# Google Data Analytics Capstone Project

Gigio Gomes

2022-08-13

# Case Study 2 - How can a wellness technology company play it smart?

## Introduction and background

### Scenario

Bellabeat is a high-tech manufacturer of health-focused products for women. This is a successful small company, but they have the potential to become a larger player in the global smart device market.

As a data analyst working on the marketing analyst team, I have been asked to focus on one of Bellabeat's products and analyze smart device data to gain insight into how consumers are using their smart devices. The insights I discover will help guide marketing strategy for the company. I will present my analysis to the Bellabeat executive team along with recommendations for Bellabeat's marketing strategy.

The questions that will guide the analysis are:

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

In order to answer the key business questions, it will be followed the steps of the data analysis process: **ask**, **prepare**, **process**, **analyze**, **share** and **act**. And a report with the following deliverable will be produced:

1. A clear summary of the business task
2. A description of all data sources used
3. Documentation of any cleaning or manipulation of data
4. Supporting visualizations and key findings
5. The top high-level content recommendations based on the analysis

### About the company

Urška Sršen and Sando Mur founded Bellabeat, a high-tech company that manufactures health-focused smart products. Sršen used her background as an artist to develop beautifully designed technology that informs and inspires women around the world. Collecting data on activity, sleep, stress, and reproductive health has allowed Bellabeat to empower women with knowledge about their own health and habits. Since it was founded in 2013, Bellabeat has grown rapidly and quickly positioned itself as a tech-driven wellness company for women.

By 2016, Bellabeat had opened offices around the world and launched multiple products. Bellabeat products became available through a growing number of online retailers in addition to their own e-commerce channel on their website (https://bellabeat.com/). The company has invested in traditional advertising media, such as radio, out-of-home billboards, print, and television, but focuses on digital marketing extensively. Bellabeat

invests year-round in Google Search, maintaining active Facebook and Instagram pages, and consistently engages consumers on Twitter. Additionally, Bellabeat runs video ads on Youtube and display ads on the Google Display Network to support campaigns around key marketing dates.

Sršen knows that an analysis of Bellabeat's available consumer data would reveal more opportunities for growth. She has asked the marketing analytics team to focus on a Bellabeat product and analyze smart device usage data in order to gain insight into how people are already using their smart devices. Then, using this information, she would like high-level recommendations for how these trends can inform Bellabeat marketing strategy.

# Data Analisys Process

## 1. Ask

As described in the introduction section, the challenge is to discover some trends in smart devices usage, and then apply theses trends to the company's customers. It's expected that this research helps Bellabeat marketing strategy.

One of our stakeholders, Urška Sršen, cofounder and Chief Creative Officer of Bellabeat, believes that analyzing smart devices fitness data could help unlock new growth opportunities for the company.

Other key stakeholder to be considered is Sando Mur, mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team.

As a secondary stakeholder, the Bellabeat marketing analytics team, is a team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy.

## 2. Prepare

The data set used is the FitBit Fitness Tracker Data (https://www.kaggle.com/datasets/arashnic/fitbit) (CC0:Public Domain, dataset made available through Mobius (https://www.kaggle.com/arashnic)), as sugested by Sršen.This Kaggle data set contains personal fitness tracker from thirty fitbit users. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and heart rate that can be used to explore users' habits.

```
# importing libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(ggthemes)
library(ggpubr)
```

Importing and having an overview of the data

```
# dailyActivity
dailyActivity <- read.csv("dailyActivity_merged.csv")
glimpse(dailyActivity)
```

```
## 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~
```

```
# sleepDay
sleepDay <- read.csv("sleepDay_merged.csv")
glimpse(sleepDay)
```

```
## Rows: 413
## Columns: 5
## $ Id                 <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150~
## $ SleepDay           <chr> "4/12/2016 12:00:00 AM", "4/13/2016 12:00:00 AM", "~
## $ TotalSleepRecords  <int> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ TotalMinutesAsleep <int> 327, 384, 412, 340, 700, 304, 360, 325, 361, 430, 2~
## $ TotalTimeInBed     <int> 346, 407, 442, 367, 712, 320, 377, 364, 384, 449, 3~
```

```
# weightLogInfo
weightLogInfo <- read.csv("weightLogInfo_merged.csv")
glimpse(weightLogInfo)
```

```
## Rows: 67
## Columns: 8
## $ Id             <dbl> 1503960366, 1503960366, 1927972279, 2873212765, 2873212~
## $ Date           <chr> "5/2/2016 11:59:59 PM", "5/3/2016 11:59:59 PM", "4/13/2~
## $ WeightKg       <dbl> 52.6, 52.6, 133.5, 56.7, 57.3, 72.4, 72.3, 69.7, 70.3, ~
## $ WeightPounds   <dbl> 115.9631, 115.9631, 294.3171, 125.0021, 126.3249, 159.6~
## $ Fat            <int> 22, NA, NA, NA, NA, 25, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ BMI            <dbl> 22.65, 22.65, 47.54, 21.45, 21.69, 27.45, 27.38, 27.25,~
## $ IsManualReport <chr> "True", "True", "False", "True", "True", "True", "True"~
## $ LogId          <dbl> 1.462234e+12, 1.462320e+12, 1.460510e+12, 1.461283e+12,~
```

The data set is in long format, i.e., each row is one time point per Id, so each Id has data in multiple rows.

