Capstone – Bellabeat Case Study

Meilin Zheng

July 01, 2022

1 Introduction

Bellabeat is a manufacturer of health-focused products for women. This project focuses on the “Leaf” product from Bellabeat, which is a classic wellness tracker that connects to the Bellabeat app to track activity, sleep, and stress.

2 Ask

2.1 Business Objective

The main objective is to Analyze smart device usage data, identify trends and gain insights on how people are using smart devices, then apply these insights into decision makings that can help improve Bellabeat’s marketing strategies on the Leaf product.

2.2 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

2.3 Some possible directions:

- What kind of users use the smart device more often?
- What are the main reasons that users use these smart devices? (for tracing their sleeping quality, record their calories burnout, etc)
- Any factors that are common among general users but do not apply to Bellabeat's users? For example, Bellabeat focuses mainly on females. Does this specific group weight some functionalities more than the general population, such as tracking menstrual cycles, and the design/appearance of the product?

3 Prepare

3.1 Source of data

This is the FitBit Fitness Tracker Data from Kaggle.

3.2 Check for integrity

This is an open-source dataset from Kaggle, which is a reliable website.

However, the data was gathered in 2016, which is six years ago. This makes our dataset a little outdated and thus can bias our results.

The number of samples in our dataset is very small (at most 33 participants), and we lack personal information about them (such as gender, age, and geographical location). Therefore, we are not very confident that these individuals can represent our interested population.

3.3 Description of the data

The DailyActivity dataset contains all the information in the DailyCalories, DailyIntensities, and DailySteps datasets. To have a look at the code that I used to preview the datasets to get the basic information, please see the section 4.1.

Dataset	Description
DailyActivity	33 participants, with total steps, distance walked, active minutes, and calories burned recorded over a one-month study period
DailyCalories	33 participants, daily calories burned recorded over a one-month study period
DailyIntensities	33 participants with (very/fairly/lightly/sedentary) active minutes and (very/fairly/lightly/sedentary) distance recorded over 1-month period
DailySteps	33 participants, daily total steps recorded over 1 month
DailySleep	24 participants, daily number of sleep, total minutes slept (daily), and total time in bed daily recorded over less than 1 month

Dataset	Description
HeartRate	14 participants, heart rate per second recorded
Weight	8 participants, weight (in kg and pounds) and BMI recorded
HourStep	33 participants, with activity hour and total steps per hour recorded

4 Process

4.1 Load and Preview Data

```
library(tidyverse)
```

We import our data

```
DailyActivity <- read_csv("dailyActivity_merged.csv")
DailyCalories <- read_csv("dailyCalories_merged.csv")
DailyIntensity <- read_csv("dailyIntensities_merged.csv")
DailySteps <- read_csv("dailySteps_merged.csv")
DailySleep <- read_csv("sleepDay_merged.csv")
HeartRate <- read_csv("heartrate_seconds_merged.csv")
Weight <- read_csv("weightLogInfo_merged.csv")
HourStep <- read_csv("hourlySteps_merged.csv")
```

We check the structure of these datasets and the number of participants in each dataset

```
str(DailyActivity)
length(unique(DailyActivity$Id))
str(DailyCalories)
length(unique(DailyCalories$Id))
str(DailyIntensity)
length(unique(DailyIntensity$Id))
str(DailySteps)
length(unique(DailySteps$Id))
str(DailySleep)
length(unique(DailySleep$Id))
str(HeartRate)
length(unique(HeartRate$Id))
str(Weight)
length(unique(Weight$Id))
str(HourStep)
length(unique(HourStep$Id))
```

4.2 Data Cleaning and Manipulation

To access the Data Cleaning Log, which records all the steps I did to clean up the data, please refer to the Appendix

4.2.1 Remove duplicates and NAs

```
#check number of NAs
is.na(DailyActivity) %>% sum()
is.na(DailySleep) %>% sum()
is.na(HeartRate) %>% sum()
is.na(Weight) %>% sum()
is.na(HourStep) %>% sum()

#deal with NAs in Weight
Weight <- Weight %>% select(-Fat) #Fat column contains only two values while the others are a
ll NAs

#check for duplicates
duplicated(DailyActivity) %>% sum()
duplicated(DailyCalories) %>% sum()
duplicated(DailyIntensity) %>% sum()
duplicated(DailySteps) %>% sum()
duplicated(DailySleep) %>% sum()
duplicated(HeartRate) %>% sum()
duplicated(Weight) %>% sum()
duplicated(HourStep) %>% sum()

#Remove duplicates in DailySleep
DailySleep <- DailySleep %>% distinct()
```

4.2.2 Check for ranges and outliers

```
#(DailySleep$TotalMinutesAsleep > 1440)
DailyActivity %>% select(TotalSteps, TotalDistance, Calories, ) %>% summary()
```

```
##    TotalSteps    TotalDistance       Calories
##  Min.   :    0   Min.   : 0.000   Min.   :   0
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.:1828
##  Median : 7406   Median : 5.245   Median :2134
##  Mean   : 7638   Mean   : 5.490   Mean   :2304
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:2793
##  Max.   :36019   Max.   :28.030   Max.   :4900
```

```
HeartRate %>% select(Value) %>% summary()
```

```
##      Value
##  Min.   : 36.00
##  1st Qu.: 63.00
##  Median : 73.00
##  Mean   : 77.33
##  3rd Qu.: 88.00
##  Max.   :203.00
```

```
DailySleep %>% select(TotalMinutesAsleep) %>% summary()
```

```
##  TotalMinutesAsleep
##  Min.   : 58.0
##  1st Qu.:361.0
##  Median :432.5
##  Mean   :419.2
##  3rd Qu.:490.0
##  Max.   :796.0
```

```
Weight %>% select(WeightKg, BMI) %>% summary()
```

```
##     WeightKg           BMI
##  Min.   : 52.60   Min.   :21.45
##  1st Qu.: 61.40   1st Qu.:23.96
##  Median : 62.50   Median :24.39
##  Mean   : 72.04   Mean   :25.19
##  3rd Qu.: 85.05   3rd Qu.:25.56
##  Max.   :133.50   Max.   :47.54
```

4.2.3 Rename variables

```
Weight <- Weight %>% rename(WeightDate = Date)
```

4.2.4 fix the format of date

```
#Seperate date-time format to date and time
DailySleep$SleepDay <- as.POSIXct(DailySleep$SleepDay, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.
timezone())
DailySleep$sleeptime <- format(DailySleep$SleepDay,format = "%H:%M:%S")
DailySleep$Date <- format(DailySleep$SleepDay,format = "%m/%d/%y")

DailyActivity$ActivityDate = as.POSIXct(DailyActivity$ActivityDate, format="%m/%d/%Y", tz=Sy
s.timezone())
DailyActivity$Date <- format(DailyActivity$ActivityDate, format = "%m/%d/%y")

Weight$WeightDate = as.POSIXct(Weight$WeightDate, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timez
one())
Weight$Date <- format(Weight$WeightDate, format = "%m/%d/%y")
Weight$time <- format(Weight$WeightDate, format = "%H:%M:%S")

HourStep$ActivityHour = as.POSIXct(HourStep$ActivityHour, format="%m/%d/%Y %I:%M:%S %p", tz=S
ys.timezone())
HourStep$activity_hour <- format(HourStep$ActivityHour,format = "%H:%M:%S")
HourStep$Activity_Date <- format(HourStep$ActivityHour, format = "%m/%d/%y")
```

