# Google Data Analytics Capstone

Marko Mršić

2022-08-05

# Case Study: How can a wellness company play it smart?

## Introduction

In this case study, I'm a junior data analyst working on the marketing analyst team at Bellabeat, a high-tech manufacturer of health-focused products for women. Bellabeat is a successful small company, but they have the potential to become a larger player in the global smart device market.

Urška Sršen, co-founder and Chief Creative Officer of Bellabeat, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company. 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 then help guide the marketing strategy for the company. I will present my analysis to the Bellabeat executive team along with my high-level recommendations for Bellabeat's marketing strategy.

Some of the Bellabeat's most popular products:

1. The Bellabeat app provides users with health data related to their activity, sleep, stress, menstrual cycle, and mindfulness habits. This data can help users better understand their current habits and make healthy decisions.
2. Leaf is Bellabeat's classic wellness tracker can be worn as a bracelet, necklace, or clip.
3. Time is wellness watch combines the timeless look of a classic timepiece with smart technology to track user activity, sleep, and stress.
4. Spring is a water bottle that tracks daily water intake using smart technology to ensure that you are appropriately hydrated throughout the day.

## 1. Ask

Sršen asks me to analyze smart device usage data to gain insight into how consumers use non-Bellabeat smart devices. She then wants me to select one Bellabeat product to apply these insights to in my presentation.

**Identifying the business tasks**

A business task is a question or problem data analysis answers for business. In this case study I want to find answers to these questions:

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

**Addressing the key 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: A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy.

## 2. Prepare

**Data source**

In this case study, I use publicly available data that I downloaded from Kaggle as suggested by the Bellabeat team. It is Fitbit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius). This 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. Data is publicly available on Kaggle: FitBit Fitness Tracker Data and stored in 18 csv files. The data is stored on my personal computer.

**Data credibility check**

Good data sources are ROCCC - reliable, original, comprehensive, current, and cited. This cannot be said for this dataset, because a quick data analysis (filtering and sorting in spreadsheets) shows that the data was collected in 2016, the sample is only 30 people, which certainly cannot be representative of the entire population of users of Fitbit products. The data is not even original because it is a third party source (Amazon Mechanical Turk). The dataset is comprehensive because it contains information that should be sufficient to provide answers to key questions. Bellabeat is aware that this dataset has its limitations. In any case, at the end of the analysis, all of the above should be taken into account.

## 3. Process

In this phase, the task is to check the integrity of the data, that is, to carry out the process of cleaning and transforming the data to ensure its integrity.

For this I will use the R programming language, that is, RStudio.

First, I will load the necessary R packages.

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6     ✔ purrr   0.3.4
## ✔ tibble  3.1.7     ✔ dplyr   1.0.9
## ✔ tidyr   1.2.0     ✔ stringr 1.4.0
## ✔ readr   2.1.2     ✔ forcats 0.5.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(dplyr)
library(skimr)
library(lubridate)
```

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

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

After that, I will import the necessary CSV documents using the read.csv command. Although there are 18 CSV documents in the dataset, after analysis in spreadsheets, I found that we need only three documents: dailyActivity_merged, sleepDay_merged, and weightLogInfo_merged. Namely, these three tables contain all the data found in the other tables.

```
setwd("C:/Users/marko/Desktop/Case Study & Portfolio/Data")
daily_activity <- read.csv("dailyActivity_merged.csv")
daily_sleep <- read.csv("sleepDay_merged.csv")
weight_log <- read.csv("weightLogInfo_merged.csv")
```

Next, I will use the glimpse function which returns a summary of the data frame, including the number of columns and rows.

```
glimpse(daily_activity)
```

```
## 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…
```

```
glimpse(daily_sleep)
```

```
## 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, 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…
```

```
glimpse(weight_log)
```

```
## 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 Head function will give us a brief insight into each of these tables.

```
head(daily_activity)
```

```
##           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
```

```
head(daily_sleep)
```

```
##           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
```

```
head(weight_log)
```

```
##           Id                  Date WeightKg WeightPounds Fat   BMI
## 1 1503960366  5/2/2016 11:59:59 PM     52.6     115.9631  22 22.65
## 2 1503960366  5/3/2016 11:59:59 PM     52.6     115.9631  NA 22.65
## 3 1927972279  4/13/2016 1:08:52 AM    133.5     294.3171  NA 47.54
## 4 2873212765 4/21/2016 11:59:59 PM     56.7     125.0021  NA 21.45
## 5 2873212765 5/12/2016 11:59:59 PM     57.3     126.3249  NA 21.69
## 6 4319703577 4/17/2016 11:59:59 PM     72.4     159.6147  25 27.45
##   IsManualReport        LogId
## 1           True 1.462234e+12
## 2           True 1.462320e+12
## 3          False 1.460510e+12
## 4           True 1.461283e+12
## 5           True 1.463098e+12
## 6           True 1.460938e+12
```

