Bellabeat Data Analysis Case Study

Aichu Tan

Introduction

Scenario

This data analytics report is the second case of the Google Data Analytics Capstone project. I am part of a team of junior data analysts working at Bellabeat, a high-tech company focused on women's health devices. These devices include fitness trackers, a wellness device tracker, a smart water bottle, and a membership program with personalized guidance on nutrition, activity, sleep, health, beauty, and mindfulness based on users' lifestyles and goals. My task on the team is to analyze smart device data to gain insights into how consumers are using their smart devices. The data will be used to guide the company's marketing strategy.

This case study follows the six-step data analysis process:

Ask -> Prepare -> Process -> Analyze -> Share -> Act

1. Ask

Stakeholders:

- Urška Sršen: Bellabeat's cofounder and Chief Creative Officer
- Sando Mur: Mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team
- Bellabeat marketing analytics team: responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy. I am a junior data analyst in the team.

Business questions:

- What are some trends in smart device usage?
- How could these trends apply to Bellabeat customers?
- How could these trends help influence Bellabeat marketing strategy?

2. Prepare

About Data The data for this case study comes from public dataset: FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius). These 33 Fitbit users consented to the submission of all personal tracker data contained in this dataset.

Limitation the Data:

- These data sample is the daily activity from 33 users of Fitbit user in duration from 4/12/2016 ~ 5/9/2016.

- These data cannot represent of all Fitbit users.
- The data is a little outdated since it was collected in 2016.

Installing and loading common packages and libraries To implement the data preparation, the following packages and tools are installed.

```
library(rmarkdown)
library(tidyverse)
library(lubridate)
library(here)
library(janitor)
library(skimr)
```

```
getwd()
```

Loading CSV files and previewing data

```
## [1] "/Users/aichutan/Desktop/Bellabeat Case Study/My first Case Study - Bellabeat"
```

```
setwd("/Users/aichutan/Desktop/Bellabeat Case Study/My first Case Study - Bellabeat")
daily_activity <- read.csv("dailyActivity_merged.csv")
daily_sleep <- read.csv("sleepDay_merged.csv")
hourly_step <- read.csv("hourlySteps_merged.csv")
```

```
head(daily_activity)
```

Exploring Key Tables in daily_activity Data

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366    4/12/2016      13162          8.50            8.50
## 2 1503960366    4/13/2016      10735          6.97            6.97
## 3 1503960366    4/14/2016      10460          6.74            6.74
## 4 1503960366    4/15/2016       9762          6.28            6.28
## 5 1503960366    4/16/2016      12669          8.16            8.16
## 6 1503960366    4/17/2016       9705          6.48            6.48
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               1.88                     0.55
## 2                        0               1.57                     0.69
## 3                        0               2.44                     0.40
## 4                        0               2.14                     1.26
## 5                        0               2.71                     0.41
## 6                        0               3.19                     0.78
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
```

```
## 3                3.91                       0                30
## 4                2.83                       0                29
## 5                5.04                       0                36
## 6                2.51                       0                38
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  13                  328              728     1985
## 2                  19                  217              776     1797
## 3                  11                  181             1218     1776
## 4                  34                  209              726     1745
## 5                  10                  221              773     1863
## 6                  20                  164              539     1728
```

```r
str(daily_activity) # Rows: 940, Column:16, ID converted to chr, ActivityDate converted to date format
```

```
## 'data.frame':    940 obs. of  15 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps              : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ TotalDistance           : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ Calories                : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```r
table(daily_activity$Id) # Some missing data
```

```
## 
## 1503960366 1624580081 1644430081 1844505072 1927972279 2022484408 2026352035 
##         31         31         30         31         31         31         31 
## 2320127002 2347167796 2873212765 3372868164 3977333714 4020332650 4057192912 
##         31         18         31         20         30         31          4 
## 4319703577 4388161847 4445114986 4558609924 4702921684 5553957443 5577150313 
##         31         31         31         31         31         31         30 
## 6117666160 6290855005 6775888955 6962181067 7007744171 7086361926 8053475328 
##         28         29         26         31         26         31         31 
## 8253242879 8378563200 8583815059 8792009665 8877689391 
##         19         31         31         29         31
```

```r
# Check how many users are in the data
n_distinct(daily_activity$Id) # 33
```

```
## [1] 33
```

```r
# Convert Id columns to character type
daily_activity$Id <- as.character(daily_activity$Id)

# Convert date columns to Date type, add Weekday column, convert active minutes to active hour to bette
daily_activity <- daily_activity %>%
  mutate(ActivityDate = as.Date(ActivityDate, format="%m/%d/%Y"),
         Weekday = weekdays(ActivityDate),
         VeryActiveHours = round(VeryActiveMinutes/60, 1),
         FairlyActiveHours = round(FairlyActiveMinutes/60, 1),
         LightlyActiveHours = round(LightlyActiveMinutes/60, 1),
         SedentaryHours = round(SedentaryMinutes/60, 1))

# Identify the date range of the daily_activity data
mindate <- min(daily_activity$ActivityDate)
maxdate <- max(daily_activity$ActivityDate) # The data range is from 2016-04-12 to 2016-05-12
```

```r
head(daily_sleep)
```

Exploring Key Tables in daily_sleep Data

```
##           Id              SleepDay TotalSleepRecords TotalMinutesAsleep
## 1 1503960366 4/12/2016 12:00:00 AM                 1                327
## 2 1503960366 4/13/2016 12:00:00 AM                 2                384
## 3 1503960366 4/15/2016 12:00:00 AM                 1                412
## 4 1503960366 4/16/2016 12:00:00 AM                 2                340
## 5 1503960366 4/17/2016 12:00:00 AM                 1                700
## 6 1503960366 4/19/2016 12:00:00 AM                 1                304
##   TotalTimeInBed
## 1            346
## 2            407
## 3            442
## 4            367
## 5            712
## 6            320
```