4.2.5 Combine datasets

We combined the `DailyActivity` dataset and the `DailySleep` dataset. Now, the `DailyActivity` dataset is updated with new information on users' sleep monitoring.

```
DailyActivity <- merge(DailyActivity, DailySleep, by = c("Id", "Date"))
```

```
head(DailyActivity)
```

```
##           Id     Date ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366 04/12/16   2016-04-12      13162          8.50            8.50
## 2 1503960366 04/13/16   2016-04-13      10735          6.97            6.97
## 3 1503960366 04/15/16   2016-04-15       9762          6.28            6.28
## 4 1503960366 04/16/16   2016-04-16      12669          8.16            8.16
## 5 1503960366 04/17/16   2016-04-17       9705          6.48            6.48
## 6 1503960366 04/19/16   2016-04-19      15506          9.88            9.88
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0              1.88                     0.55
## 2                        0              1.57                     0.69
## 3                        0              2.14                     1.26
## 4                        0              2.71                     0.41
## 5                        0              3.19                     0.78
## 6                        0              3.53                     1.32
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                2.83                       0                29
## 4                5.04                       0                36
## 5                2.51                       0                38
## 6                5.03                       0                50
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories   SleepDay
## 1                  13                  328              728     1985 2016-04-12
## 2                  19                  217              776     1797 2016-04-13
## 3                  34                  209              726     1745 2016-04-15
## 4                  10                  221              773     1863 2016-04-16
## 5                  20                  164              539     1728 2016-04-17
## 6                  31                  264              775     2035 2016-04-19
##   TotalSleepRecords TotalMinutesAsleep TotalTimeInBed sleeptime
## 1                 1                327            346  00:00:00
## 2                 2                384            407  00:00:00
## 3                 1                412            442  00:00:00
## 4                 2                340            367  00:00:00
## 5                 1                700            712  00:00:00
## 6                 1                304            320  00:00:00
```

We also combine `Weight` dataset with `DailyActivity` dataset. Now, the `DailyActivity` dataset contains information on individuals' weight and BMI. We named this new dataset as `DailyActivity_Weight` .

```
DailyActivity_Weight <- merge(DailyActivity, Weight, by = c("Id", "Date"))
head(DailyActivity_Weight)
```

```
##           Id     Date ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366 05/02/16   2016-05-02      14727          9.71            9.71
## 2 1503960366 05/03/16   2016-05-03      15103          9.66            9.66
## 3 1927972279 04/13/16   2016-04-13        356          0.25            0.25
## 4 4558609924 05/01/16   2016-05-01       3428          2.27            2.27
## 5 5577150313 04/17/16   2016-04-17      12231          9.14            9.14
## 6 6962181067 04/12/16   2016-04-12      10199          6.74            6.74
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0              3.21                     0.57
## 2                        0              3.73                     1.05
## 3                        0              0.00                     0.00
## 4                        0              0.00                     0.00
## 5                        0              5.98                     0.83
## 6                        0              3.40                     0.83
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                5.92                       0                41
## 2                4.88                       0                50
## 3                0.25                       0                 0
## 4                2.27                       0                 0
## 5                2.32                       0               200
## 6                2.51                       0                50
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories   SleepDay
## 1                  15                  277              798     2004 2016-05-02
## 2                  24                  254              816     1990 2016-05-03
## 3                   0                   32              986     2151 2016-04-13
## 4                   0                  190             1121     1692 2016-05-01
## 5                  37                  159              525     4552 2016-04-17
## 6                  14                  189              796     1994 2016-04-12
##   TotalSleepRecords TotalMinutesAsleep TotalTimeInBed sleeptime
## 1                 1                277            309  00:00:00
## 2                 1                273            296  00:00:00
## 3                 1                398            422  00:00:00
## 4                 1                115            129  00:00:00
## 5                 1                549            583  00:00:00
## 6                 1                366            387  00:00:00
##            WeightDate WeightKg WeightPounds   BMI IsManualReport        LogId
## 1 2016-05-02 23:59:59     52.6     115.9631 22.65           TRUE 1.462234e+12
## 2 2016-05-03 23:59:59     52.6     115.9631 22.65           TRUE 1.462320e+12
## 3 2016-04-13 01:08:52    133.5     294.3171 47.54          FALSE 1.460510e+12
## 4 2016-05-01 23:59:59     69.9     154.1031 27.32           TRUE 1.462147e+12
## 5 2016-04-17 09:17:55     90.7     199.9593 28.00          FALSE 1.460885e+12
## 6 2016-04-12 23:59:59     62.5     137.7889 24.39           TRUE 1.460506e+12
##       time
## 1 23:59:59
## 2 23:59:59
## 3 01:08:52
## 4 23:59:59
## 5 09:17:55
## 6 23:59:59
```

4.2.6 transform variables

We created a new variable – `slep_p` , which is `TotalMinuetesAsleep` divided by the `TotalTimeInBed` . By doing this, we get the proportion of the time that each user is asleep during their total time in bed.

We also converted the time measurement into hours, which is easier to interpret and analyze.

```
#
DailyActivity <- DailyActivity %>%
  mutate(
    slep_p =  TotalMinutesAsleep / TotalTimeInBed
   ,awake_p = 1 - (TotalMinutesAsleep / TotalTimeInBed)
   ,Hr_Sleep = TotalMinutesAsleep/60
   ,Hr_InBed = TotalTimeInBed/60
  )
```

5 Analyze and Share

To gain insights into how users are using our product, one of the most efficient approaches is to understand our users' characteristics from multiple dimensions, such as how they are sleeping, how active they are, during what time period they are active, and whether they have a healthy BMI.

5.1 Sleep

To begin our analysis, I will first focus on the sleeping hours. A healthy life is strongly related to good quality sleeping. To help users to promote their health when using our product, we first need to get to know how they are sleeping.

Below we calculated each individual's daily average sleeping hours over the study period.