All variables of a data set can be showed with the function colnames(), from the dplyr package such as below:

```
# dailyActivity
colnames(dailyActivity)
```

```
##  [1] "Id"                       "ActivityDate"
##  [3] "TotalSteps"               "TotalDistance"
##  [5] "TrackerDistance"          "LoggedActivitiesDistance"
##  [7] "VeryActiveDistance"       "ModeratelyActiveDistance"
##  [9] "LightActiveDistance"      "SedentaryActiveDistance"
## [11] "VeryActiveMinutes"        "FairlyActiveMinutes"
## [13] "LightlyActiveMinutes"     "SedentaryMinutes"
## [15] "Calories"
```

```
# sleepDay
colnames(sleepDay)
```

```
## [1] "Id"                 "SleepDay"           "TotalSleepRecords"
## [4] "TotalMinutesAsleep" "TotalTimeInBed"
```

```
# weightLogInfo
colnames(weightLogInfo)
```

```
## [1] "Id"             "Date"           "WeightKg"       "WeightPounds"
## [5] "Fat"            "BMI"            "IsManualReport" "LogId"
```

Some meta data from the data sets are showed in the table below

```
data.frame(data_set = c("dailyActivity", "sleepDay", "weightLogInfo"),
                       no_of_rows = c(nrow(dailyActivity), nrow(sleepDay), nrow(weightLogInf
o)),
                       no_of_distinct_rows = c(n_distinct(dailyActivity), n_distinct(sleepDa
y),
                                               n_distinct(weightLogInfo)),
                       no_of_columns = c(ncol(dailyActivity), ncol(sleepDay), ncol(weightLogI
nfo)),
                       no_of_distinct_Ids = c(n_distinct(dailyActivity$Id), n_distinct(sleepD
ay$Id),
                                              n_distinct(weightLogInfo$Id)))
```

```
##        data_set no_of_rows no_of_distinct_rows no_of_columns no_of_distinct_Ids
## 1 dailyActivity        940                940            15                33
## 2      sleepDay        413                410             5                24
## 3 weightLogInfo         67                 67             8                 8
```

As we can see, there are three duplicate observations in the sleepDay data set, once we have 413 number of rows and 410 number of distinct rows.

The chunk code below shows all duplicate in the sleepDay data set:

```
duplicate_rows <- sleepDay %>%
  filter(duplicated(sleepDay))
duplicate_rows
```

```
##           Id              SleepDay TotalSleepRecords TotalMinutesAsleep
## 1 4388161847  5/5/2016 12:00:00 AM                 1                471
## 2 4702921684  5/7/2016 12:00:00 AM                 1                520
## 3 8378563200 4/25/2016 12:00:00 AM                 1                388
##   TotalTimeInBed
## 1            495
## 2            543
## 3            402
```

Another conclusion that we can have is that although it was said that 30 participants attended the research, there were found 33 distinct IDs in the dailyActivity data set, therefore, more than we previously thought, but not all of them attending the complete research because we have 24 for distinct IDs in the sleepDay data set and only 8 in the weighLogInfo data set.

In order to check the credibility and integrity of the data, the method **ROCCC** will be applied:

1. **Reliability**: These data do not seem to be reliable, as the sample is too small (30 users only). Once the entire population of smart devices users should be much bigger, the margin of error is very big for an acceptable confidence level.
2. **Originality**: The data set is not original as it is a third-party data, i.e., data provided from outside sources who did not collect it directly.
3. **Comprehensiveness**: No information about the sample in the research is given, such as age or gender of the participants, so we cannot state that the data is comprehensive enough.

4. **Current**: These data were collected in the first half of 2016, so it is been six years. Although some things may have changed, especially related to new technologies implemented in smart devices, we can consider the data is not too old and may still reflect the present.

5. **Cited**: No information about the credibility of the data is given besides it was created by Amazon Mechanical Murk, so we cannot state that the data is cited.

Because of the data's integrity and credibility, it may be not possible to provide reliable and compreensive analisys to Bellabeat's executive team, and just a direction to new researchs to be taken in the future. All the caveats will be clearly exposed in the following sections.

## 3. Process

Now it is time to do some data cleaning!

To get started, I will remove the duplicate rows in the sleepDay data set:

```
sleepDay2 <- unique(sleepDay)

# checking if all rows in sleepDay2 are unique
nrow(sleepDay2) == n_distinct(sleepDay2)
```

```
## [1] TRUE
```

To avoid skewed results, I will assume that is unreasonable a person to take 0 steps and walk 0 meters during the day, so I will filter out these cases.

```
dailyActivity2 <- dailyActivity %>% filter(TotalSteps != 0, TotalDistance != 0)
nrow(dailyActivity2)
```

```
## [1] 862
```

And now we have 862 observations in the dailyActivity2 data set.