Using these functions, I determined the number of columns and rows and the data type for each column. The only problem is that the data type for date columns is a character, which I have to change to the DateTime data type.

```
daily_activity$Rec_Date <- as.Date(daily_activity$ActivityDate,"%m/%d/%y")
head(daily_activity)
```

```
##           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   Rec_Date
## 1                  13                  328              728     1985 2020-04-12
## 2                  19                  217              776     1797 2020-04-13
## 3                  11                  181             1218     1776 2020-04-14
## 4                  34                  209              726     1745 2020-04-15
## 5                  10                  221              773     1863 2020-04-16
## 6                  20                  164              539     1728 2020-04-17
```

```
daily_activity$month <- format(daily_activity$Rec_Date,"%B")
daily_activity$day_of_week <- format(daily_activity$Rec_Date,"%A")
```

The next step is to see how many unique IDs we have, that is, how many respondents participated in the survey. I found that there are 33 unique IDs in the daily_activity table, in the daily-sleep 24, and in weight_log only 8, which automatically calls into question the reliability of the adoption conclusions on such a small sample.

```
n_distinct(daily_activity$Id)
```

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

```
n_distinct(daily_sleep$Id)
```

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

```
n_distinct(weight_log$Id)
```

```
## [1] 8
```

The following is to determine if there are duplicates in our tables and if there are, to remove them.

```
sum(duplicated(daily_activity))
```

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

```
sum(duplicated(daily_sleep))
```

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

```
sum(duplicated(weight_log))
```

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

There are three duplicates in the daily_sleep table that we will remove using distinct function.

```
daily_sleep <- daily_sleep %>%
  distinct()
```

## 4. Analyze and 5. Share

In this phase, the goal is to gain additional insight into the data through aggregation and calculations and to identify trends and relationships.

Also, I will use different types of charts to create visualizations to help spot relationships and trends.

I will start by summarizing the data.

The summarized data shows that the user of the Fitbit application takes an average of 7,638 steps in one day, which is below the recommended level of 10,000 steps. Also, the average distance a user walks is about 5.5 kilometers. On average, the respondent sits for about 991 minutes a day, which is about 16.5 hours, which is significantly more than 10 hours, which is the upper level that is considered harmful to health. The respondents have an average of 21 minutes of active physical activity per day, which is also below the recommended 30 minutes or 3 hours per week.

```
daily_activity %>%
  select(TotalSteps,TotalDistance,SedentaryMinutes,VeryActiveMinutes, Calories) %>%
  summary()
```

```
##    TotalSteps    TotalDistance    SedentaryMinutes VeryActiveMinutes
##  Min.   :    0   Min.   : 0.000   Min.   :   0.0   Min.   :  0.00
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 729.8   1st Qu.:  0.00
##  Median : 7406   Median : 5.245   Median :1057.5   Median :  4.00
##  Mean   : 7638   Mean   : 5.490   Mean   : 991.2   Mean   : 21.16
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:1229.5   3rd Qu.: 32.00
##  Max.   :36019   Max.   :28.030   Max.   :1440.0   Max.   :210.00
##     Calories
##  Min.   :   0
##  1st Qu.:1828
##  Median :2134
##  Mean   :2304
##  3rd Qu.:2793
##  Max.   :4900
```

The summarized data from the weight_log table shows that the average weight of the respondents is 72 kilograms, but since we do not have height data, this information does not mean much to us. The fact that the average Body Mass Index (BMI) of 25.19 means more to us. This means that on average the respondents are slightly overweight. However, the maximum BMI is 47.54, which is significantly above the recommended amount for both sexes.

```
weight_log %>%
  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
```

The daily_sleep table allows us to simply summarize the data and determine that the respondents sleep on average a little more than 6 hours per day, which is below the recommended 7 to 8 hours.

```
(sum(daily_sleep$TotalMinutesAsleep)/sum(daily_sleep$TotalSleepRecords))/60
```

```
## [1] 6.240414
```

### Creation of a pie chart

In the continuation of the analysis, I will use the available data in order to graphically display the activity of the respondents in terms of daily steps taken. According to the average number of steps, I will divide all respondents into four groups: weakly active with less than 5,000 steps, moderately active with 5,000 to 7,499 steps, active with 7,500 to 9,999, and very active with more than 10,000 steps per day.