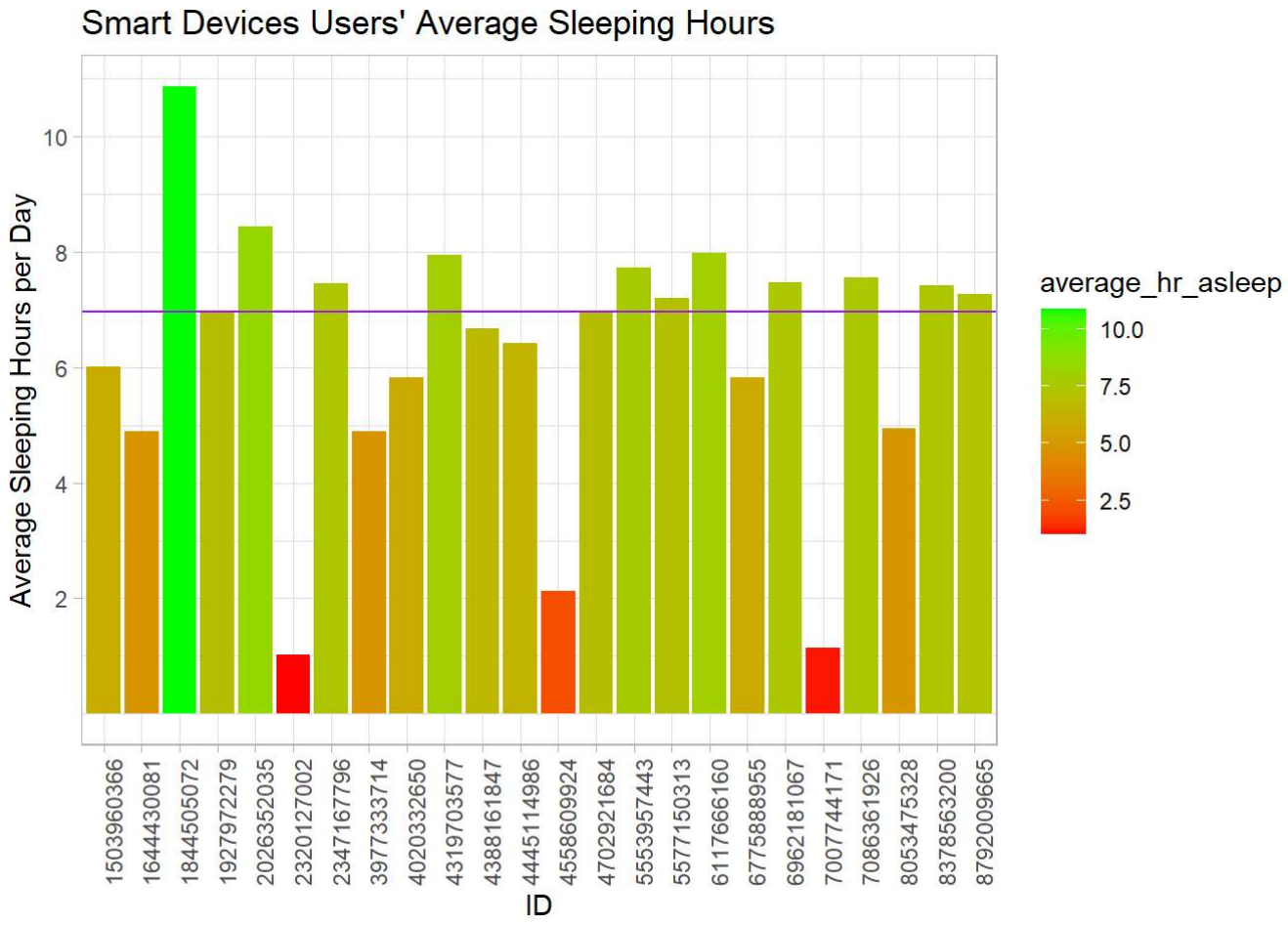
```
Average_Hour_Asleep <- DailyActivity %>% group_by(Id) %>% summarize( total = n(), average_hr_
asleep = (sum(Hr_Sleep)/total))
Average_Hour_Asleep %>%
  mutate( Id = Average_Hour_Asleep$Id %>% as.character())
```

```
## # A tibble: 24 x 3
##    Id         total average_hr_asleep
##    <chr>      <int>             <dbl>
##  1 1503960366    25              6.00
##  2 1644430081     4              4.9
##  3 1844505072     3             10.9
##  4 1927972279     5              6.95
##  5 2026352035    28              8.44
##  6 2320127002     1              1.02
##  7 2347167796    15              7.45
##  8 3977333714    28              4.89
##  9 4020332650     8              5.82
## 10 4319703577    26              7.94
## # ... with 14 more rows
```

```
#str(Average_Hour_Asleep)
```

```
library(ggplot2)

p <- ggplot(Average_Hour_Asleep, aes(x=as.character(Id), y=average_hr_asleep, fill = average_
hr_asleep)) + theme_light()
p <- p + geom_col() + theme(axis.text.x = element_text(angle = 90))
p <- p + scale_fill_gradient(low = "red", high = "green")
p <- p + geom_hline(yintercept = median(Average_Hour_Asleep$average_hr_asleep), color = "purp
le")
p <- p + labs(title = "Smart Devices Users' Average Sleeping Hours", x = "ID", y = "Average S
leeping Hours per Day" )
p <- p + scale_y_continuous(breaks = c(2, 4, 6, 8, 10))
print(p)
```



From this bar chart above, we can see that the **majority** of our samples **sleep around 6 to 7 hours**, while **three** of them **sleep much less** than others (around 1.5 hours) and **one** of them **sleep much more** than others (around 11 hours).

According to the CDC, the *recommended hours of sleep per day* for is 7 or more hours for adults (link is here) (https://www.cdc.gov/sleep/about_sleep/how_much_sleep.html). To gain an insight on how much of our population meets this recommendation, I divided our sample into three categories based on their sleeping hours records:

- lack of sleep: users who sleep less than the recommended 7 hours
- normal sleep: users who sleep more than or equal to 7 hours but less than or equal to 10 hours per day
- excessive sleep: users who sleep more than 10 hours per day

```
Average_Hour_Asleep <- Average_Hour_Asleep %>%
  mutate(
    sleep_habit = case_when(
      average_hr_asleep < 7 ~ 'lack of sleep'
     ,average_hr_asleep >= 7 & average_hr_asleep <= 10 ~ 'normal sleep'
     ,average_hr_asleep > 10 ~ 'excessive sleep'
    )
  )

Average_Hour_Asleep_percent <- Average_Hour_Asleep %>%
  group_by(sleep_habit) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(sleep_habit) %>%
  summarise(total_percent = total / totals) %>%
  mutate(prop = paste(100*round(total_percent, 2), "%"))
Average_Hour_Asleep_percent$sleep_habit <- factor(Average_Hour_Asleep_percent$sleep_habit , l
evels = c("lack of sleep", "normal sleep", "excessive sleep"))
```

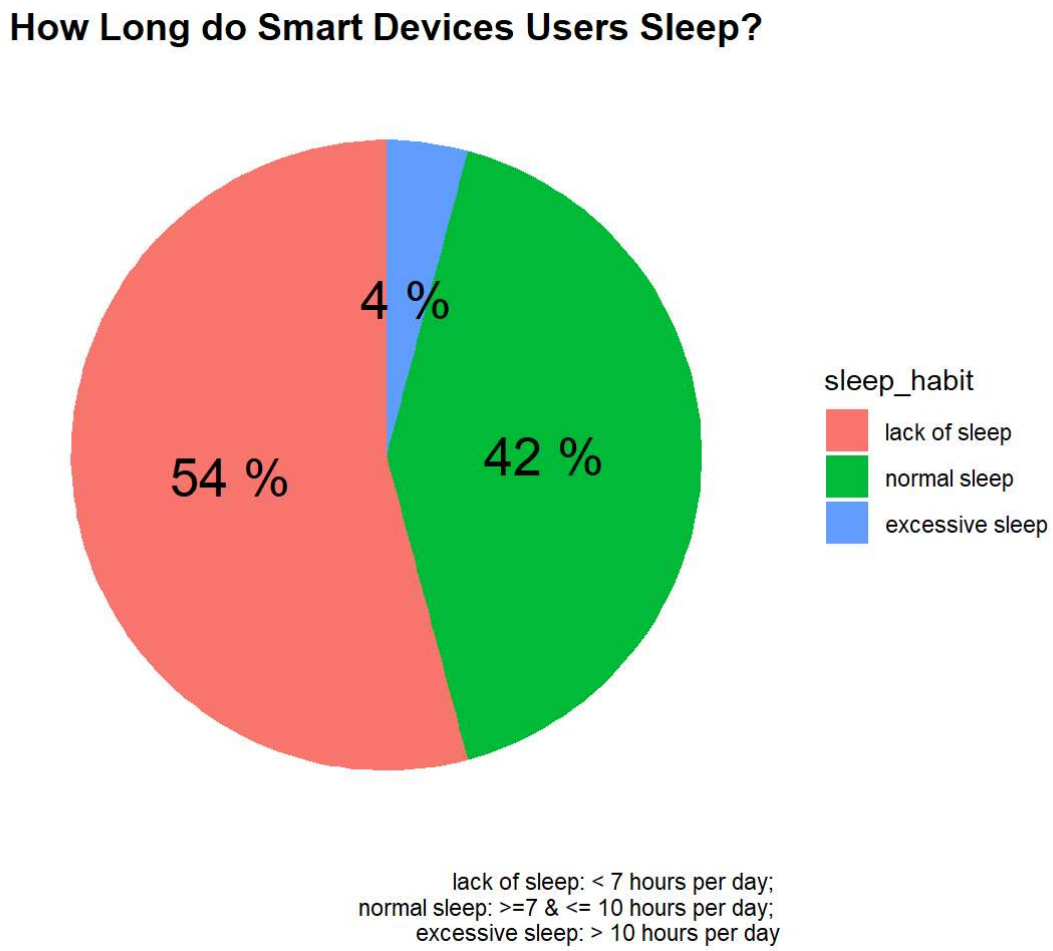
The following table describes the proportion of individuals who do not follow the 7-hour recommendation by the CDC and sleep less than it (`lack of sleep`), the proportion of individuals who follow the 7-hour recommendation and have a sufficient amount of sleeping (`normal sleep`), and proportion of individuals who sleep way more than the recommended 7-hours (`excessive sleep`).

```
Average_Hour_Asleep_percent
```

```
## # A tibble: 3 x 3
##   sleep_habit     total_percent prop
##   <fct>                   <dbl> <chr>
## 1 excessive sleep        0.0417 4 %
## 2 lack of sleep          0.542  54 %
## 3 normal sleep           0.417  42 %
```