```r
str(daily_sleep) # Rows: 413, Columns: 5, ID converted to chr, convert SleepdDay to date format
```

```
## 'data.frame':    413 obs. of  5 variables:
##  $ Id                : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : chr  "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:00 AM"
##  $ TotalSleepRecords : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : int  346 407 442 367 712 320 377 364 384 449 ...
```

```r
table(daily_sleep$Id) # some missing data
```

```
##
## 1503960366 1644430081 1844505072 1927972279 2026352035 2320127002 2347167796
##         25          4          3          5         28          1         15
```

```
## 3977333714 4020332650 4319703577 4388161847 4445114986 4558609924 4702921684
##         28          8         26         24         28          5         28
## 5553957443 5577150313 6117666160 6775888955 6962181067 7007744171 7086361926
##         31         26         18          3         31          2         24
## 8053475328 8378563200 8792009665
##          3         32         15
```

```
# Check how many users are in the data
n_distinct(daily_sleep$Id) # 24
```

```
## [1] 24
```

```
# Convert Id columns to character type
daily_sleep$Id <- as.character(daily_sleep$Id)

# Convert date columns to Date type, mdy, change column name consistent with others, and convert total 
daily_sleep <- daily_sleep %>%
  mutate(SleepDay = as.Date(SleepDay, format="%m/%d/%Y"),
         TotalHoursAsleep = round(TotalMinutesAsleep/60, 1),
         TotalHoursInBed = round(TotalTimeInBed/60, 1)) %>%
  rename(ActivityDate = SleepDay)
```

```
head(hourly_step)
```

Exploring Key Tables in hourly_step Data

```
##           Id          ActivityHour StepTotal
## 1 1503960366 4/12/2016 12:00:00 AM       373
## 2 1503960366  4/12/2016 1:00:00 AM       160
## 3 1503960366  4/12/2016 2:00:00 AM       151
## 4 1503960366  4/12/2016 3:00:00 AM         0
## 5 1503960366  4/12/2016 4:00:00 AM         0
## 6 1503960366  4/12/2016 5:00:00 AM         0
```

```
str(hourly_step) # Rows: 22099, Columns: 3, ID converted to chr, ActivityHour converted to date format,
```

```
## 'data.frame':    22099 obs. of  3 variables:
##  $ Id          : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour: chr  "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00 AM" "4/12/20
##  $ StepTotal   : int  373 160 151 0 0 0 0 0 250 1864 ...
```

```
table(hourly_step$Id) # some missing data
```

```
## 
## 1503960366 1624580081 1644430081 1844505072 1927972279 2022484408 2026352035
##        717        736        708        731        736        736        736
## 2320127002 2347167796 2873212765 3372868164 3977333714 4020332650 4057192912
##        735        414        736        472        696        732         88
```

```
## 4319703577 4388161847 4445114986 4558609924 4702921684 5553957443 5577150313
##        724        735        735        736        731        730        708
## 6117666160 6290855005 6775888955 6962181067 7007744171 7086361926 8053475328
##        660        665        610        732        601        733        735
## 8253242879 8378563200 8583815059 8792009665 8877689391
##        431        735        718        672        735
```

```r
# Check how many users are in the data
n_distinct(hourly_step$Id) # 33
```

```
## [1] 33
```

```r
# Convert Id columns to character type
hourly_step$Id <- as.character(hourly_step$Id)

# Convert date columns to Date type and extract time
hourly_step <- hourly_step %>%
  mutate(ActivityHour = as.POSIXct(ActivityHour, format="%m/%d/%Y %I:%M:%S %p"),
         ActivitybyDate = as.Date(ActivityHour),
         ActivitybyTime = format(ActivityHour, "%H:%M"))
```

After reviewing the data, in the preparation step, I made the following modifications:

1. Converted daily_activity/Id, daily_sleep/Id, hourly_calories/Id, and hourly_step/Id to consistent character format.
2. Changed daily_activity/ActivityDate and daily_sleep/SleepDay to date format. Formatted and extracted activity time from hourly_calories/ActivityHour and hourly_step/ActivityHour into new columns ActivitybyDate and ActivitybyTime.
3. Add a new column weekday.
4. Identified 33 active users in daily_activity, hourly_calories, and hourly_step, and 24 active users in daily_sleep from the table data.
5. Noted that there are missing data points across the datasets.
6. Rename daily_sleep/SleepDay to daily_sleep/ActivityDate for consistent naming in preparing for future merging.

3. Process

For processing the data for analysis, the following tasks will be implemented: checking and cleaning for any duplicated data.

```r
sum(duplicated(daily_activity)) #0
```

Checking for any duplicated data.

```
## [1] 0
```

```r
sum(duplicated(daily_sleep))  #3
```

```
## [1] 3
```

```r
sum(duplicated(hourly_step)) #0
```

```
## [1] 0
```

```r
#daily_activity
daily_activity <- daily_activity %>%
  distinct() %>%
  drop_na()

#daily_sleep
daily_sleep <- daily_sleep %>%
  distinct() %>%
  drop_na()

#hourly_step
hourly_step <- hourly_step %>%
  distinct() %>%
  drop_na()
```

Removing duplicated data

Confirming there are no duplicated data :

```
## [1] 0
```

```
## [1] 0
```

```
## [1] 0
```

4. Analyze & Share

```r
# Merge two data sets of "daily_activity" and "daily_sleep", merged data is named as "activity_sleep"
activity_sleep <- merge(daily_activity, daily_sleep, by= c("Id", "ActivityDate"))

