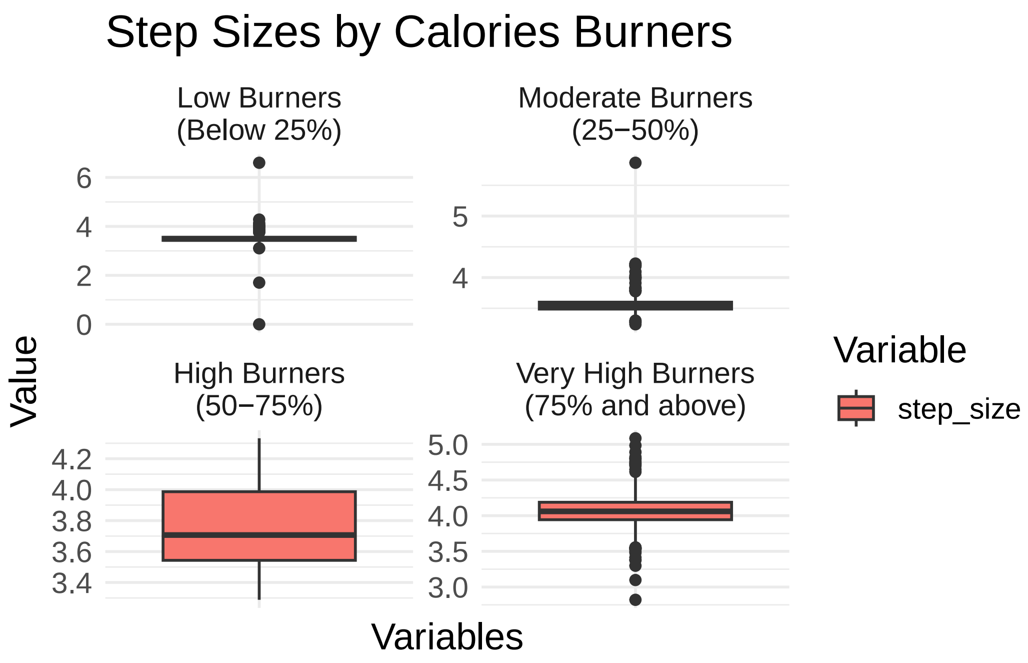
```
calories_active_distance <- daily_activity |>
  pivot_longer(cols = c(
    #TotalSteps,
    TotalDistance, VeryActiveDistance,
    LightActiveDistance
    #SedentaryActiveDistance,
    #ModeratelyActiveDistance,
    #LightlyActiveMinutes
    #FairlyActiveMinutes, VeryActiveMinutes
    #FairlyActiveMinutes, LightlyActiveMinutes,
    #SedentaryMinutes
  ),
  names_to = "Variable", values_to = "Value")
ggplot(calories_active_distance, aes(x = Variable, y = Value, fill = Variable)) +
  geom_boxplot() +
  facet_wrap(~ calories, scales = "free_y") +
  theme_minimal() +
  labs(title = "Walking Distances by Calories Burners", x = "Variables", y = "Value") +
  theme(axis.text.x = element_blank()) +
  theme(text = element_text(size = 14))
```



4.5.4. Step Sizes by Calories Burners

```
daily_activity <- daily_activity |>
  mutate(
    step_size = daily_activity$TotalDistance * 5280 /daily_activity$TotalSteps
  )
daily_step_size <- daily_activity |>
  pivot_longer(cols = c(
    #TotalSteps,
    #TotalDistance, VeryActiveDistance,
    #LightActiveDistance
    #SedentaryActiveDistance,
    #ModeratelyActiveDistance,
    #LightlyActiveMinutes
    #FairlyActiveMinutes, VeryActiveMinutes
    #FairlyActiveMinutes, LightlyActiveMinutes,
    step_size
  ),
  names_to = "Variable", values_to = "Value")
ggplot(daily_step_size, aes(x = Variable, y = Value, fill = Variable)) +
  geom_boxplot() +
  facet_wrap(~ calories, scales = "free_y") +
  theme_minimal() +
  labs(title = "Step Sizes by Calories Burners", x = "Variables", y = "Value") +
  theme(axis.text.x = element_blank()) +
  theme(text = element_text(size = 14))
```

```
Warning: Removed 77 rows containing non-finite values (`stat_boxplot()`).
```