This pie chart below gives a better visualization of the proportions that each group takes up.

```
p <- ggplot(Average_Hour_Asleep_percent, aes(x = "", y = total_percent, fill = sleep_habit))
 + geom_bar(width = 1, stat = "identity")
p <- p + coord_polar(theta = "y")
p <- p + theme_minimal()
p <- p + 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"))
p <- p + geom_text(aes(label = prop), position = position_stack(vjust = 0.5), size = 7)
p <- p + labs(title = "How Long do Smart Devices Users Sleep?", caption = "lack of sleep: < 7
hours per day; \nnormal sleep: >=7 & <= 10 hours per day; \nexcessive sleep: > 10 hours per d
ay")
print(p)
```

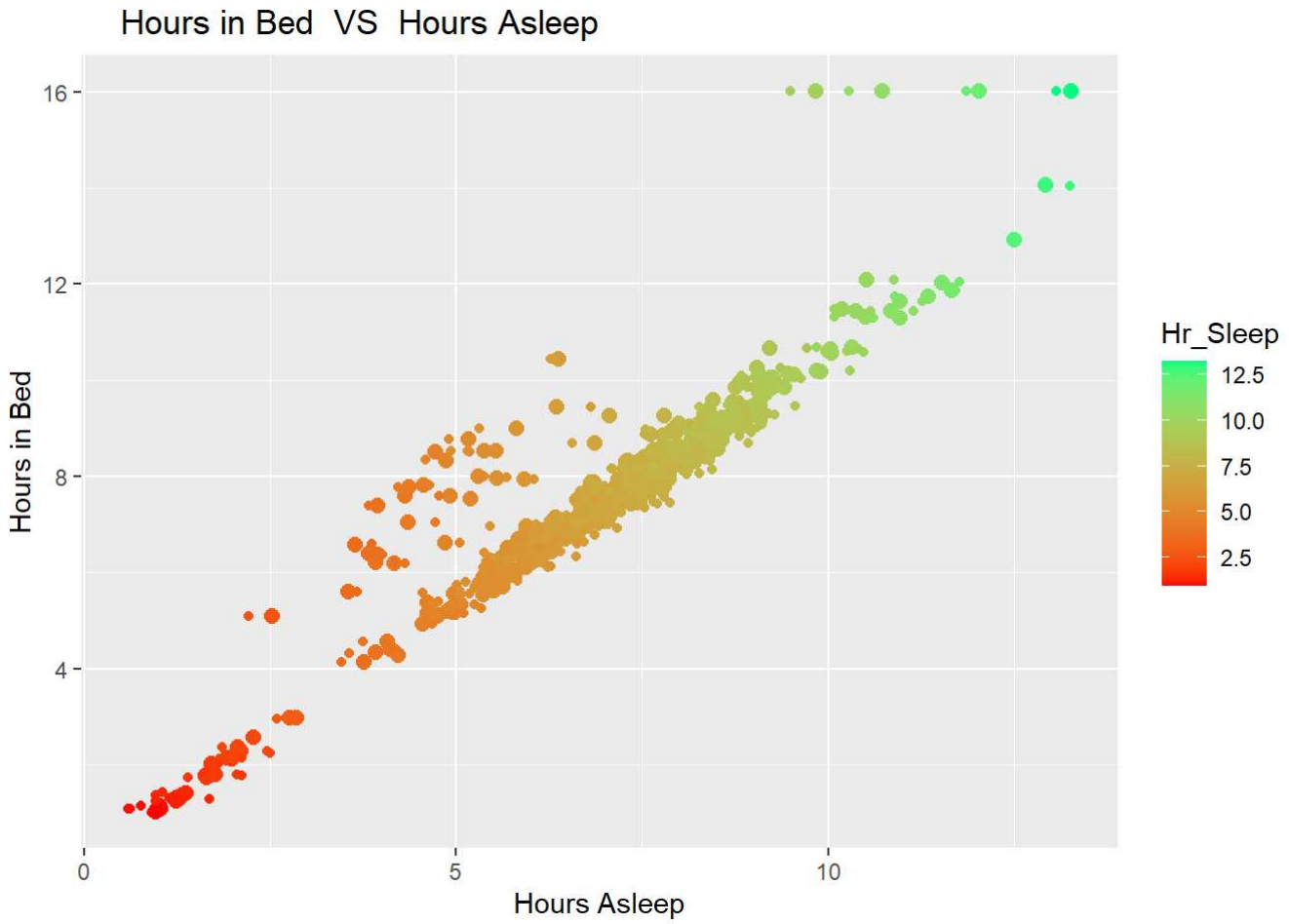


The table and the pie chart both show that **over half of our samples (54%)** do not follow the advice given by the CDC that they **sleep less than 7 hours** on average. If our sample represents the population (there are some limitations that impair the reliability of the sample), we can get a rough overview of the population that over 50% of them do not have enough sleep to maintain their health.

Therefore, we need to find a way to help users increase their sleeping time.

In the scatter plot below, we can see that the **total time in bed** is **highly correlated** with the **total time asleep**, which means that the **higher the hours in bed**, the **longer the hours asleep**. Therefore, I have a suggestion: the “Leaf” product can develop/ fine-tune its functionality in reminding users to go to bed, which helps users to stay in bed longer and thus increase their total hours of sleep.

```
p <- ggplot(DailyActivity, aes(x = Hr_Sleep, y = Hr_InBed))
p <- p + geom_point(aes(color=Hr_Sleep), size = 2.5)
p <- p + scale_color_gradient(low = "red1", high = "springgreen1")
p <- p + geom_jitter(position = position_jitter(0.5), aes(color = Hr_Sleep))
p <- p + labs(title = "    Hours in Bed  VS  Hours Asleep", x = "Hours Asleep", y = "Hours in
Bed")
print(p)
```



If reminding users to go to bed helps lengthen their total time in bed and thus help them sleep longer, when should our product send this notification?