In the three data sets, the date columns are formatted as "character", So a new column will be created in each one as "date" and other with the weekdays as well

```
# setting location
Sys.setlocale("LC_TIME", "C")
```

```
## [1] "C"
```

```
# dailyActivity
dailyActivity3 <- dailyActivity2 %>%
  mutate(Date = mdy(ActivityDate), DayOfWeek = weekdays(Date)) %>%
  select(-c(2)) %>%
  relocate(Date, .after = 1) %>%
  relocate(DayOfWeek, .after = 2)
glimpse(dailyActivity3)
```

```
## Rows: 862
## Columns: 16
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ Date                     <date> 2016-04-12, 2016-04-13, 2016-04-14, 2016-04-~
## $ DayOfWeek                <chr> "Tuesday", "Wednesday", "Thursday", "Friday",~
## $ 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~
```

```
# sleepDay
sleepday3 <- sleepDay2 %>%
  separate(SleepDay, c("Date", "Time"), sep = " ") %>%
  mutate(Date = mdy(Date), DayOfWeek = weekdays(Date)) %>%
  select(-c(3)) %>%
  relocate(DayOfWeek, .after = 2)
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 410 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
glimpse(sleepday3)
```

```
## Rows: 410
## Columns: 6
## $ Id                 <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150~
## $ Date               <date> 2016-04-12, 2016-04-13, 2016-04-15, 2016-04-16, 20~
## $ DayOfWeek          <chr> "Tuesday", "Wednesday", "Friday", "Saturday", "Sund~
## $ TotalSleepRecords  <int> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ TotalMinutesAsleep <int> 327, 384, 412, 340, 700, 304, 360, 325, 361, 430, 2~
## $ TotalTimeInBed     <int> 346, 407, 442, 367, 712, 320, 377, 364, 384, 449, 3~
```

```
# weightLogInfo
weightLogInfo2 <- weightLogInfo %>%
  separate(Date, c("Date", "Time"), sep = " ") %>%
  mutate(Date = mdy(Date), DayOfWeek = weekdays(Date)) %>%
  select(-c(3)) %>%
  relocate(DayOfWeek, .after = 2)
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 67 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
glimpse(weightLogInfo2)
```

```
## Rows: 67
## Columns: 9
## $ Id             <dbl> 1503960366, 1503960366, 1927972279, 2873212765, 2873212~
## $ Date           <date> 2016-05-02, 2016-05-03, 2016-04-13, 2016-04-21, 2016-0~
## $ DayOfWeek      <chr> "Monday", "Tuesday", "Wednesday", "Thursday", "Thursday~
## $ WeightKg       <dbl> 52.6, 52.6, 133.5, 56.7, 57.3, 72.4, 72.3, 69.7, 70.3, ~
## $ WeightPounds   <dbl> 115.9631, 115.9631, 294.3171, 125.0021, 126.3249, 159.6~
## $ Fat            <int> 22, NA, NA, NA, NA, 25, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ BMI            <dbl> 22.65, 22.65, 47.54, 21.45, 21.69, 27.45, 27.38, 27.25,~
## $ IsManualReport <chr> "True", "True", "False", "True", "True", "True", "True"~
## $ LogId          <dbl> 1.462234e+12, 1.462320e+12, 1.460510e+12, 1.461283e+12,~
```

Merging the dailyActivity an sleepday data sets

```
dailyActivitySleepDay_merged <- merge(dailyActivity3, sleepday3, by = c("Id", "Date")) %>%
  select(Id, Date, TotalSteps, TotalMinutesAsleep, VeryActiveMinutes, FairlyActiveMinutes, Li
ghtlyActiveMinutes, SedentaryMinutes)

dailyActivitySleepDay_merged2 <- dailyActivitySleepDay_merged %>%
  group_by(Id) %>%
  summarise(meanTotalSteps = mean(TotalSteps), meanVeryActiveMinutes = mean(VeryActiveMinute
s), meanFairlyActiveMinutes = mean(FairlyActiveMinutes), meanLightlyActiveMinutes = mean(Ligh
tlyActiveMinutes), meanSedentaryMinutes = mean(SedentaryMinutes), meanTotalMinutesAsleep = me
an(TotalMinutesAsleep))
```

## 4. Analyse

I will now get some statistical summary of the each data set

```
# dailyActivity2
dailyActivity2 %>%
  select(TotalSteps, TotalDistance,
         VeryActiveDistance, ModeratelyActiveDistance, LightActiveDistance, SedentaryActiveDi
stance,
         VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, SedentaryMinutes,
         Calories) %>%
  summary()
```

```
##    TotalSteps    TotalDistance    VeryActiveDistance ModeratelyActiveDistance
##  Min.   :    8   Min.   : 0.010   Min.   : 0.000     Min.   :0.0000
##  1st Qu.: 4927   1st Qu.: 3.373   1st Qu.: 0.000     1st Qu.:0.0000
##  Median : 8054   Median : 5.590   Median : 0.410     Median :0.3100
##  Mean   : 8329   Mean   : 5.986   Mean   : 1.639     Mean   :0.6189
##  3rd Qu.:11096   3rd Qu.: 7.905   3rd Qu.: 2.277     3rd Qu.:0.8675
##  Max.   :36019   Max.   :28.030   Max.   :21.920     Max.   :6.4800
##  LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
##  Min.   : 0.000      Min.   :0.000000        Min.   :  0.00
##  1st Qu.: 2.350      1st Qu.:0.000000        1st Qu.:  0.00
##  Median : 3.580      Median :0.000000        Median :  7.00
##  Mean   : 3.643      Mean   :0.001752        Mean   : 23.04
##  3rd Qu.: 4.897      3rd Qu.:0.000000        3rd Qu.: 35.00
##  Max.   :10.710      Max.   :0.110000        Max.   :210.00
##  FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes    Calories
##  Min.   :  0.00      Min.   :  0.0        Min.   :   0.0   Min.   :  52
##  1st Qu.:  0.00      1st Qu.:147.0        1st Qu.: 721.2   1st Qu.:1857
##  Median :  8.00      Median :208.5        Median :1020.5   Median :2220
##  Mean   : 14.79      Mean   :210.3        Mean   : 955.2   Mean   :2362
##  3rd Qu.: 21.00      3rd Qu.:272.0        3rd Qu.:1189.0   3rd Qu.:2832
##  Max.   :143.00      Max.   :518.0        Max.   :1440.0   Max.   :4900
```

```
# sleepDay2
sleepDay2 %>%
  select(TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed) %>%
  summary()
```

```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.00      Min.   : 58.0      Min.   : 61.0
##  1st Qu.:1.00      1st Qu.:361.0      1st Qu.:403.8
##  Median :1.00      Median :432.5      Median :463.0
##  Mean   :1.12      Mean   :419.2      Mean   :458.5
##  3rd Qu.:1.00      3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :3.00      Max.   :796.0      Max.   :961.0
```

```
# weightLogInfo
weightLogInfo %>%
  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
```