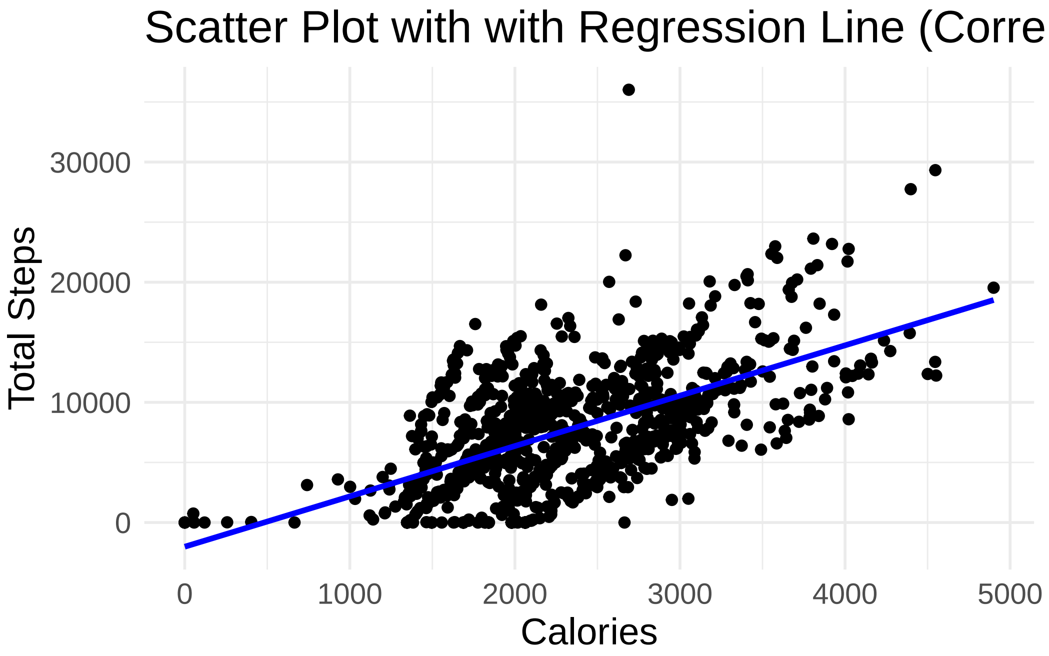
4.6. Analysis of Correlations

As an additional means of identifying effective daily activities for burning more calories, I plotted the correlation between calories burned and three activities: total steps, very active minutes, and total distance.

4.6.1. Correlation of Total Steps with Calories Burned

```
df <- daily_activity
x <- df$Calories
y <- df$TotalSteps

  ggplot(df, aes(x = x, y = y)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = paste("Scatter Plot with with Regression Line (Correlation =", round(cor(x,
  theme_minimal() +
  theme(text = element_text(size = 14))
```

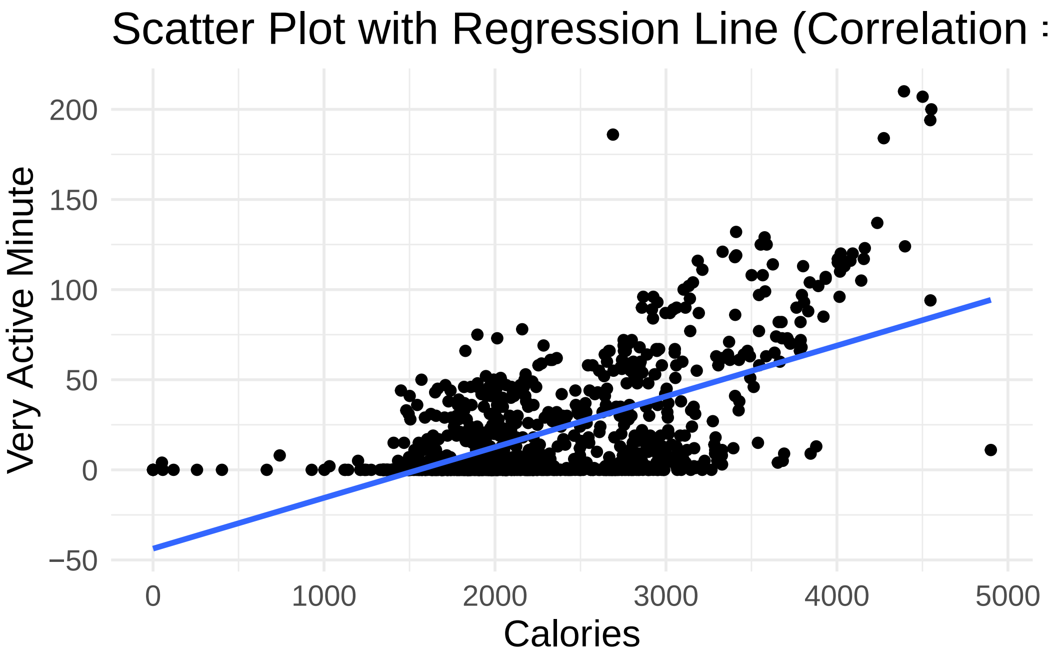
```
`geom_smooth()` using formula = 'y ~ x'
```



4.6.2. Correlation of Very Active Minute with Calories Burned

```
df <- daily_activity
x <- daily_activity$Calories
y <- daily_activity$VeryActiveMinutes
ggplot(df, aes(x = x, y = y)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = paste("Scatter Plot with Regression Line (Correlation =", round(cor(x, y),
   theme_minimal() +
  theme(text = element_text(size = 14))
```