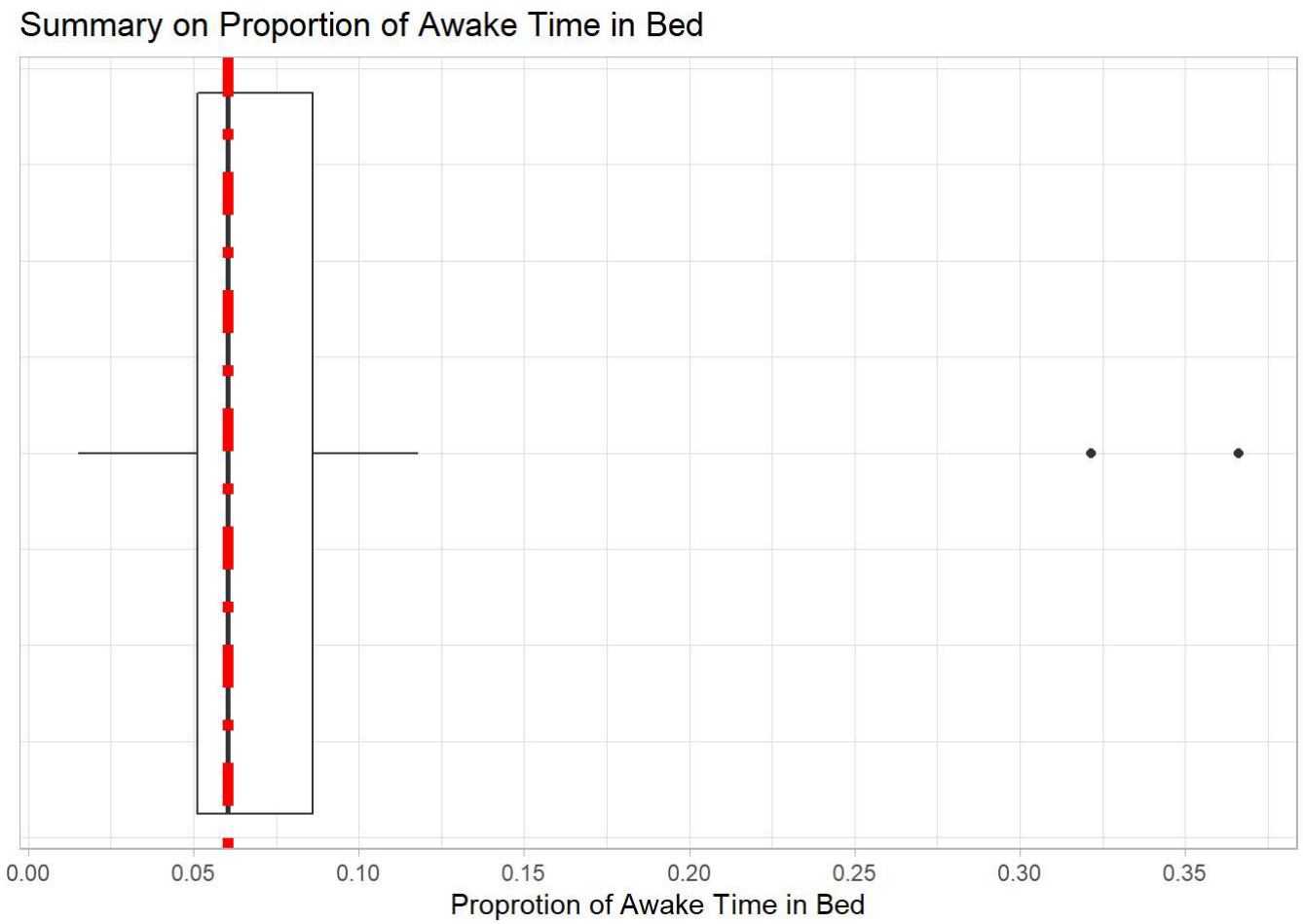
In the code below, I first calculated each individual's average daily awake time proportion, over this study period. Then, the five-number summary, a statistical summary of the min, 1st quarter, median, third quarter, and max is performed.

```
Average_Awake_Proprotion <- DailyActivity %>%
  group_by(Id) %>%
  summarize(num = n(), average_awake_p = sum(awake_p)/num)
Average_Awake_Proprotion$average_awake_p %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01511 0.05126 0.06030 0.08697 0.08600 0.36633
```

This box plot creates a better visualization of describing the median (indicated by the red dashed line), min, max, first quarter, and third quarter of the average proportions of awake time in bed.

```
p <- ggplot(Average_Awake_Proprotion, aes(y = average_awake_p))
p <- p + geom_boxplot()
p <- p + geom_hline(yintercept = median(Average_Awake_Proprotion$average_awake_p), linetype =
4, size = 2, color = "red") + theme_light()
p <- p + theme(axis.title.y = element_blank(), axis.text.y=element_blank (), axis.ticks.y=ele
ment_blank ())
p <- p + coord_flip()
p <- p + scale_y_continuous( breaks = c(0.0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4))
p <- p + labs(title = "Summary on Proportion of Awake Time in Bed", y = "Proprotion of Awake
 Time in Bed")
print(p)
```



From the five-number summary and the box plot, we can see that the median is about 0.06, which means that **50% of our sample has 6% or less of their total time in bed awaking**.

Therefore, here is my ***first suggestion*** to the Bellabeat company on improving our "Leaf" product:

- To remind users to sleep, the "Leaf" product should **send a notification to users** reminding them to go to bed **more than 6% of their average total time in bed before their usual sleeping time**. By doing so, we expect to put them in bed longer and increase their amount of sleep. For example, if a user's usual time in bed is 8 hours per day, and he wants to be asleep at 10 pm, then the "Leaf" product should send a notification about 0.5 to 1 hour before 10 pm. By **lengthening their time in bed**, **we expect that users who lack sleep can sleep longer**.
- If it is the user's first time using this product and we do not have the records about their normal sleeping time, we can ask them to input their ideal time in bed and their desired sleeping time, and our product will automatically calculate the amount of time that a notification needs to be sent in advance. While users continue to use this functionality, the product will record and update the data on total time in bed and hours asleep and thus make the calculation more accurate.

Some **limitations** of this suggestion are that:

- The product can have control over when users go to sleep, but no control over when they will rise. For example, a person's usual time in bed is only 5 hours, we can send a notification to them and remind them to go to bed early, but we can not control their time to rise. If the person slept 5 hours and rises up, we can still not help them to meet the recommended 7 hours of sleep.

5.2 Activeness

Activeness is another aspect of our users that we want to investigate. The total number of steps or the total distance walked are both possible measures of an individual's activeness. However, since different people may walk longer/shorter in one step, the total distance walked is a more reliable measurement since its unit is consistent. Therefore, in this section of our analysis, we use `TotalDistance` to measure the activeness of our users.

Based on the (average) total distance people walk daily, we categorize these individuals into 4 groups:

- Sedentary: people who walk less than 0.5 miles.
- Lightly Active: people who walk more than 0.5 miles but less than 3 miles.
- Active: people who walk between 3 miles and 6 miles.
- Very Active: people who walk more than 6 miles per day.