As shown, in the dailyActivity2 data set only a few observations have a very activity routine in comparison with the entire, once the median and mean of VeryActiveDistance and VeryActiveMinutes is too big in percent as well the coefficient of variation.

```
data.frame(cv_VeryActiveDistance = percent(sd(dailyActivity2$VeryActiveDistance) / mean(daily
Activity2$VeryActiveDistance)), cv_VeryActiveMinutes = percent(sd(dailyActivity2$VeryActiveMi
nutes) / mean(dailyActivity2$VeryActiveMinutes)))
```

```
##   cv_VeryActiveDistance cv_VeryActiveMinutes
## 1                  167%                 146%
```

On the other hand the coefficient of variation of the light activity routine is smaller, that means the sample trends to have this kind of routine everyday.

```
data.frame(cv_LightActiveDistance = percent(sd(dailyActivity2$LightActiveDistance) / mean(dai
lyActivity2$LightActiveDistance)), cv_LightlyActiveMinutes = percent(sd(dailyActivity2$Lightl
yActiveMinutes) / mean(dailyActivity2$LightlyActiveMinutes)))
```

```
##   cv_LightActiveDistance cv_LightlyActiveMinutes
## 1                    51%                    46%
```

In the sleepDay2 data set it is possible to see all participants sleeps at least once a day, and three times at most.

In addition, in the weightLogInfo data set, the BMI stands on around 25, considered normal for adults both male and female. Unfortunately only two records of fat are available, so it is not possible to make any analysis of this metric.

```
weightLogInfo %>%
  count(is.na(Fat))
```

```
##   is.na(Fat)  n
## 1      FALSE  2
## 2       TRUE 65
```

It is possible to see that during the research period, there were no significant changes in the weight of the participants, where the average variation was less than 0.1% as well the BMI variation.

```
variationOfWeightKg <- weightLogInfo %>%
  group_by(Id) %>%
  mutate(firstWeightKg =  WeightKg[Date == min(Date)],
         lastWeightKg = WeightKg[Date == max(Date)],
         weightVariation = (lastWeightKg - firstWeightKg) / firstWeightKg,
         firstBMI = BMI[Date == min(Date)],
         lastBMI = BMI[Date == max(Date)],
         bmiVariation = (lastBMI - firstBMI) / firstBMI) %>%
  select(Id, firstWeightKg, lastWeightKg, weightVariation, firstBMI, lastBMI, bmiVariation)

variationOfWeightKg2 <- unique(variationOfWeightKg)

variationOfWeightKg2
```

```
## # A tibble: 8 x 7
## # Groups:   Id [8]
##           Id firstWeightKg lastWeightKg weightVariation first~1 lastBMI bmiVar~2
##        <dbl>         <dbl>        <dbl>           <dbl>   <dbl>   <dbl>    <dbl>
## 1 1503960366          52.6         52.6         0          22.6    22.6  0
## 2 1927972279         134.         134.          0          47.5    47.5  0
## 3 2873212765          56.7         57.3         0.0106     21.5    21.7  0.0112
## 4 4319703577          72.4         72.3        -0.00138    27.5    27.4 -0.00255
## 5 4558609924          69.7         69.1        -0.00861    27.2    27   -0.00917
## 6 5577150313          90.7         90.7         0          28      28    0
## 7 6962181067          62.5         62.4        -0.00160    24.4    24.4 -0.00164
## 8 8877689391          85.8         85.5        -0.00350    25.7    25.6 -0.00273
## # ... with abbreviated variable names 1: firstBMI, 2: bmiVariation
```

```
data.frame(metric = c("weightVariation", "bmiVariation"),
           meanValue = percent(c(mean(variationOfWeightKg2$weightVariation),
                                 mean(variationOfWeightKg2$bmiVariation))))
```

```
##            metric meanValue
## 1 weightVariation  -0.0563%
## 2    bmiVariation  -0.0613%
```