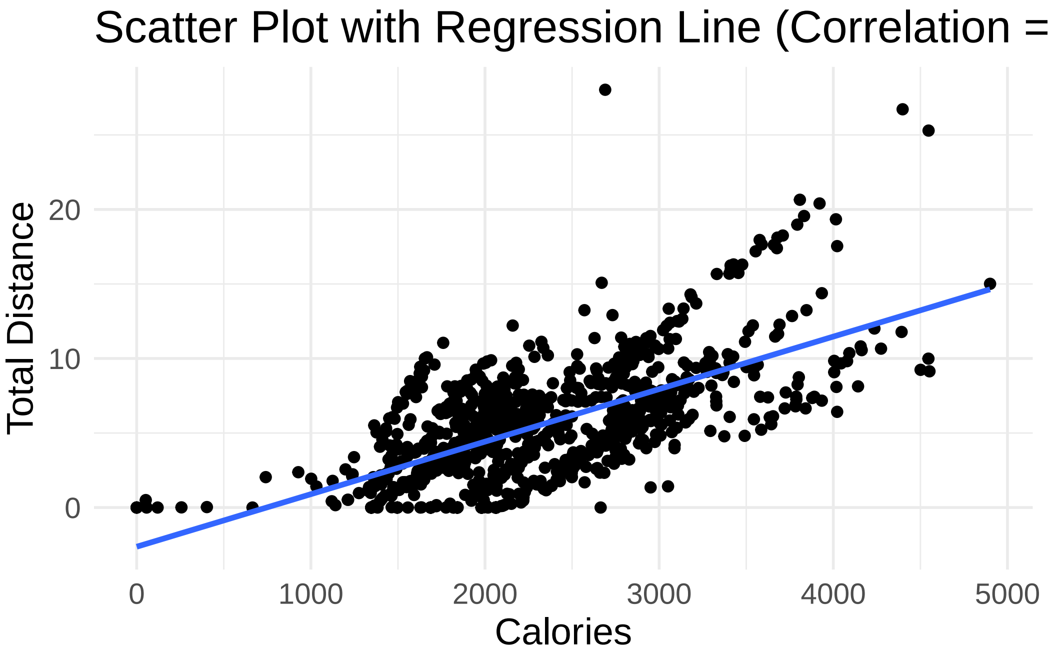
```
`geom_smooth()` using formula = 'y ~ x'
```



4.6.3. Correlation of Total Distance with Calories Burned

```
df <- daily_activity
x <- daily_activity$Calories
y <- daily_activity$TotalDistance
ggplot(df, aes(x = x, y = y)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = paste("Scatter Plot with Regression Line (Correlation =", round(cor(x, y),
   theme_minimal() +
  theme(text = element_text(size = 14))
```

```
`geom_smooth()` using formula = 'y ~ x'
```



5. Share

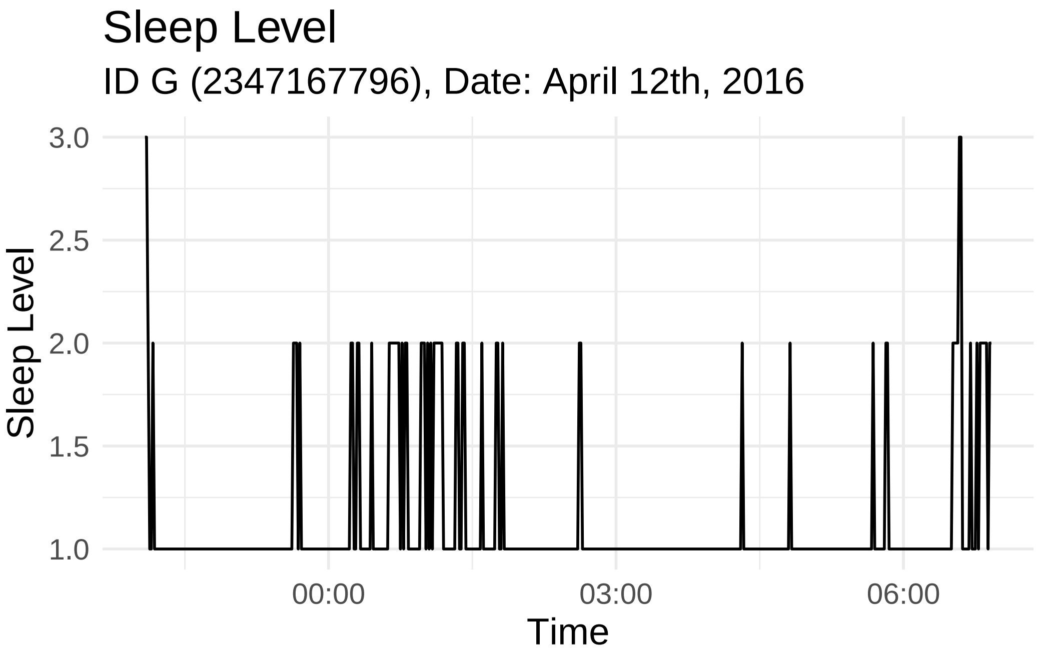
5.1. Findings

As shown in the previous section of analysis, the first half of the analysis is done to monitor the vitals of Id G. The second half of the analysis id done to explore the routines of the top tier of calorie burners.