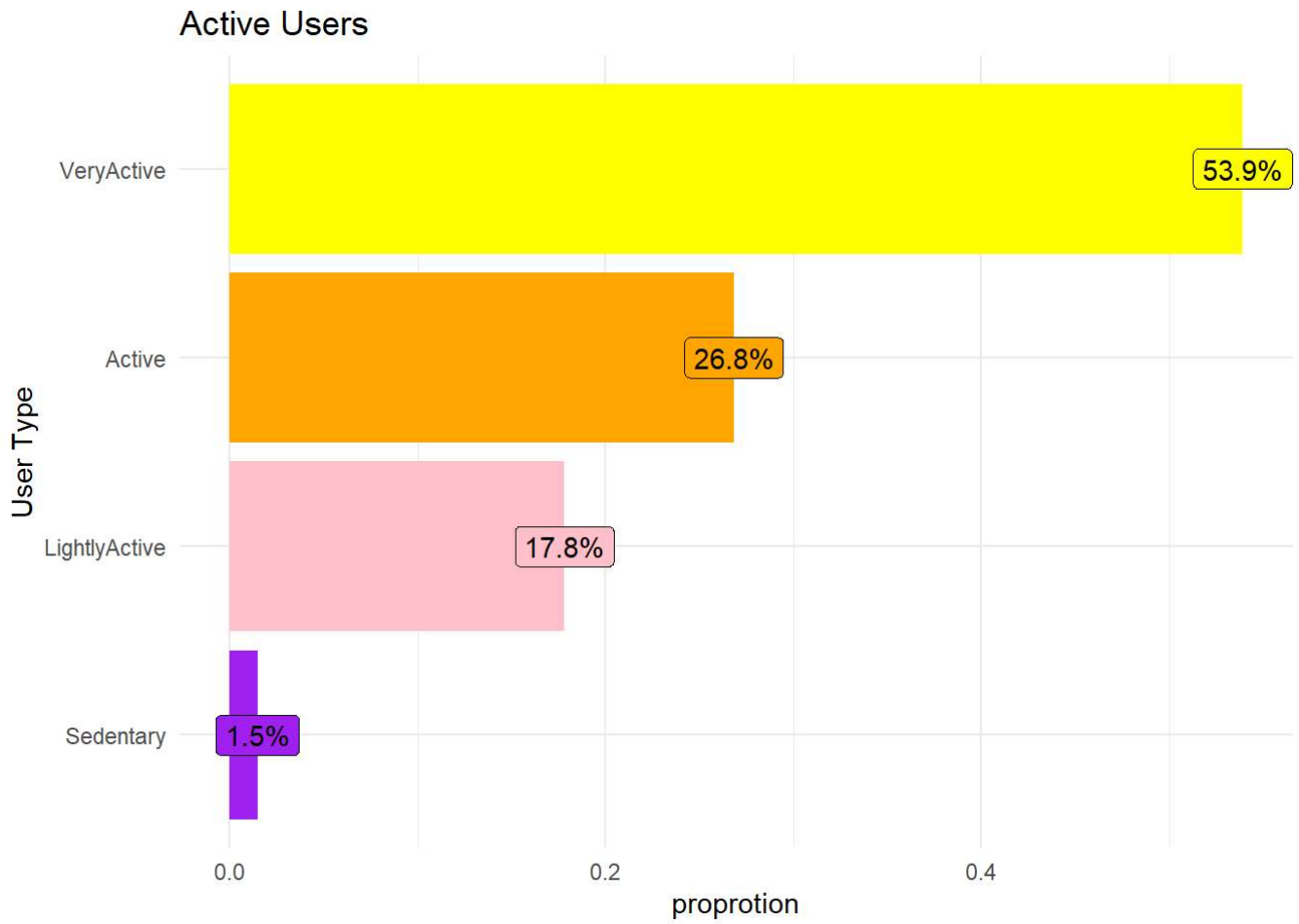
```
DailyActivity <- DailyActivity %>%
  mutate(
    UserType = case_when(
      TotalDistance < 0.5 ~ "Sedentary"
     ,TotalDistance >= 0.5 & TotalDistance < 3 ~ "LightlyActive"
     ,TotalDistance >= 3 & TotalDistance <6 ~ "Active"
     ,TotalDistance >= 6 ~ "VeryActive"
    )
  )

UserType <- DailyActivity %>% select(Id, UserType)
UserType$UserType <- factor(UserType$UserType , levels = c("Sedentary", "LightlyActive", "Act
ive", "VeryActive"))
UserType_Percent <- UserType %>%
  group_by(UserType) %>%
  summarize(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(UserType) %>%
  summarize(total_percent = total / totals) %>%
  mutate(lab = scales::percent(total_percent))


UserType_Percent
```

```
## # A tibble: 4 x 3
##   UserType      total_percent lab
##   <fct>                 <dbl> <chr>
## 1 Sedentary            0.0146 1.5%
## 2 LightlyActive        0.178  17.8%
## 3 Active               0.268  26.8%
## 4 VeryActive           0.539  53.9%
```

```
p <- ggplot(UserType_Percent, aes(x = UserType, y = total_percent)) + geom_col(aes(fill = Use
rType), show.legend = FALSE) + coord_flip()
p <- p + theme_minimal()
p <- p + scale_fill_manual(values = c("purple", "pink", "orange", "yellow"))
p <- p + geom_label(aes(label = lab, fill = UserType), show.legend = FALSE)
p <- p + labs(title = "Active Users", y = "proprotion", x = "User Type")
print(p)
```



The above table and bar graph show that the **majority of our sample is very active**: 80.7% of them walk more than 3 miles a day on average, and 53.9% of them walk more than 6 miles per day on average.

Therefore, we expect a large proportion of our "Leaf" product users to be very active and exercise very often. In other words, one of our **main targets** is the group of **consumers who enjoy outdoor exercise more often**.

Based on this insight, here is my ***second suggestion***:

- When **advertising our "Leaf" product**, we should **focus largely on the group of people who like outdoor exercises**. We can use algorithms to identify and categorize the population based on their search keywords. For example, if a person like outdoor exercise, he or she might search for keywords such as "biking", "jogging", and "hiking" more often. Once we identified the "active" group, we can send more advertisements (such as videos about a person who wears a "Leaf" product enjoying hiking on the beautiful trail) to this type of incoming customer.

However, we don't want to "give up" on the `Sedentary` group. We want to investigate these "sedentary" people, and we expect to figure out a way to encourage them to do more exercise.

Here are the Ids of all the samples in the `Sedentary` group.

```
SedentaryPeople <- UserType %>% filter(UserType == "Sedentary")
unique(SedentaryPeople$Id)
```

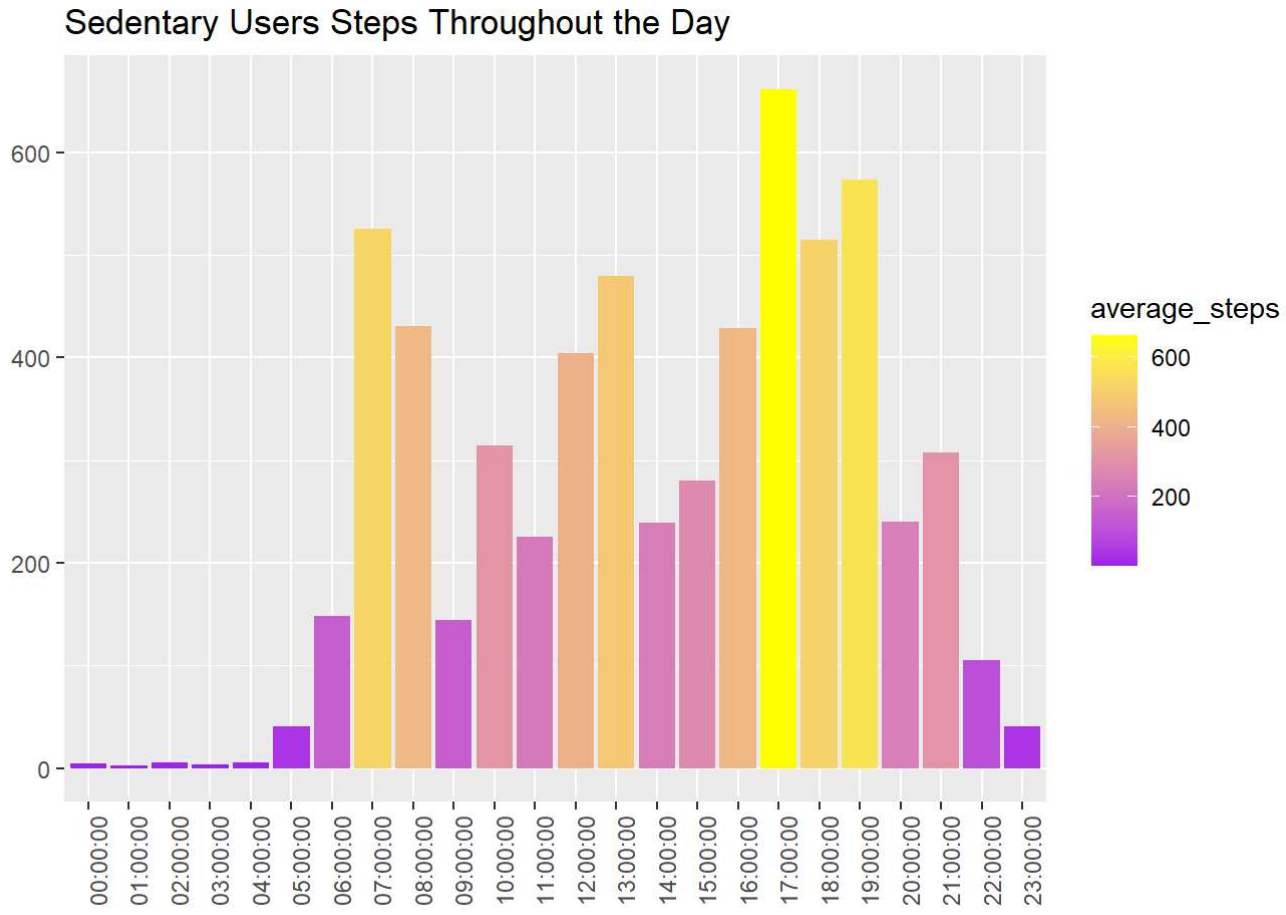
```
## [1] 1927972279 2026352035 2347167796 4319703577 5553957443
```

We used another dataset – `HourStep` to investigate on these people's activeness throughout the day.

```
HourStep_Sedentary <- HourStep %>% filter( Id == c("1927972279", "2026352035", "2347167796",
"4319703577", "5553957443"))
unique(HourStep_Sedentary$Id) #check the Id of these selected people
```

```
## [1] 1927972279 2026352035 2347167796 4319703577 5553957443
```

```
HourStep_Sedentary %>%
  group_by(activity_hour) %>%
  summarize(average_steps = mean(StepTotal)) %>%
  ggplot(aes(x=activity_hour, y = average_steps, fill = average_steps)) +
  geom_col() +
  labs(title = "Sedentary Users Steps Throughout the Day", x="", y="") +
  scale_fill_gradient(low = "purple", high = "yellow")+
  theme(axis.text.x = element_text(angle = 90))
```



From the bar graph above, we can see that people in the **"Sedentary group" are most active from 5 PM to 7 PM**.