# Confirm any duplicated data.
sum(duplicated(activity_sleep)) # 0
```

Merge data sets

```
## [1] 0
```

Finding the Statistical Data in The activity_sleep Dataset

```
##      Weekday     TotalSteps    TotalDistance    VeryActiveHours
##  Monday   :46   Min.   :   17   Min.   : 0.010   Min.   :0.000
##  Tuesday  :65   1st Qu.: 5189   1st Qu.: 3.592   1st Qu.:0.000
##  Wednesday:66   Median : 8913   Median : 6.270   Median :0.100
##  Thursday :64   Mean   : 8515   Mean   : 6.012   Mean   :0.412
##  Friday   :57   3rd Qu.:11370   3rd Qu.: 8.005   3rd Qu.:0.600
##  Saturday :57   Max.   :22770   Max.   :17.540   Max.   :3.500
##  Sunday   :55
##  FairlyActiveHours LightlyActiveHours SedentaryHours     Calories
##  Min.   :0.0000    Min.   :0.00       Min.   : 0.00   Min.   : 257
##  1st Qu.:0.0000    1st Qu.:2.60       1st Qu.:10.50   1st Qu.:1841
##  Median :0.2000    Median :3.50       Median :11.90   Median :2207
##  Mean   :0.2949    Mean   :3.61       Mean   :11.87   Mean   :2389
##  3rd Qu.:0.4000    3rd Qu.:4.40       3rd Qu.:13.07   3rd Qu.:2920
##  Max.   :2.4000    Max.   :8.60       Max.   :21.10   Max.   :4900
##
##  TotalHoursAsleep TotalHoursInBed
##  Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 6.000   1st Qu.: 6.725
##  Median : 7.200   Median : 7.700
##  Mean   : 6.987   Mean   : 7.639
##  3rd Qu.: 8.200   3rd Qu.: 8.800
##  Max.   :13.300   Max.   :16.000
##
```

```
## [1] 24
```

Finding the Statistical Data in The daily_activity Dataset

```
##      Weekday      TotalSteps    TotalDistance    VeryActiveMinutes
##  Monday   :120   Min.   :    0   Min.   : 0.000   Min.   :  0.00
##  Tuesday  :152   1st Qu.: 3790   1st Qu.: 2.620   1st Qu.:  0.00
##  Wednesday:150   Median : 7406   Median : 5.245   Median :  4.00
##  Thursday :147   Mean   : 7638   Mean   : 5.490   Mean   : 21.16
##  Friday   :126   3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.: 32.00
##  Saturday :124   Max.   :36019   Max.   :28.030   Max.   :210.00
##  Sunday   :121
##  FairlyActiveMinutes LightlyActiveMinutes SedentaryHours     Calories
##  Min.   :  0.00      Min.   :  0.0        Min.   : 0.00   Min.   :   0
##  1st Qu.:  0.00      1st Qu.:127.0        1st Qu.:12.20   1st Qu.:1828
##  Median :  6.00      Median :199.0        Median :17.60   Median :2134
##  Mean   : 13.56      Mean   :192.8        Mean   :16.52   Mean   :2304
##  3rd Qu.: 19.00      3rd Qu.:264.0        3rd Qu.:20.50   3rd Qu.:2793
##  Max.   :143.00      Max.   :518.0        Max.   :24.00   Max.   :4900
##
```

```
## [1] 33
```

From the summary data, we can determine that in the daily_activity table, the daily average total steps for 33 users is 7,638, and the daily calories burned is 2,308. We can also determine that the average very active hours, fairly active hours, and lightly active hours are 21.1, 13.5 and 192.8 minutes, respectively, while sedentary hours average 16.52 hours/day. This means that users spend most of the day being inactive.

From the activity_sleep table, the daily average total hours asleep for 24 users is 6.9 hours per day. Additionally, the average total hours in bed is 0.652 hours longer than asleep.

For further analysis, we can visualize the relationships between the average total steps, average calories burned, and average sleep hours over a week.

```r
# Finding the Daily Average data of Fitbit Users
# Verify data transfromation steps

daily_summary <- daily_activity %>%
  group_by(Weekday) %>%
  summarise(AvgTotalSteps = mean(TotalSteps), AvgTotalCalories = mean(Calories))
summary(daily_summary)
```

Finding the Daily Average Total Step

```
##       Weekday  AvgTotalSteps  AvgTotalCalories
##  Monday   :1   Min.   :6933   Min.   :2200
##  Tuesday  :1   1st Qu.:7427   1st Qu.:2283
##  Wednesday:1   Median :7559   Median :2324
##  Thursday :1   Mean   :7629   Mean   :2305
##  Friday   :1   3rd Qu.:7953   3rd Qu.:2343
##  Saturday :1   Max.   :8153   Max.   :2356
##  Sunday   :1
```