5.1.1. Monitoring Vitals

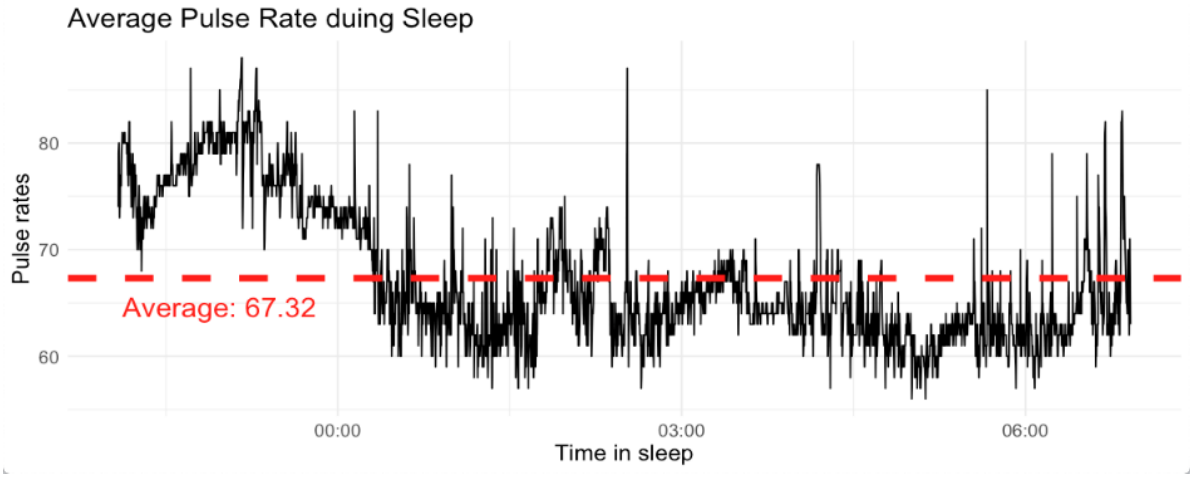
5.1.1.1. Sleep Levels of Id G

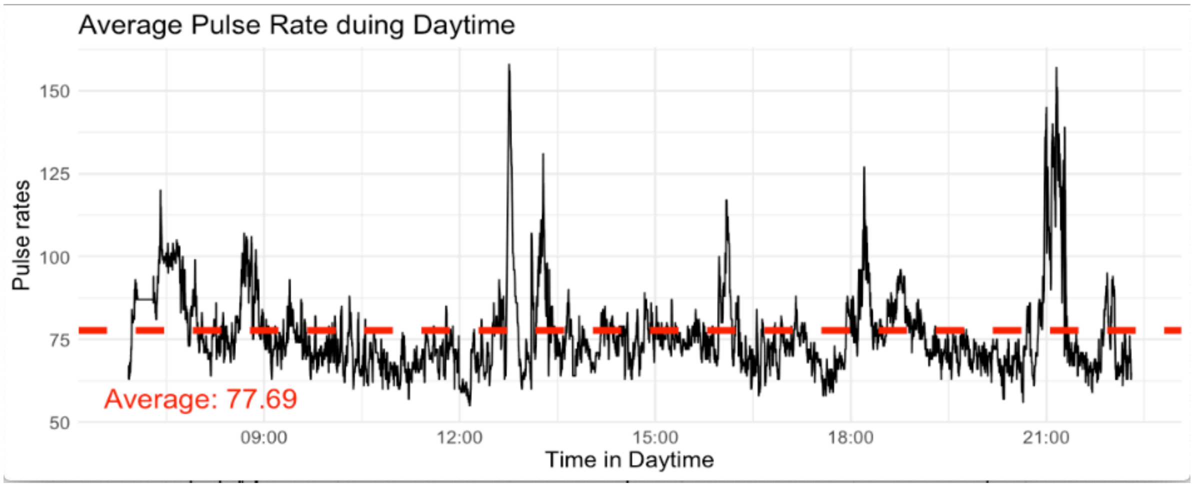
Three stages of sleep level are monitored. Within the scope of this analysis, three stages of sleep levels are meticulously monitored. By closely scrutinizing these three distinct sleep stages, the aim is to gain deeper insights into the participants' sleep quality and overall well-being.