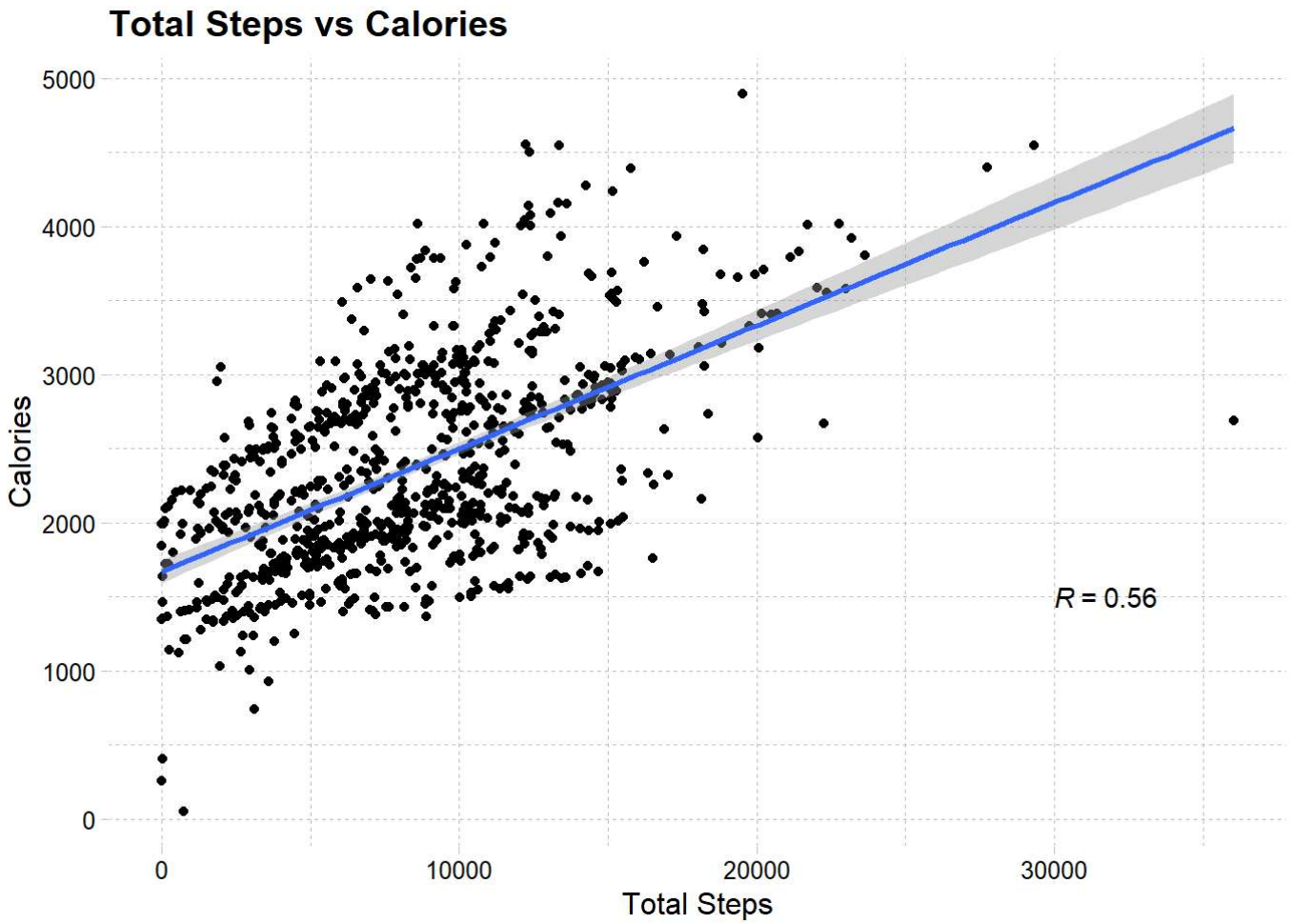
## 5. Share

The goal now is to support our analysis through visualizations that will help to find insights that can help the stakeholders make better decisions.

To get started, I will check the relationship between total steps and calories burned:

```
ggplot(data = dailyActivity3, mapping = aes(x = TotalSteps, y = Calories)) +
  geom_point() +
  theme_pander(base_size = 12) +
  geom_smooth(method = "lm") +
  stat_cor(aes(label = ..r.label..), r.accuracy = 0.01, label.x = 30000, label.y = 1500) +
  labs(title = "Total Steps vs Calories", x = "Total Steps")
```