After observing this pattern, I have my ***third suggestion***:

- In order to encourage the sedentary group to do more exercise and help them to live a healthier way of life, our “Leaf” product can **send users a reminder that encourages them to go out for a walk before 5:00 PM**. For example, it can send a good quote such as “All truly great thoughts are conceived while walking.”
- If we want to attract people who are less active to use our product, we can **use advertisements differently and creatively**. For example, we can invite some “Leaf” users to talk about how this product has changed their way of living or helped promote their health, and then we can send this advertisement to the “sedentary” type of incoming consumers.

5.3 BMI

BMI is a reliable measurement of body fat. Body fat is highly associated with people’s health conditions, so we want to investigate our sample in terms of their body fat.

Our samples are separated into four categories:

- Underweight: people with BMI less than 18
- Normal: people with BMI greater than or equal to 18 and less than 25
- Overweight: people with BMI greater than or equal to 25 but less than 30
- Obese: people with BMI greater than or equal to 30

```
bodyfatness <- Weight %>%
  group_by(Id) %>%
  summarize(total = n(), average_bmi = sum(BMI)/total)
```

```
DailyActivity_W <- bodyfatness %>%
  mutate(
    bodyfat =
      case_when(
        average_bmi < 18 ~ "Underweight",
        average_bmi >= 18  & average_bmi < 25 ~ "healthy",
        average_bmi >= 25 & average_bmi < 30 ~ "Overweight",
        average_bmi > 30 ~ "Obese"
        )
  )



DailyActivity_W$bodyfat <- DailyActivity_W$bodyfat %>% factor(levels = c("Underweight", "heal
thy", "Overweight", "Obese"))
DailyActivity_W %>% select(-total)
```

```
## # A tibble: 8 x 3
##           Id average_bmi bodyfat
##        <dbl>       <dbl> <fct>
## 1 1503960366        22.6 healthy
## 2 1927972279        47.5 Obese
## 3 2873212765        21.6 healthy
## 4 4319703577        27.4 Overweight
## 5 4558609924        27.2 Overweight
## 6 5577150313        28   Overweight
## 7 6962181067        24.0 healthy
## 8 8877689391        25.5 Overweight
```

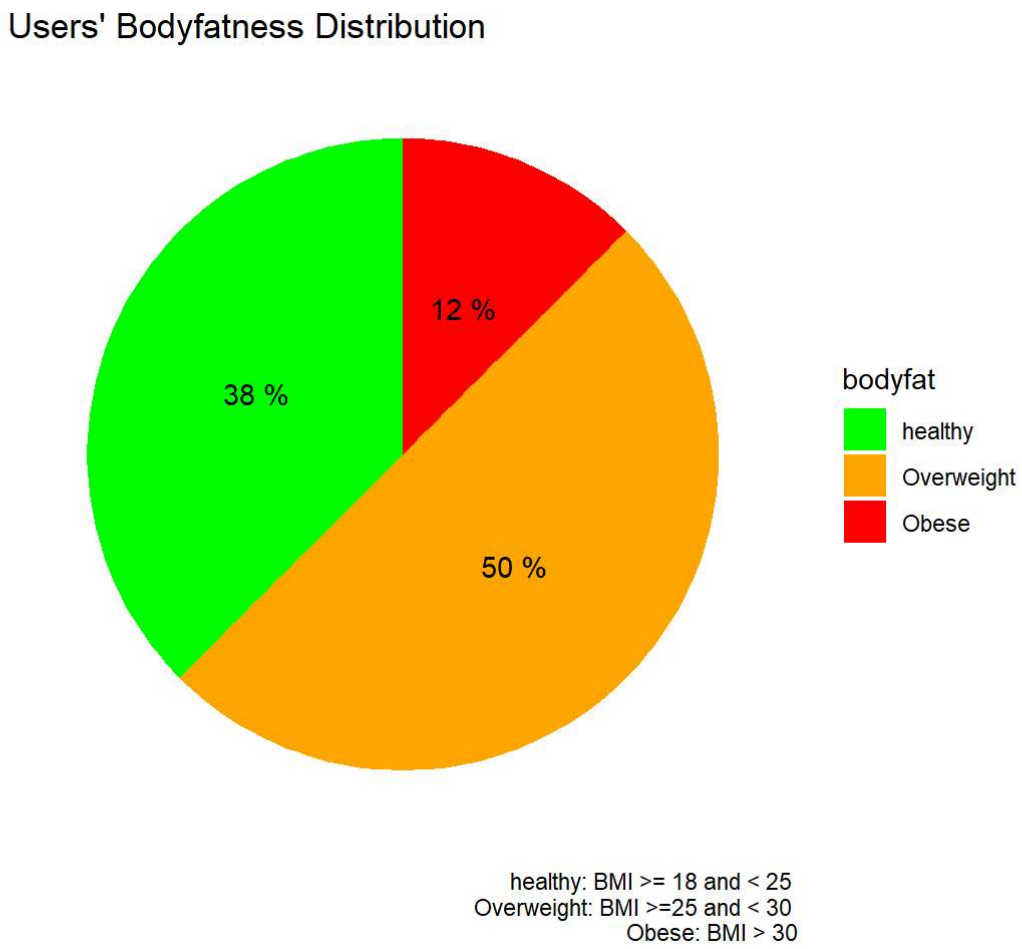
```
DailyActivity_Weight_P <- DailyActivity_W %>%
  group_by(bodyfat) %>%
  summarize(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(bodyfat) %>%
  summarize(total_percent = total/totals) %>%
  mutate(prop = paste(100*round(total_percent, 2), "%"))



DailyActivity_Weight_P
```

```
## # A tibble: 3 x 3
##   bodyfat    total_percent prop
##   <fct>              <dbl> <chr>
## 1 healthy            0.375 38 %
## 2 Overweight         0.5   50 %
## 3 Obese              0.125 12 %
```

```
p <- ggplot(DailyActivity_Weight_P, aes(x = "", y = total_percent, fill = bodyfat))
p <- p + geom_bar(width = 1, stat = "identity") + coord_polar(theta = "y")
p <- p + scale_fill_manual(values = c("green", "orange", "red"))
p <- p + 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() )
p <-  p + geom_text(aes(label = prop), position = position_stack(vjust = 0.5))
p <- p + labs(title = "Users' Bodyfatness Distribution", caption = "healthy: BMI >= 18 and <
 25 \nOverweight: BMI >=25 and < 30 \nObese: BMI > 30")
print(p)
```