5.1.1.2. Pulse Rate of Id G

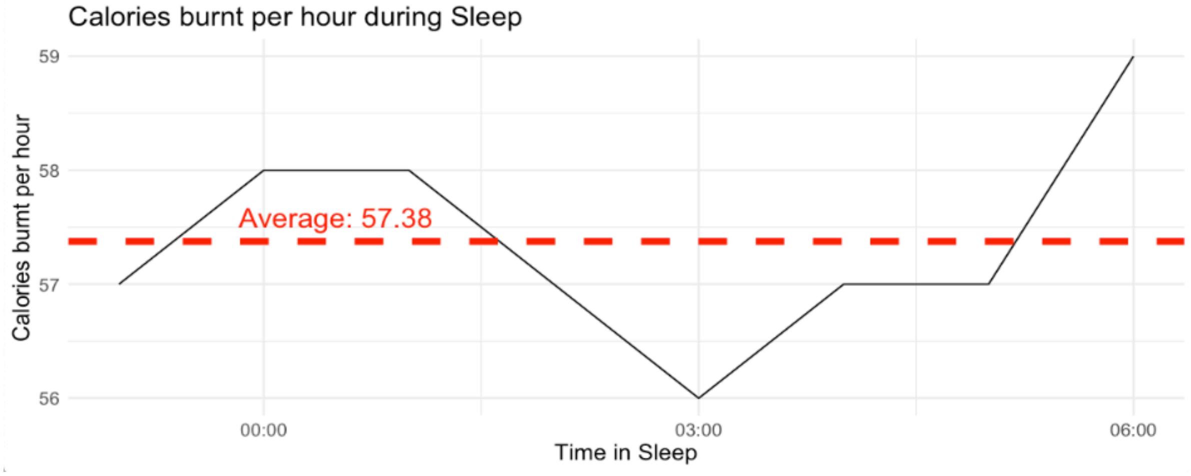
For a 24-hour period, Id G's pulse rate is monitored. The average pulse rate during sleep hours is 67, contrasting with an average pulse rate of 78 during waking hours. The observed 14% drop in pulse rate during sleep is a well-known physiological phenomenon, indicating that Id G's pulse rate is within the normal range.

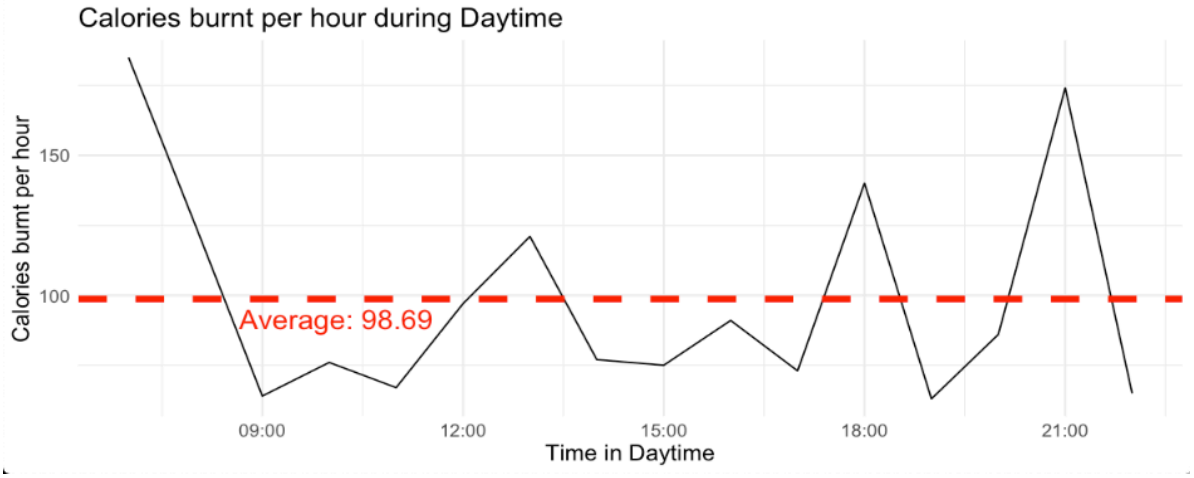




5.1.1.3. Calorie Consumption/Burn by Id G

The calorie consumption of Id G is monitored over a 24-hour period. The average calorie consumption per hour during sleep is 57, contrasting with an average of 99 calories per hour during waking hours. This observed 42% drop in calorie burn during sleep is a well-known physiological phenomenon, indicating that Id G's metabolic activity is within the normal range for a healthy individual.