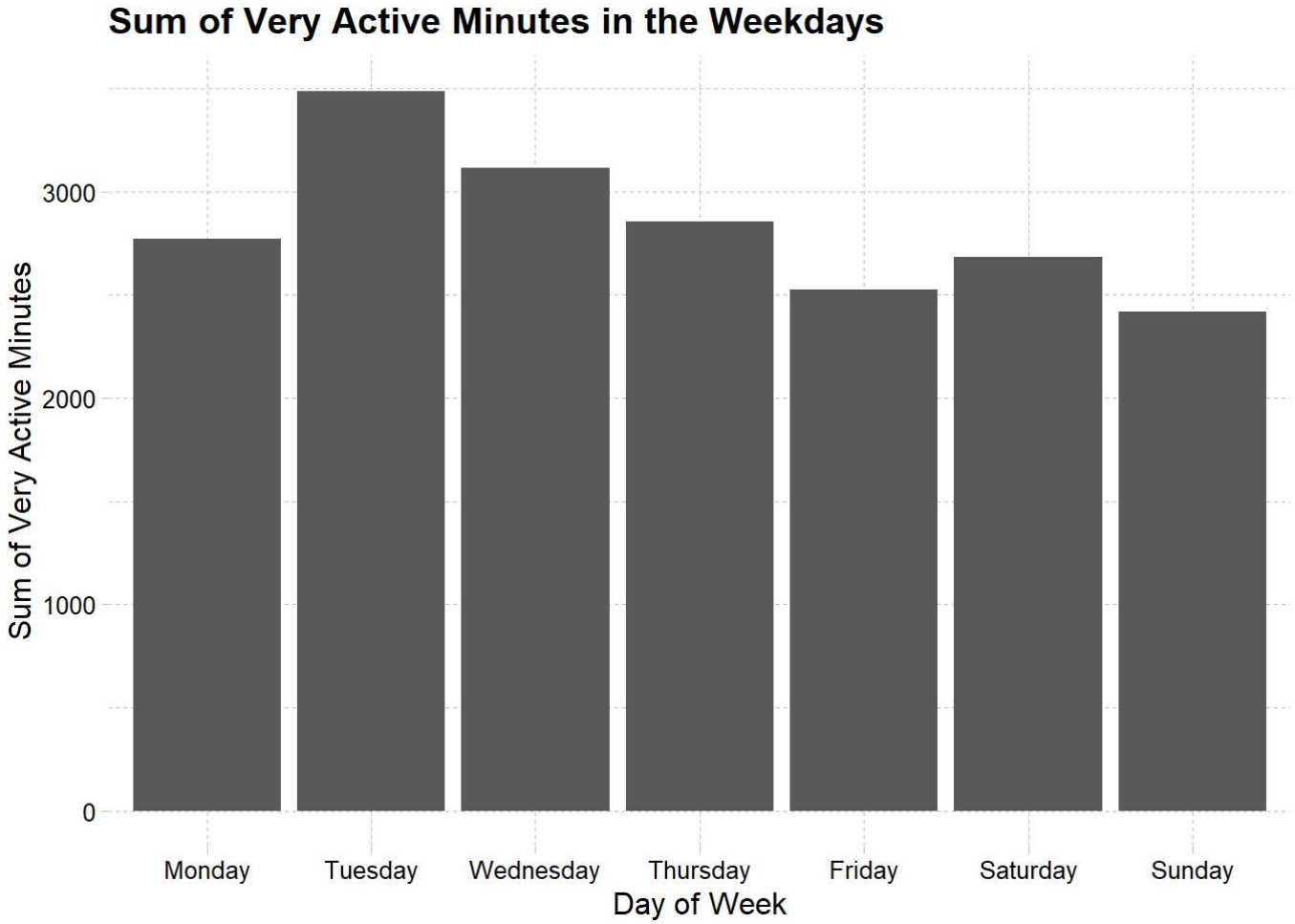
```
## `geom_smooth()` using formula 'y ~ x'
```




In the chart above we see a positive correlation between the total steps and calories burned, supported for a correlation coefficient of 0.56.

Assessing the more activity days:

```
mostActiveDays <- dailyActivity3 %>%
  group_by(DayOfWeek) %>%
  summarise(VeryActiveMinutes = sum(VeryActiveMinutes))

# Ordering the days
mostActiveDays$DayOfWeek <- factor(mostActiveDays$DayOfWeek, levels = c("Monday", "Tuesday",
"Wednesday",
                                                                        "Thursday","Friday",
"Saturday",
                                                                        "Sunday"))

ggplot(data = mostActiveDays, mapping = aes(x = DayOfWeek, y = VeryActiveMinutes)) +
  geom_col() +
  theme_pander(base_size = 12) +
  labs(title = "Sum of Very Active Minutes in the Weekdays", x = "Day of Week", y = "Sum of V
ery Active Minutes")
```