```r
ggplot(daily_summary, aes(x = Weekday, y = AvgTotalSteps)) +
  geom_bar(stat = "identity", fill = "lightgreen") +
  labs(title = "Daily Average Steps",
       x = "Weekday",
       y = "Average Total Steps",
       caption = "Figure 1: Data collected by FITBit Fitness Tracker Data ") +
  theme_minimal() +
  scale_y_continuous(limits = c(0, 10000), breaks = seq(1000, 10000, by = 1000))
```

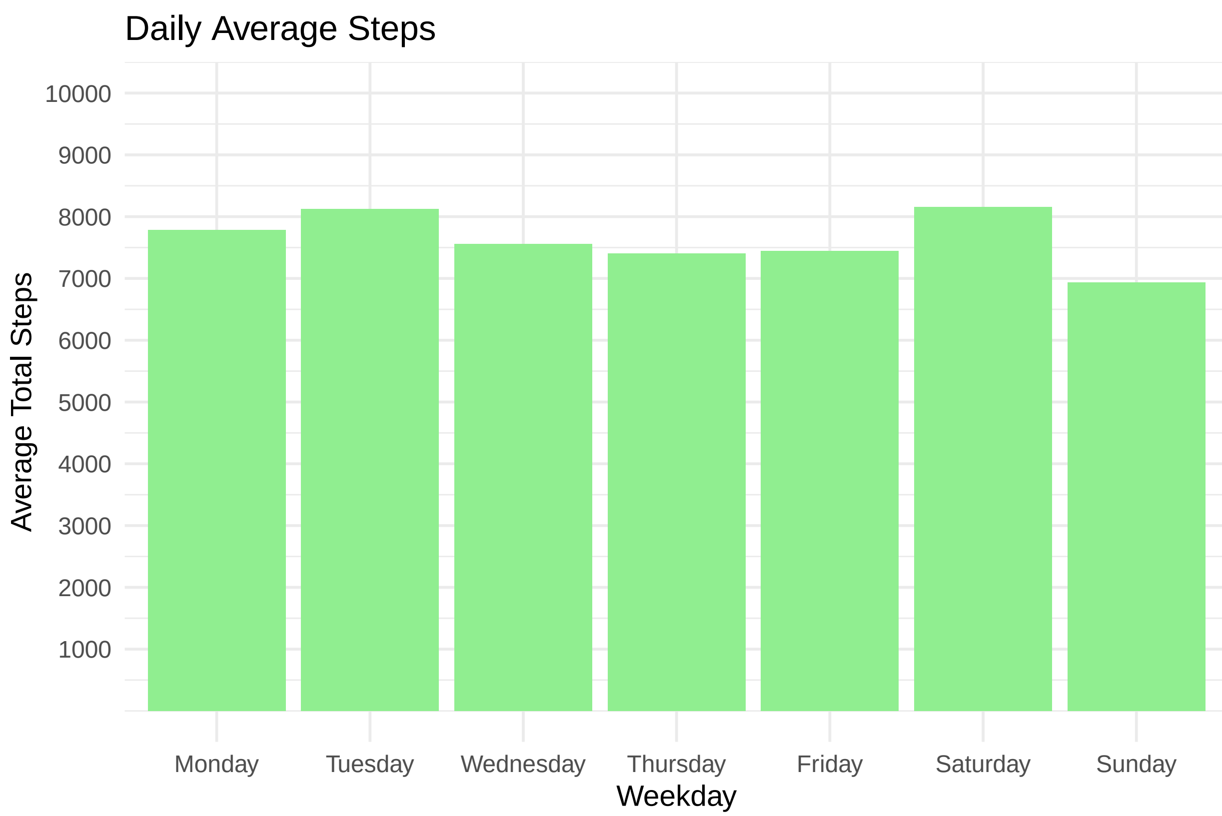

Figure 1: Data collected by FITBit Fitness Tracker Data

Figure 1 and the table show the average number of steps taken per weekday, derived from data collected by a FITBit Fitness Tracker from 33 users. Based on this dataset, we observe that the highest total number of steps occurs on Tuesday and Saturday, with the lowest total steps recorded on Sunday. The calculated weekly average total steps is 7,629, reflecting overall activity levels throughout the week.

According CDC and MedicineNet, physical activity level can be categorized as follows:

- Sedentary: less than 5000 steps per day
- Low Active: 5000 to 7499 steps per day
- Somewhat Active: 7500 to 9999 steps per day
- Active: 10000 to 12,499 steps per day
- Highly Active: 12,500 or more steps per day

Referring back to the table data, the total steps of the Fitbit users average 7,629, categorizing them as **somewhat active users**. Studies suggest maintaining an activity level between 7,500 to 10,000 steps per day to improve health, including better blood sugar levels, lower blood pressure, and alleviating symptoms of depression and anxiety.

```r
# Finding the Daily Average Calories of Fitbit Users
# Create a bar chart showing the Daily Average Calories of 33 Fitbit Users
ggplot(daily_summary, aes(x = Weekday, y = AvgTotalCalories)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Daily Average Calories",
       x = "Weekday",
```

```r
      y = "Average Calories",
      caption = "Figure 2: Data collected by FITBit Fitness Tracker Data") +
  theme_minimal()
```

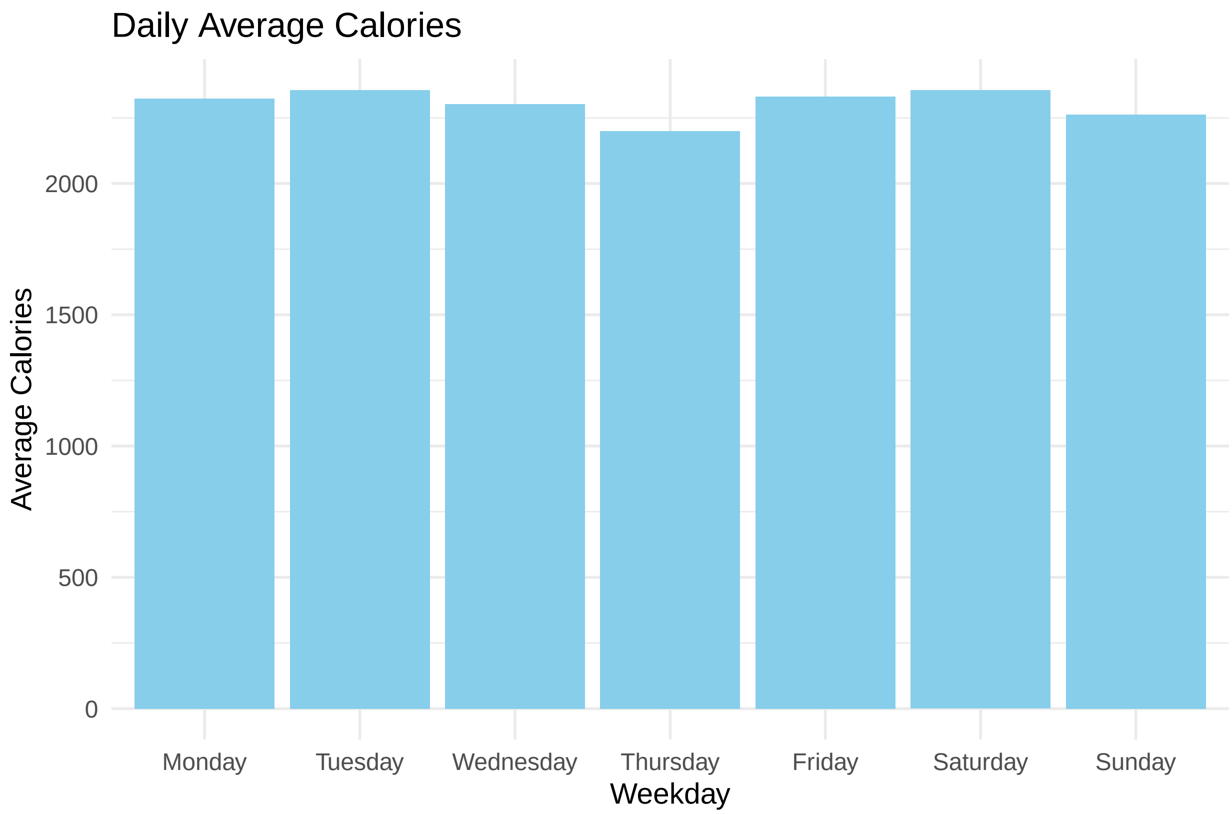

Figure 2: Data collected by FITBit Fitness Tracker Data

Figure 2 and the table show the average calorie for each weekday, derived from data collected by a FITBit Fitness Tracker from 33 users. Based on this dataset, we observe that the daily average calorie consumption remains stable around 2300, with a minimum of 2200 and a maximum of 2356 throughout the week.