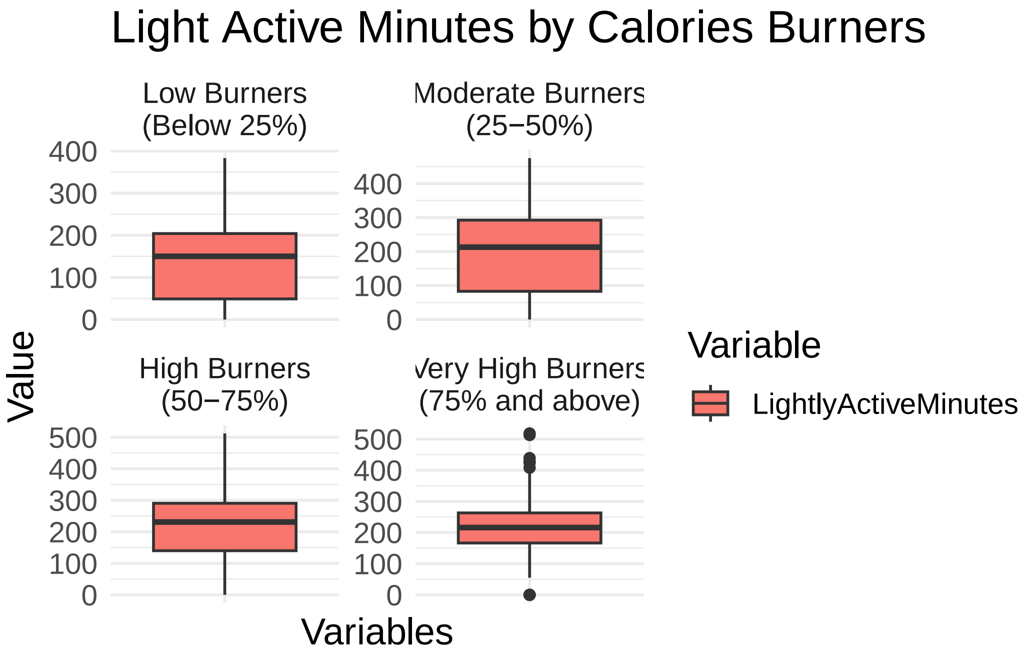


5.1.2. Daily Activities

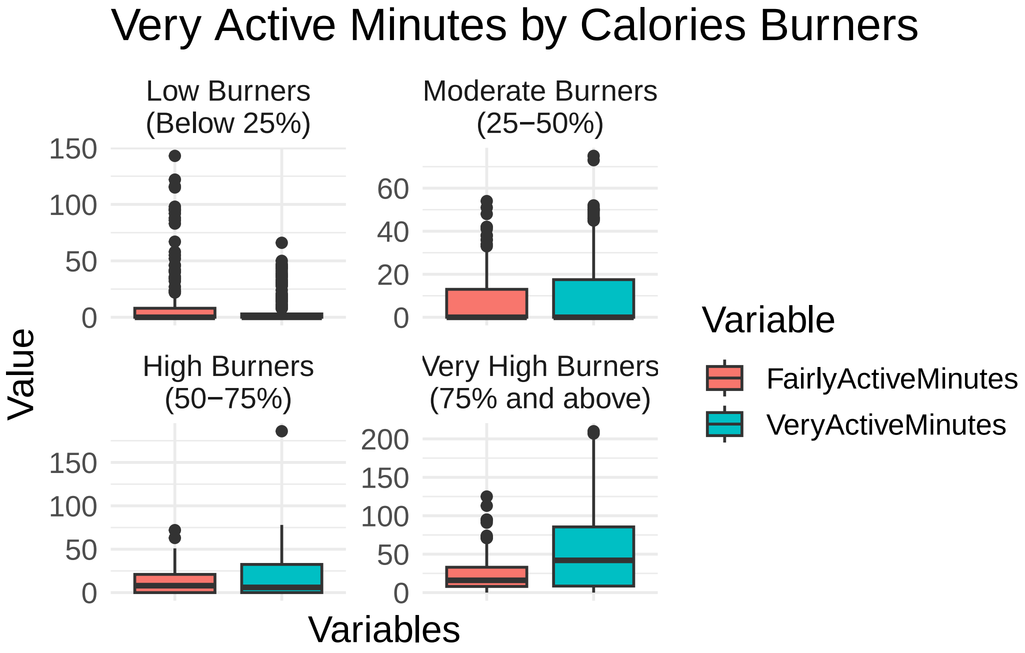
Thirty-three participants' daily activities are examined based on their calorie consumption/burn levels. The groups are categorized as follows:

- Top Tier: This group comprises individuals with the highest calorie burn, known as Very High Burners (75% above).
- Second Tier: The group with the second-highest calorie burn, known as High Burners (50 - 75%)
- Third Tier: The group with the third-highest calorie burn, known as Moderate Burners (25 - 50%)
- Fourth Tier: The group with the fourth-highest calorie burn, known as Lower Burners (below 25%