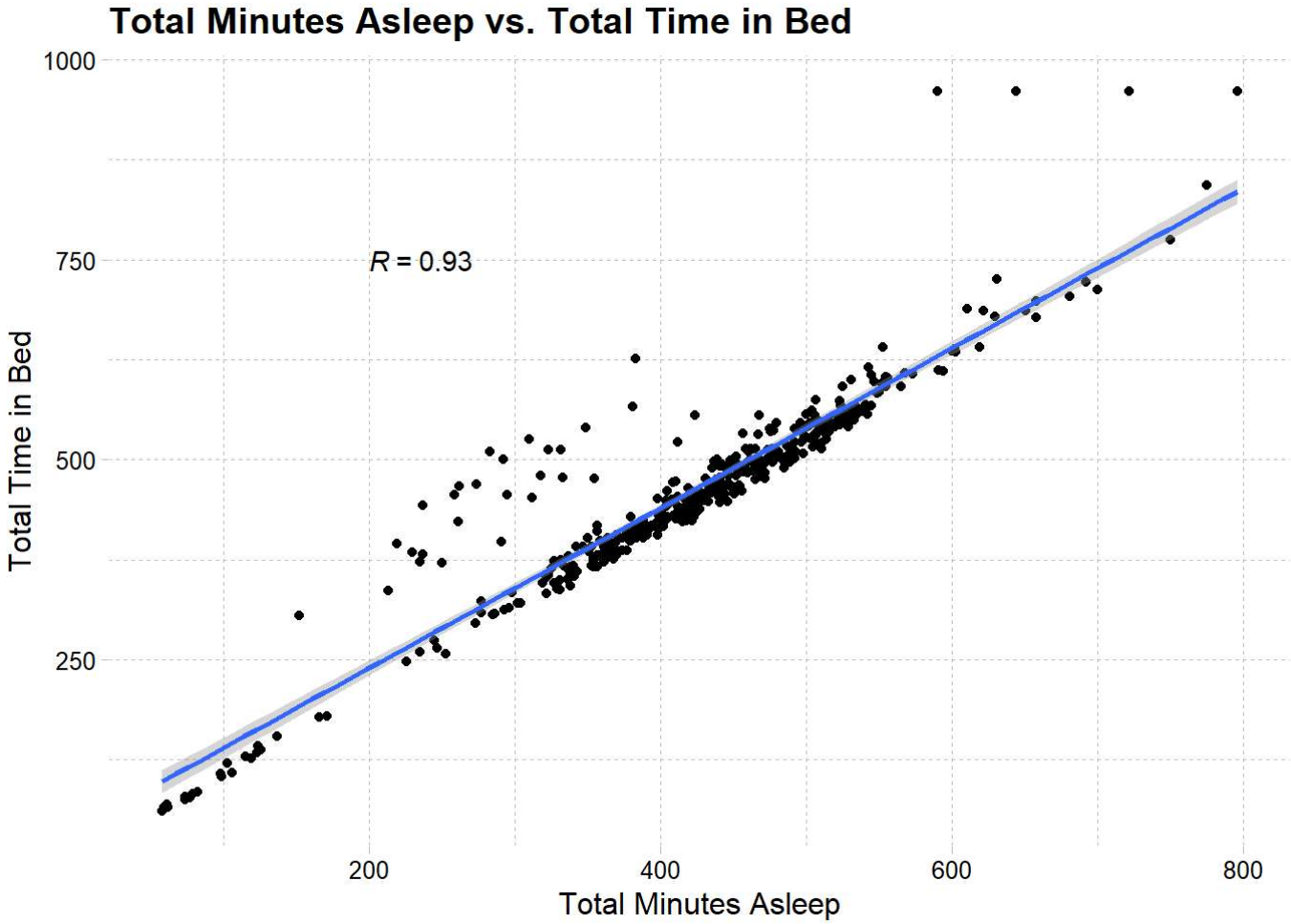



The chart above shows us that the most activity days are in the middle of the week, from Tuesday to Thursday and at the weekend the sample trends to reduce its activity.

Assessing the relationship between the sleep minutes and the total time in bed:

```
ggplot(sleepday3, mapping = aes(x = TotalMinutesAsleep, y = TotalTimeInBed)) +
  geom_point() +
  geom_smooth(method = "lm") +
  stat_cor(aes(label = ..r.label..), r.accuracy = 0.01, label.x = 200, label.y = 750) +
  theme_pander(base_size = 12) +
  labs(title = "Total Minutes Asleep vs. Total Time in Bed", x = "Total Minutes Asleep", y =
"Total Time in Bed")
```