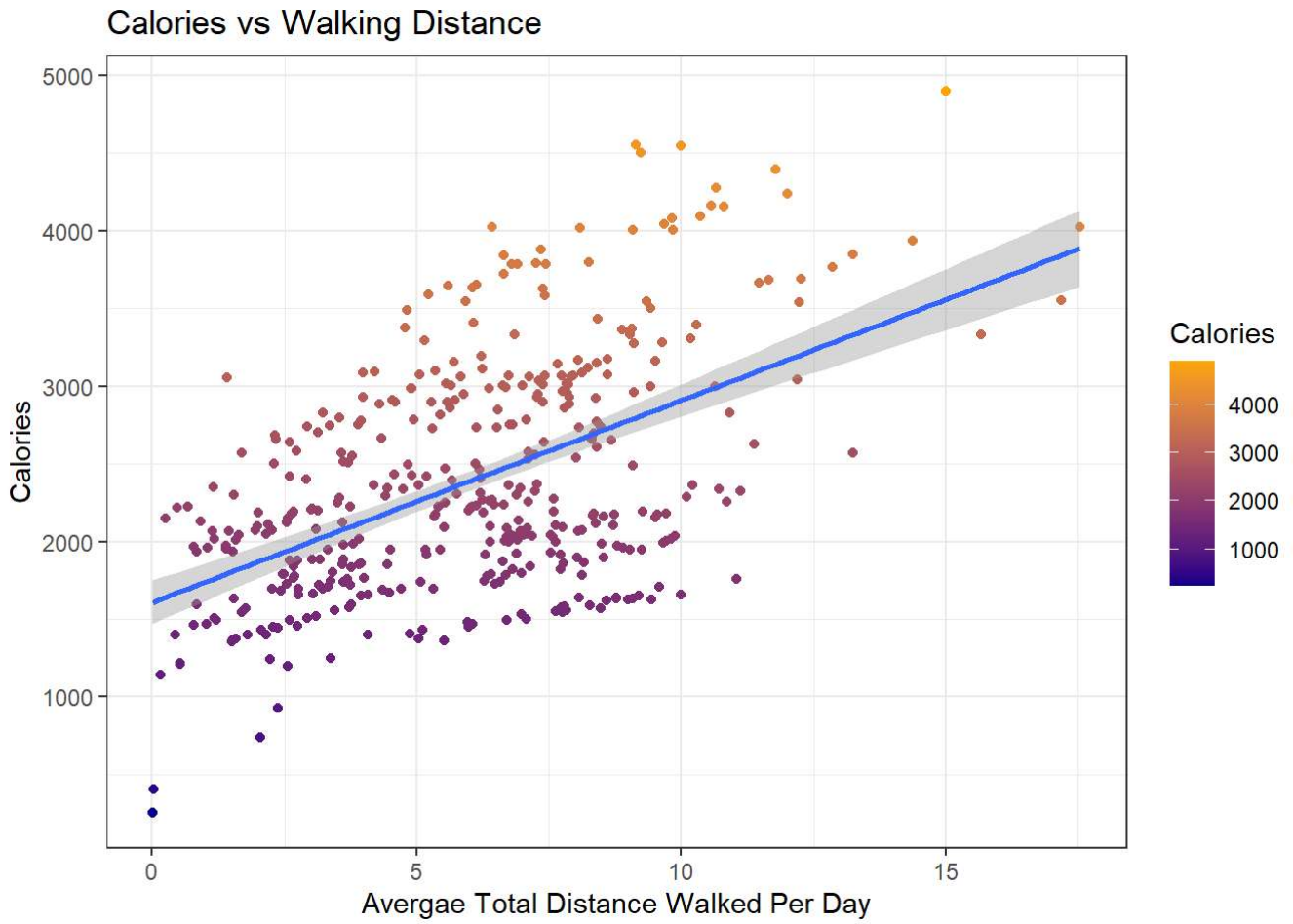


We get **a conclusion** that 38% of users have normal body fat, 50% of users are overweight, and 12% of users are obese. In other words, **half of the users have a BMI from 25 to 30**, which is only 0 to 5 points more than the limit of the healthy group. This group of people **has a great potential for losing weight**.

Note that this analysis has some **limitations**. We only have eight participants in our sample, and the number of BMI records is different among these individuals. Also, BMI is related to other predictors such as gender and age, but our dataset does not contain additional information on that. Therefore, this dataset might not be an excellent representation of the whole user population. Further data collection or using other open data sources might be needed.

```
p <- ggplot(DailyActivity, aes(x = TotalDistance, y = Calories))
p <- p + geom_point(aes(color = Calories)) + scale_color_gradient(high = "orange", low = "dar
kblue")
p <- p + geom_smooth(method = lm)
p <- p + theme_bw()
p <- p + labs(title = "Calories vs Walking Distance", x = "Avergae Total Distance Walked Per
 Day", y = "Calories")
print(p)
```

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



Losing weight relates to exercise and calories burned. From the above scatterplot, we can see that the calories are strongly related to the distance walked per day. The **longer the distance an individual walks per day**, the **higher the calories burned**.

Based on this, I have my ***fourth suggestion***:

- When a user first uses our product, we can ask them to enter in the information on their gender, age, height, and weight. Then, by following the algorithms, our product can calculate the BMI score for our users and categorize these users into the four categories above (underweight, healthy, overweight, and obese).
- If the users are obese or overweight, the "Leaf" product can notify them to **set up a goal for daily exercises**, for example, how many steps they would like to walk per day. Instead of making them think of the goal and enter it by themselves, **letting people choose from a couple of choices** might be more efficient to "force" them to take action. For example, instead of letting users enter the distance that they want to walk, our "Leaf" product can give them a list of distance values to let them choose from based on their physical activity records and BMI.
- If the users are underweight, our product can **send them more information on healthy and regular meals** or some **exercises that help them build muscles**. Our product can also suggest them to set up goals such as the number of meals they want to eat per day, or the type of exercise they want to do.
- Once our users **achieved a goal**, the product sends a **notification that congratulates them**, which can encourage them to continue to follow their plan. If a certain number of times they have achieved their goals, for example, 1000 times, we can give them a **discount on purchasing new products**.

6 Conclusion

Based on our analysis, We summarize our suggestions as below:

Advertising

We expect the majority of our customers to be active. Advertisements should focus largely on the need of this group of people.

Ex: advertisements with elements of outdoor hiking, jogging, fresh air, etc.

Customized Notifications

Half of our users are overweight, a large proportion of users sleep less than 7 hours per day, and a small proportion of our users are sedentary. Different encouraging notifications should send to different users to motivate them at the right time.

Ex1: Send notifications more than 6% of “lack of sleep” users’ total time in bed before their usual time to bed. This encourages them to go to bed early, and we expect this long time in bed will lengthen their total time asleep.

Ex2: Send motivation notifications to users who are sedentary before 5 PM, because we expect them to be the most active from 5 PM to 7 PM within a day.

Reward system

Our product can let users set up/ choose their daily goals, and once a certain number of goals are achieved, some rewards should be given to them.

Ex1: Goals can be total distance walked per day, cups of water per day, number of meals per day, and total hours of sleep per day.

Ex2: If a person achieved his/ her goals 100 times, the Bellabeat company can give them a discount that can be used when purchasing the company’s new product. This not only encourages users to live a healthier life but also simulates consumers buying our new products.

Limitation

Our sample size is small and outdated. A further collection of current data or using other open data sources is needed.

Future Directions

The patterns and insights we found can be applied to both men and women. However, since Bellabeat is a company that focuses on women’s health, some patterns that only apply to women can be further investigated. For example, we can collect and analyze female menstrual cycle records and their feelings during their periods to better help them take care of themselves.

7 Appendix

7.1 Data Cleaning Log

1. We have discovered that Weight has 65 NAs, and all of them came from Fat column – this column only contains 2 values and the others are all NAs. Therefore, instead of using drop.NA, we deleted the Fat column.
2. Duplicates are checked and deleted.
3. Some ranges of values are checked. For example, total daily sleeping hours are checked to make sure it is less than 24 hours.
4. The format of date is fixed to a consistent form across datasets
5. Two Datasets (DailyActivity and DailySleep) are merged into one based on Id and Date.

6. DailyActivity is then merged with Weight dataset. In Weight dataset, the date column is already been named as “Date”, so when we reformated the “Date” column, we ended up having two columns named “Date”, which makes our dataset look strange. Therefore, to remove any confusions, I renamed the date column as “WeightDate” first and then perform the formatting of date.