```r
# Finding the Daily Sleep Hours of Fitbit Users
# Create a bar chart showing the Daily Sleep Hours of 24 Fitbit Users

dailysleep_summary <- activity_sleep %>%
  group_by(Weekday) %>%
  summarise(AvgTotalSteps = mean(TotalSteps), AvgTotalCalories = mean(Calories),AvgTotalAsleep = mean(To
summary(dailysleep_summary)
```

Finding the Daily Sleep Hours

```
##       Weekday  AvgTotalSteps  AvgTotalCalories AvgTotalAsleep
##  Monday   :1   Min.   :7298   Min.   :2277     Min.   :6.691
##  Tuesday  :1   1st Qu.:7962   1st Qu.:2318     1st Qu.:6.755
##  Wednesday:1   Median :8184   Median :2378     Median :6.988
```

```
##   Thursday :1    Mean   :8533    Mean    :2389     Mean   :6.994
##   Friday   :1    3rd Qu.:9228    3rd Qu.:2464     3rd Qu.:7.114
##   Saturday :1    Max.   :9871    Max.    :2507     Max.   :7.544
##   Sunday   :1
```

```r
ggplot(dailysleep_summary, aes(x = Weekday, y = AvgTotalAsleep)) +
  geom_bar(stat = "identity", fill = "purple") +
  labs(title = "Average Daily Sleep Hours",
       x = "Weekday",
       y = "Average Asleep Hours",
       caption = "Figure 3: Data collected by FITBit Fitness Tracker Data") +
  theme_minimal()
```

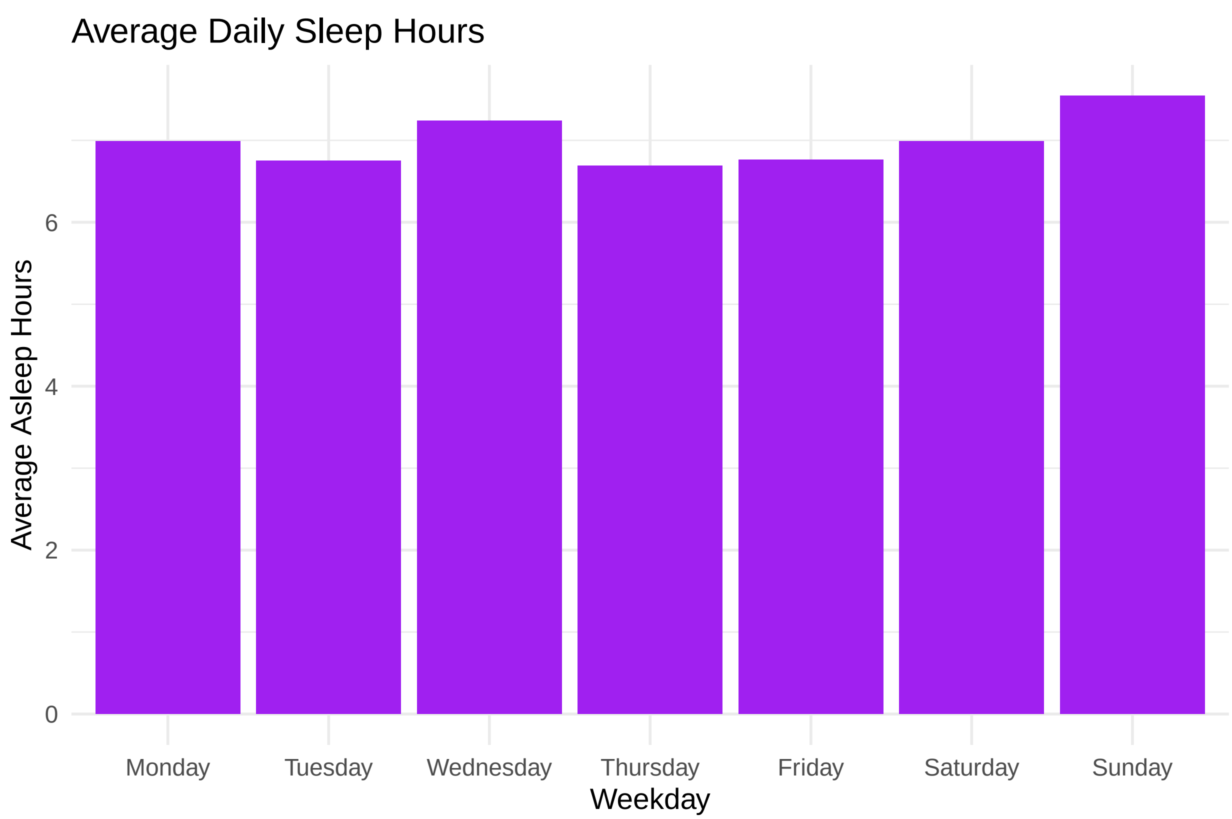

Figure 3: Data collected by FITBit Fitness Tracker Data

Figure 3 and the table show the average hours of sleep across different weekdays, derived from data collected by a FITBit Fitness Tracker from 24 users. Based on this dataset, we observe that the daily average sleep hours is 6.9 hours with longest sleep hour on Sunday, which is around 7.5 hours asleep and higher sleep hours on Wednesday.

According CDC, it is recommended that adult has 7 sleep hours and more which essential for our health and emotional well-being. Through app, we can remind users to extend their sleep duration to 7 hours or above.

It is also reflected in previous Figure 1 that the daily average steps are higher on Tuesday and Saturday. Additionally, in Figure 3, the daily sleep hours are longer on Wednesday and Sunday. This suggests that after days with higher step counts, users tend to have longer sleep durations. We can encourage users to walk more to achieve longer sleep durations to meet the standard health recommendation by the CDC.

####Creating Percentage of the Daily Active Level

Based on the data feature on the Fitbit, users can be categorized into four active levels: very active, fairly active, lightly active, and sedentary.