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



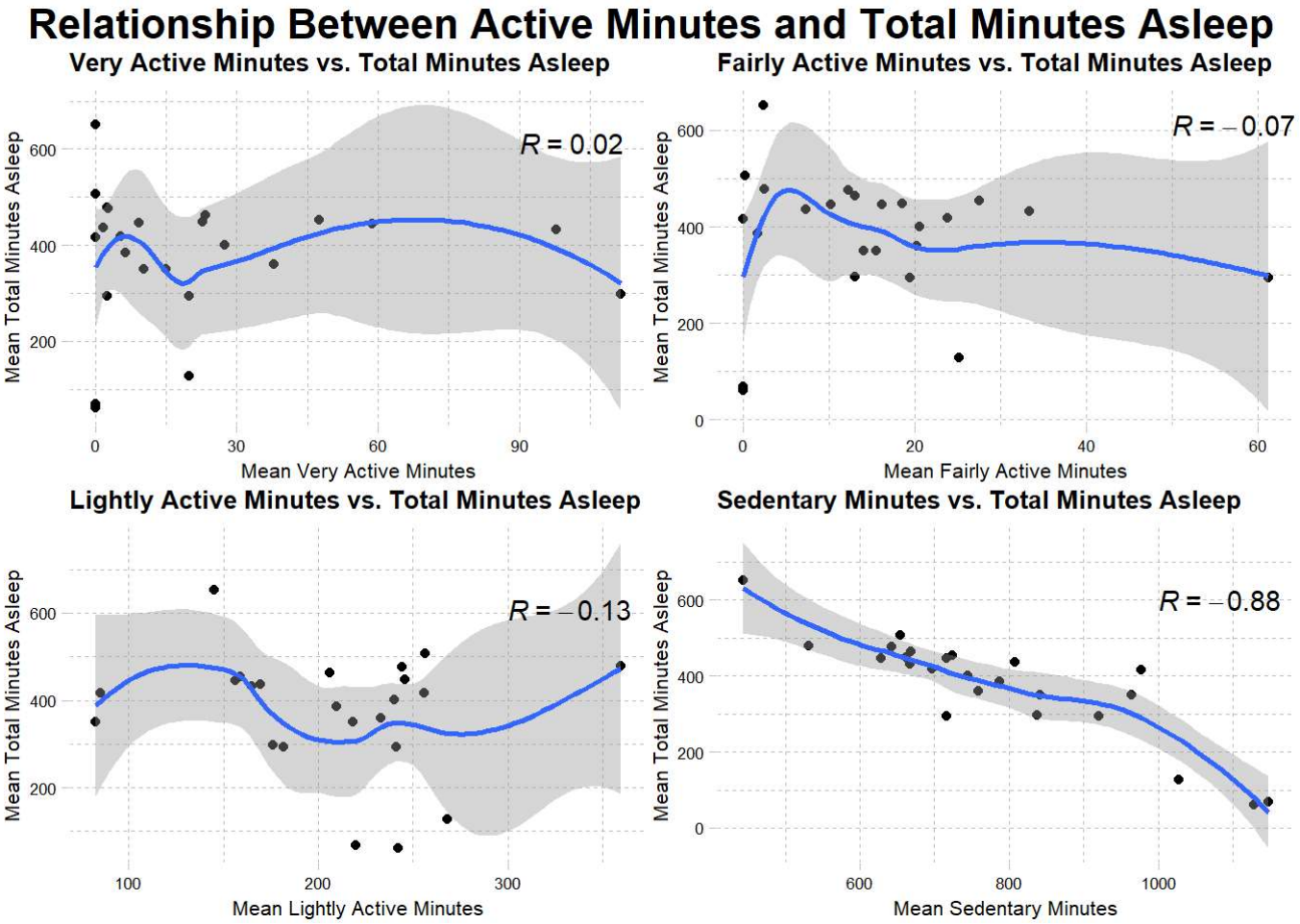

Again a clearly positive correlation between minutes asleep and time in bed.

Assessing the relationship between very active minutes and minutes asleep

```
plotVeryActiveMinutes <- ggplot(dailyActivitySleepDay_merged2, mapping = aes(x = meanVeryActi
veMinutes,
                                                                             y = meanTotalMin
utesAsleep)) +
  geom_point() +
  geom_smooth() +
  theme_pander(base_size = 8) +
  labs(title = "Very Active Minutes vs. Total Minutes Asleep", x = "Mean Very Active Minutes"
, y = "Mean Total Minutes Asleep") +
  stat_cor(aes(label = ..r.label..), r.accuracy = 0.01, label.x = 90)

plotFairlyActiveMinutes <- ggplot(dailyActivitySleepDay_merged2, mapping = aes(x = meanFairly
ActiveMinutes,
                                                                               y = meanTotalM
inutesAsleep)) +
  geom_point() +
  geom_smooth() +
  theme_pander(base_size = 8) +
  labs(title = "Fairly Active Minutes vs. Total Minutes Asleep", x = "Mean Fairly Active Minu
tes", y = "Mean Total Minutes Asleep") +
  stat_cor(aes(label = ..r.label..), r.accuracy = 0.01, label.x = 50)

plotLightlyActiveMinutes <- ggplot(dailyActivitySleepDay_merged2, mapping = aes(x = meanLight
lyActiveMinutes,
                                                                                y = meanTotal
MinutesAsleep)) +
  geom_point() +
  geom_smooth() +
  theme_pander(base_size = 8) +
  labs(title = "Lightly Active Minutes vs. Total Minutes Asleep", x = "Mean Lightly Active Mi
nutes", y = "Mean Total Minutes Asleep") +
  stat_cor(aes(label = ..r.label..), r.accuracy = 0.01, label.x = 300)

plotSedendatryMinutes <- ggplot(dailyActivitySleepDay_merged2, mapping = aes(x = meanSedentar
yMinutes,
                                                                             y = meanTotal
MinutesAsleep)) +
  geom_point() +
  geom_smooth() +
  theme_pander(base_size = 8) +
  labs(title = "Sedentary Minutes vs. Total Minutes Asleep", x = "Mean Sedentary Minutes", y
 = "Mean Total Minutes Asleep") +
  stat_cor(aes(label = ..r.label..), r.accuracy = 0.01, label.x = 1000)

arrangedPlot <- ggarrange(plotVeryActiveMinutes, plotFairlyActiveMinutes, plotLightlyActiveMi
nutes, plotSedendatryMinutes)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
annotate_figure(arrangedPlot, top = text_grob("Relationship Between Active Minutes and Total
 Minutes Asleep",
                                               face = "bold", size = 16))
```




In the charts above, we see as the activity minutes influence the quality of sleep. Clearly the more sedentary, less minutes sleeping.

## 6. Act

As stated before, the data must be more comprehensive in order to provide a complete analysis and then a better data-driven decision making.

Bellabeat products focus on women, so only this audience should be considered, but we do not have information about the gender of the participants. Another important feature that must be included is the age of the audience, as it can directly affect the routine and metabolism.

Anyway, we can still do some recommendations to Bellabeat executive team:

1. **Goal** - The users must set a goal when using Bellabeat smart devices, like losing weight or sleep better. Based on it, it would be easier to set metrics to achieve their goals.
2. **Notifications** - The users could receive notifications to track their goals, for example how many steps are left to burn “x” calories.

And finally, as a top high-level insight, it is pretty clear that keeping an active routine is the best choice anyone can make, whether it is to sleep better or stay in shape. There are only benefits. So Bellabeat must keep their users always motivated, showing theirs achievements and giving feedback.