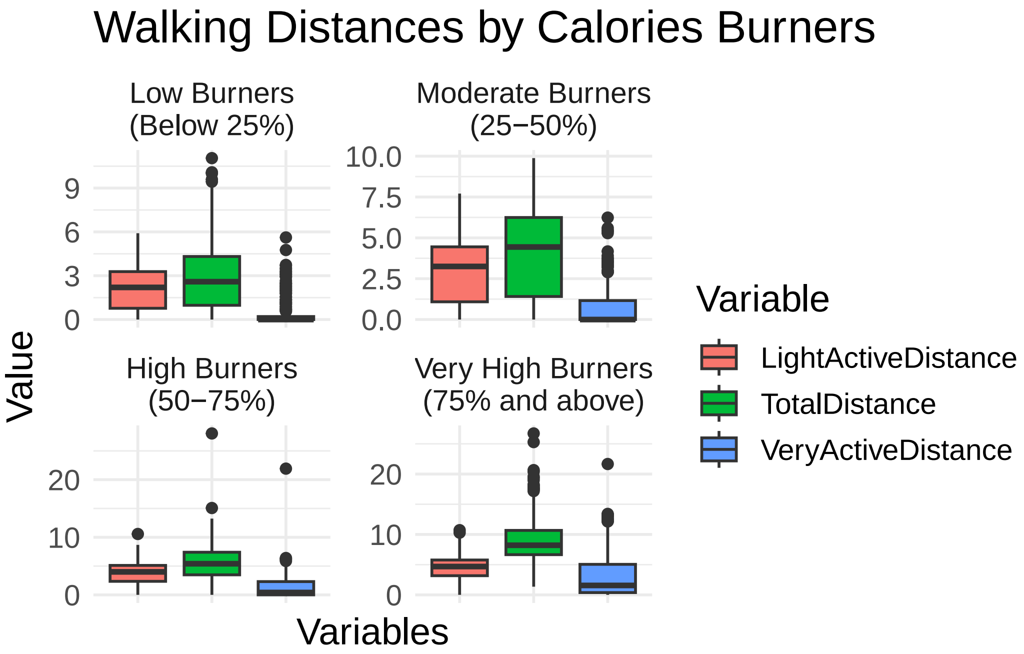
The majority of the top-tier group maintains approximately 200 minutes of lightly active minutes.



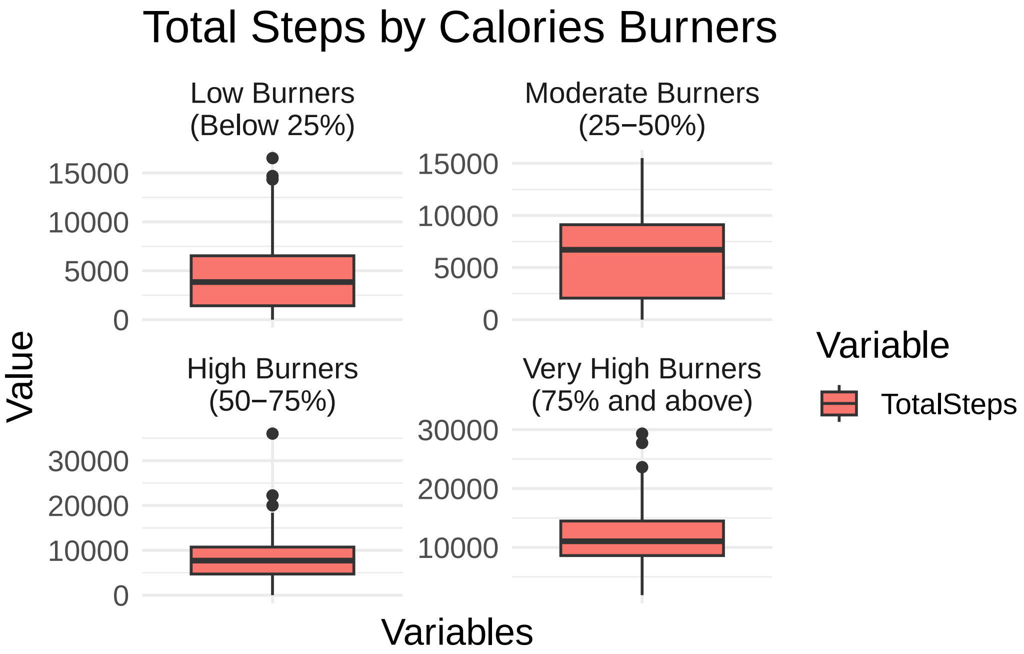
The top-tier group achieves around 50 minutes of very active minutes.



Additionally, this group demonstrates that their total distances covered in running/walking far surpass those of other tiers.