```r
# Daily Active Level
# Calculate total daily activity in minutes

total_sedentary <- sum(daily_activity$SedentaryMinutes)
total_lightly <- sum(daily_activity$LightlyActiveMinutes)
total_fairly <- sum(daily_activity$FairlyActiveMinutes)
total_veryactive <- sum(daily_activity$VeryActiveMinutes)

# Calculate total minutes (sum of all activity categories)
total_minutes <- total_sedentary + total_lightly + total_fairly + total_veryactive

# Calculate percentages
sedentary_percentage <- (total_sedentary / total_minutes) * 100
lightly_percentage <- (total_lightly / total_minutes) * 100
fairly_percentage <- (total_fairly / total_minutes) * 100
active_percentage <- (total_veryactive / total_minutes) * 100

# Data frame with percentages
percentage <- data.frame(
  level = c("Sedentary", "Lightly Active", "Fairly Active", "Very Active"),
  minutes = c(sedentary_percentage, lightly_percentage, fairly_percentage, active_percentage)
)

# Create a custom label function for the legend
percentage$label <- paste(percentage$level, sprintf("(%.1f%%)", percentage$minutes))

# Create the pie chart using ggplot
ggplot(percentage, aes(x = "", y = minutes, fill = label)) +
  geom_bar(stat = "identity", width = 1, color = "white") +
  coord_polar("y", start = 0) +
  labs(title = "Daily Active Level", fill = "Activity Level", y = "Percentage (%)", caption = "Figure 4
  theme(legend.position = "right")
```

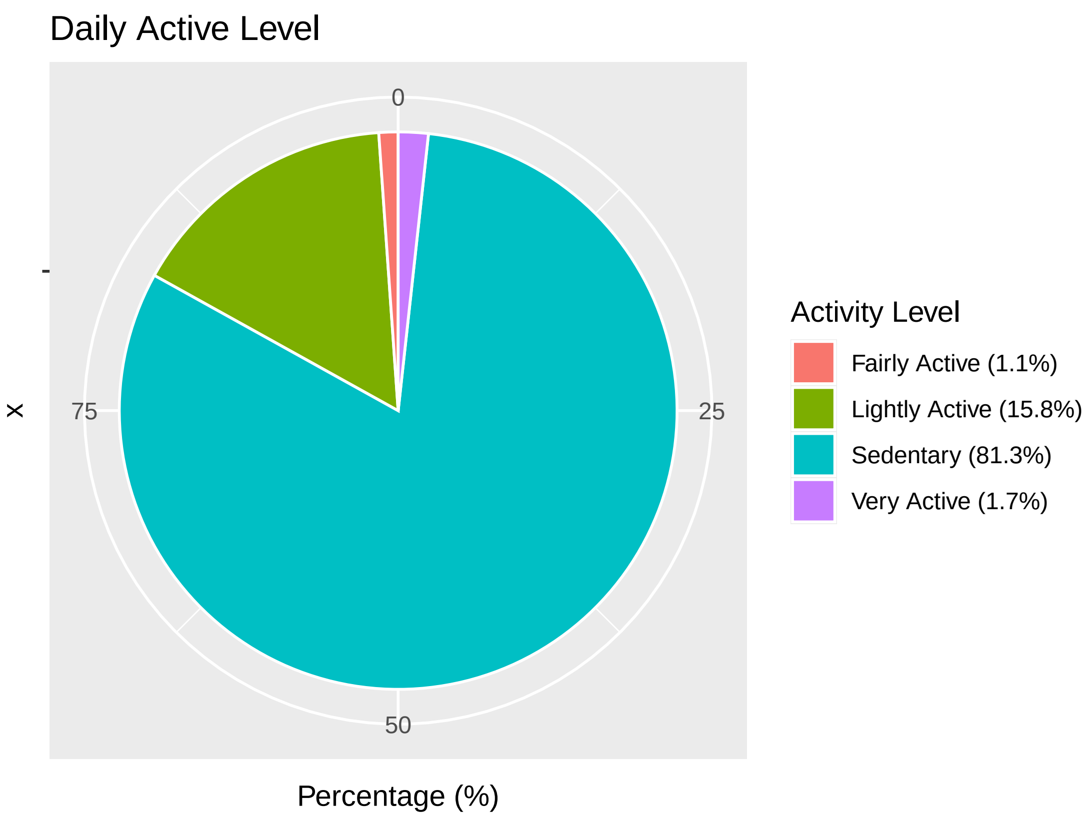

Figure 4: Data collected by FITBit Fitness Tracker Data

Figure 4 shows the percentage of the daily active levels of Fitbit users. The activite levels are categorized as very active, fairly active, lightly active, and sedentary. From the figure, we can determine that 81.3% of users are at a sedentary level and 1.7% at a very active level.

However, based on the calculation of daily total steps from Fitbit user data and the CDC's definition of activity levels, persons with fewer than 5000 steps per day are defined as sedentary, persons with 5000 to 7499 steps per day are defined as low active, persons with 7500 to 9999 steps per day are defined as somewhat active, persons with 10000 to 12499 steps per day are defined as active, and those with 12500 steps or more per day are defined as highly active. From Figure 5, we can determine that the users are mostly in the somewhat active category (27.3%) and the low active category (27.3%). Additionally, 6.1% are highly active, while only 24.2% are sedentary.

According to the CDC, daily calorie burn depends on sex, age, and activity level. For sedentary users, the number of calories burned may range from 1600 to 2000 per day and 1800 to 2400 calories per day for moderate active person. However, Figure 4 calculated the chart from the daily_activity dataset column VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, and SedentaryMinutes. The graph indicates that most of the users are at the sedentary level. This suggests that the definition from the smart device may need adjustment. For moderately active users, including somewhat active and low active individuals, calorie burn ranges between 2000 and 2800. Referring to the daily calorie burn of users, which falls within the range of 2200 to 2356, users are within the normal range.

```r
# Verify data transfromation steps
daily_average <- daily_activity %>%
  group_by(Id) %>%
  summarise(mean_daily_steps = mean(TotalSteps))

# Define
```

```r
user_type <- daily_average %>%
  mutate(user_type = case_when(
    mean_daily_steps < 5000 ~ "sedentary",
    mean_daily_steps >= 5000 & mean_daily_steps < 7500 ~ "low active",
    mean_daily_steps >= 7500 & mean_daily_steps < 10000 ~ "somewhat active",
    mean_daily_steps >= 10000 & mean_daily_steps < 12500 ~ "active",
    mean_daily_steps >= 12500 ~ "highly active"
  ))

sum(is.na(user_type)) #0
```

```
## [1] 0
```

```r
user_type_ratio <- user_type %>%
  group_by(user_type) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(user_type) %>%
  summarise(total_ratio = total / totals) %>%
  mutate(labels = scales::percent(total_ratio))

user_type_ratio$user_type <- factor(user_type_ratio$user_type , levels = c("highly active", "active", "s

user_type_ratio %>%
  ggplot(aes(x="",y=total_ratio, fill=user_type)) +
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0)+
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5, size=14, face = "bold")) +
  scale_fill_manual(values = c("#66CDAA","#8EEBEC", "#ffd480","#ffa07a", "#e55451")) +
  geom_text(aes(label = labels),
            position = position_stack(vjust = 0.5, reverse = FALSE))+
  labs(title="The Distribution of users", caption = "Figure 5: Data collected by FITBit Fitness Tracker
```

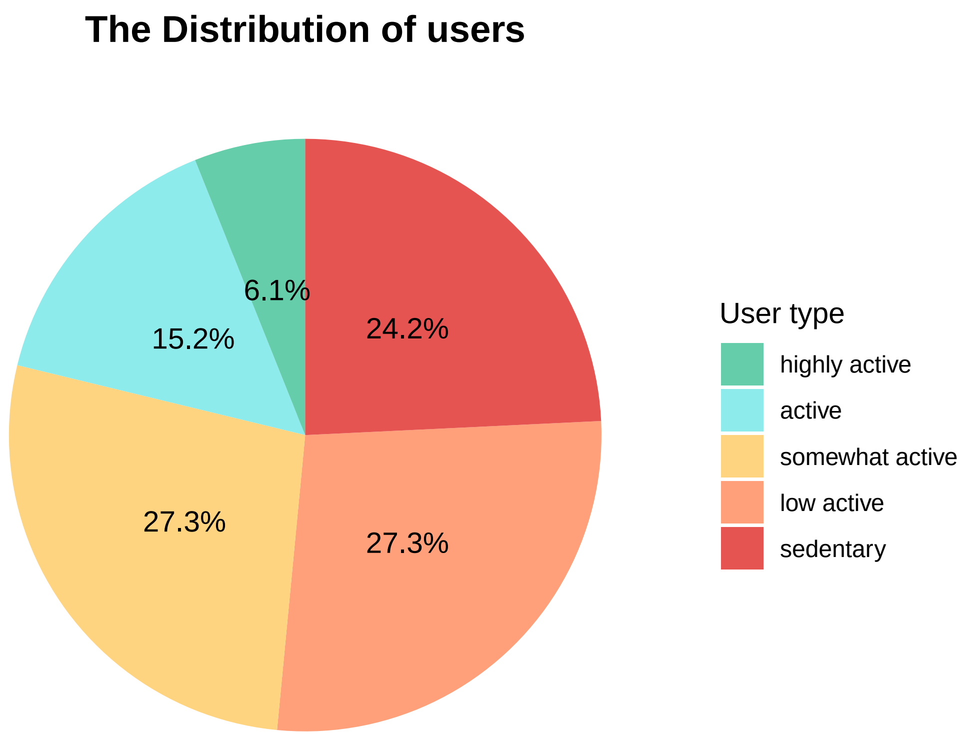

Figure 5: Data collected by FITBit Fitness Tracker Data

```
# Finding the Hourly Total Step
# Create the bar chart the Hourly Total Steps of Fitbit Users

ggplot(data = hourly_step, aes(x = ActivitybyTime, y = StepTotal, fill = ActivitybyTime)) +
  geom_bar(stat = "identity") +
  labs(title = "Hourly Total Steps",
       x = "Time",
       y = "Total Steps",
       caption = "Figure 6: Data collected by FITBit Fitness Tracker Data") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