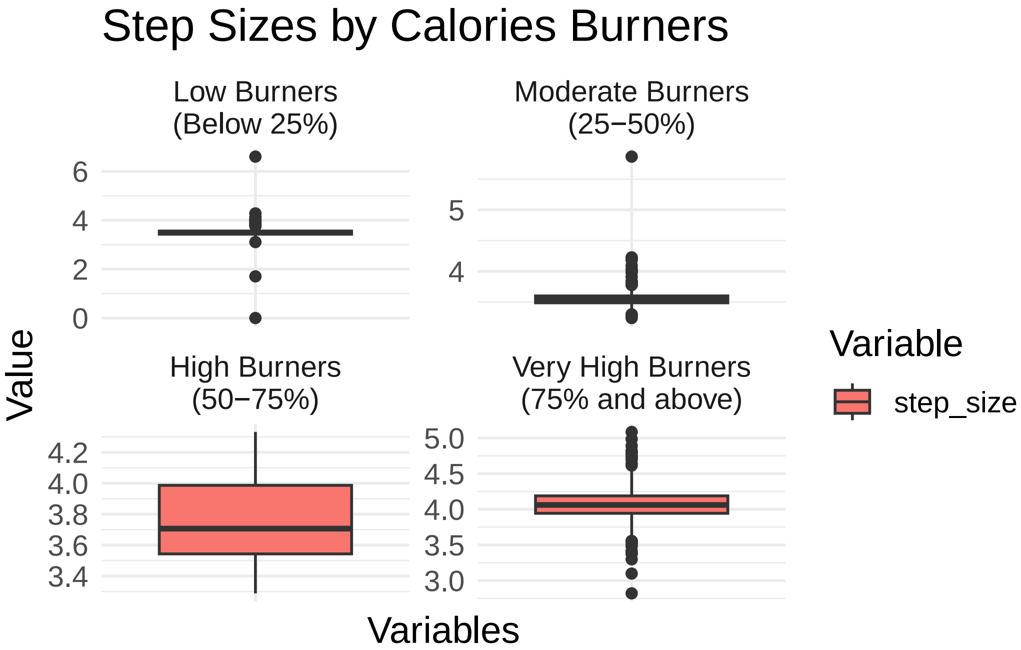


The top-tier group achieves more than 10,000 steps daily, aligning with the conventional wisdom that advocates for 10,000 steps a day as a benchmark for a healthy lifestyle.



Analyzing the data on each participant's total distances and total steps allows for the calculation of an average step size for all participants. Notably, the top-tier group exhibits an average step size significantly larger than the rest of the groups.

```
Warning: Removed 77 rows containing non-finite values (`stat_boxplot()`).
```



5.2. Summary

1. The comprehensive 24-hour monitoring of sleep levels, pulse rate, and calorie consumption/burn provides valuable insights into the participant's well-being. This information not only instills a sense of security regarding their physical conditions but also serves as a motivational tool, inspiring a desire to maintain overall health and fitness.
2. The effective daily activities that lead to increased calorie consumption/burn have been identified through the routines of the top-tier group. By consistently maintaining more active hours, covering longer distances in walk/run activities, and incorporating a wider step size, this group demonstrates successful strategies for burning more calories and promoting overall well-being.

6. Act

In the implementation phase, it is recommended that if **Bellabeat** decides to develop a device similar to the one analyzed, key features should include personalized notifications. These notifications can include real-time updates on the amount of calories burned so far and summaries of daily vital monitoring reports. Additionally, the device can provide tailored activity recommendations based on users' goals or preferences, enhancing user engagement and promoting a healthier lifestyle.

To maximize impact, the device should be strategically advertised within health-conscious communities initially, creating awareness and fostering a user base that aligns with the product's health and fitness focus. This targeted approach can help establish a strong foothold in the market and attract users who value advanced health tracking and personalized recommendations.

Conclusion

key takeaways of this analysis.

1. **Insightful Monitoring:** The 24-hour monitoring provides valuable insights into participants' sleep patterns, pulse rates, and calorie-related activities, offering a holistic view of their well-being.
2. **Sense of Security:** The data obtained from monitoring fosters a sense of security among participants regarding their physical conditions. Knowing that their vital signs and calorie-related metrics are being tracked can contribute to a heightened awareness of health.
3. **Motivation for Fitness:** The monitoring not only provides information but also acts as a motivational tool. Participants are inspired to maintain good physical conditions and are likely to be more motivated to engage in activities that contribute to their overall health and fitness.
4. **Effective Daily Activities:** The analysis of the top-tier group's routines reveals effective daily activities for burning more calories. This includes maintaining more active hours, covering longer distances in walk/run activities, and incorporating a wider step size.
5. **Recommendations for Device Features:** If the company develops a similar health-tracking device, the report suggests incorporating features such as personalized notifications for calories burned, real-time updates, and tailored activity recommendations based on users' goals or preferences.
6. **Strategic Advertising:** To maximize impact, the report recommends strategically advertising the device within health-conscious communities initially, tapping into a target audience that values advanced health tracking and personalized recommendations.

Limitations of this analysis: This represents my first of two capstone analyses using R. The dual challenge of learning R while conducting the analysis was demanding, yet ultimately rewarding. While there may be various ways to enhance this analysis, it concludes here for the purpose of the capstone.

Reference

[1]. Case Study 2: “How Can a Wellness Technology Company Play It Smart?” This capstone was a part of the google data analystic certificate program administered through Coursera.

[2]. CC0: Public Domain dataset made available through **Mobius**