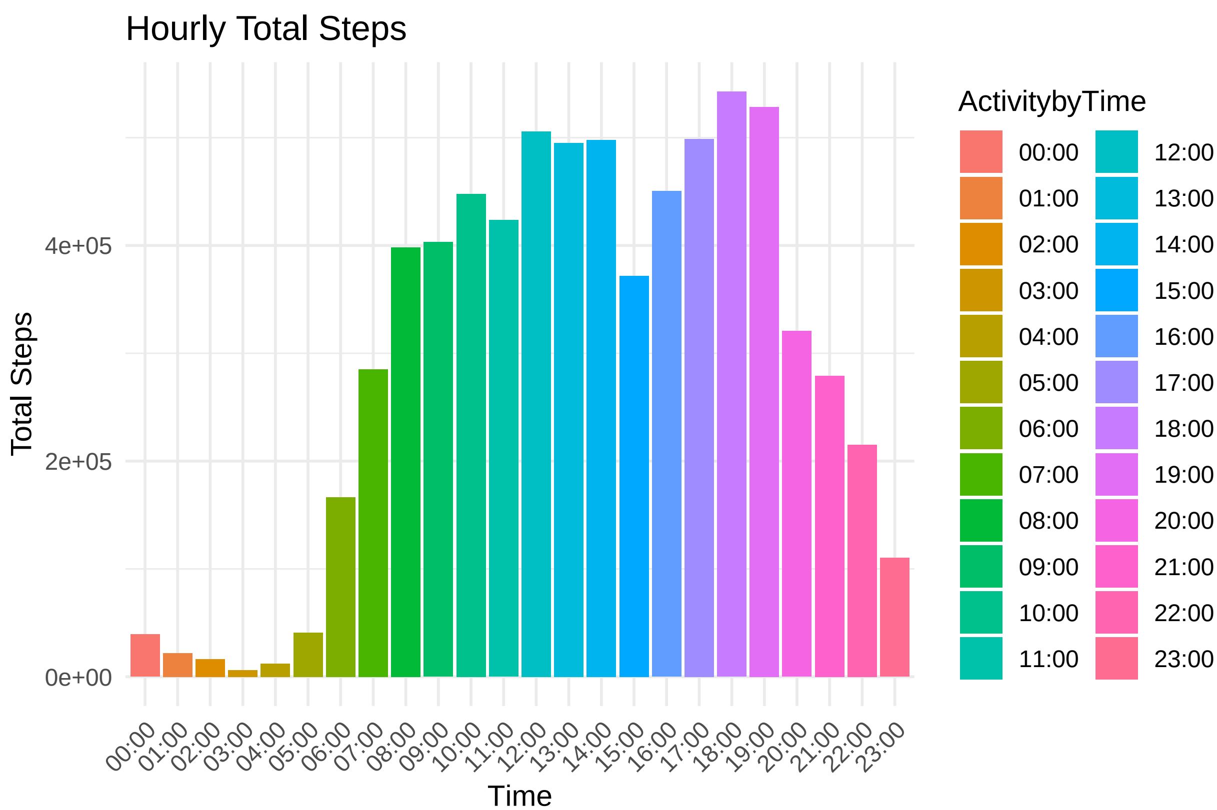

Figure 6: Data collected by FITBit Fitness Tracker Data

This chart shows the distribution of total steps taken throughout the day. It reveals that activity begins as early as 5 am, with a notable increase around 8 am. Activity remains consistently high through the morning and peaks aroun noon, gradually declining in the afternoon with a notable dip around 3 PM. The most active period appears to be at 18:00 and the total steps remain low until 23:00.

5. Act

The summary of the statistical analysis and visualization data

- Based on the calculation of daily total step data, we can observe that the daily average total steps is 7629 steps, which based on the CDC define as somewhat active users and to keep and improve health, its better to maintain an activity level between 7500 to 10,000 steps per day.
- Based on the daily average steps per weekday chart (Fig 1), we can determine that the highest total number of steps occurs on Tuesday and Saturday. This is reflected in the average daily sleep hours per weekday chart (Fig 3), which shows that users have the longest sleep hours on Wednesday and Sunday. This suggests that longer walking hours contribute to better sleep.
- From the sleep hour data, users have an average sleep duration of 6.9 hours, which is lower than the 7 hours or more recommended by the CDC. Moreover, users spend an average of 40 minutes (0.65 hours) in bed longer than they are asleep.
- From the daily active level, we found a discrepancy between the device's calculation of users' active levels. Based on the calculation of active levels from the VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, and SedentaryMinutes from the daily_activity dataset, 81.3% of users are defined as sedentary. However, based on the calculation of daily total steps and the CDC's definition of

activity levels, 27.3% of users are defined as somewhat active and low active. According to the latter calculation, users are defined as moderately active (somewhat and low active).

- From hourly step chart(Figure 6), we can observe that the most steps from 6-7 pm

Recommendations

How can these trends help influence Bellabeat's marketing strategy?

We can make marketing recommendations based on what we have learned from the daily activity of customers using smart fitness devices:

- Send notifications to users to maintain their daily walking activity between 7,500 to 10,000 steps.
- Educate users on the importance of getting at least 7 hours of sleep and inform them that one method to help achieve this is by increasing their daily steps to over 8,000.
- Match the daily active levels based on the CDC definition. Communicate with other departments if needed to update the software of the device, which categorizes as follows: Sedentary: less than 5000 steps per day, Low Active: 5000 to 7499 steps per day, Somewhat Active: 7500 to 9999 steps per day, Active: 10000 to 12,499 steps per day, Highly Active: 12,500 or more steps per day

Reference :

https://www.cdc.gov/physical-activity-basics/about/index.html. March 26, 2024.

Uttekar, Pallavi Suyog, MD. *MedicineNet.* https://www.medicinenet.com/how_many_steps_a_day_is_considered_active/article.htm