First, we'll calculate how many average steps each Fitbit user takes per day.

```
daily_average <- daily_activity %>%
  group_by(Id) %>%
  summarise (average_daily_steps = mean(TotalSteps))

head(daily_average)
```

```
## # A tibble: 6 × 2
##           Id average_daily_steps
##        <dbl>               <dbl>
## 1 1503960366              12117.
## 2 1624580081               5744.
## 3 1644430081               7283.
## 4 1844505072               2580.
## 5 1927972279                916.
## 6 2022484408              11371.
```

After that, we will divide the users into the previously mentioned groups, according to the number of steps completed.

```
user_type <- daily_average %>%
  mutate(user_type = case_when(
    average_daily_steps < 5000 ~ "low active",
    average_daily_steps >= 5000 & average_daily_steps < 7500 ~ "somewhat active",
    average_daily_steps >= 7500 & average_daily_steps <10000 ~ "active",
    average_daily_steps >= 10000 ~ "very active"
  ))

head(user_type)
```

```
## # A tibble: 6 × 3
##           Id average_daily_steps user_type
##        <dbl>               <dbl> <chr>
## 1 1503960366              12117. very active
## 2 1624580081               5744. somewhat active
## 3 1644430081               7283. somewhat active
## 4 1844505072               2580. low active
## 5 1927972279                916. low active
## 6 2022484408              11371. very active
```

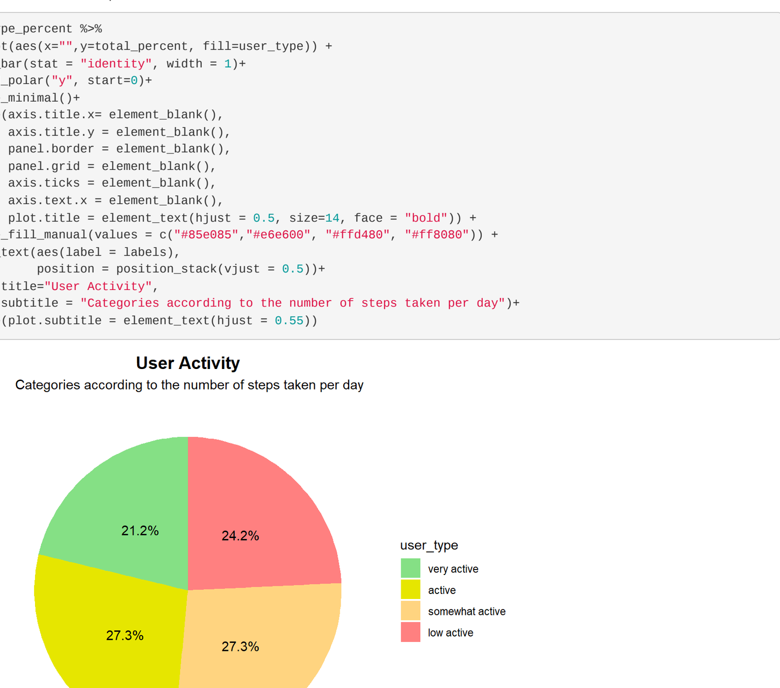
The following is the creation of a new table in which we will group users according to activity with the aim of easier display on the chart.

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

user_type_percent$user_type <- factor(user_type_percent$user_type , levels = c("very active", "active", "somewhat active", "low active"))

head(user_type_percent)
```

```
## # A tibble: 4 × 3
##   user_type       total_percent labels
##   <fct>                   <dbl> <chr>
## 1 active                  0.273 27.3%
## 2 low active              0.242 24.2%
## 3 somewhat active         0.273 27.3%
## 4 very active             0.212 21.2%
```

Next comes the creation of a pie chart.

```
user_type_percent %>%
  ggplot(aes(x="",y=total_percent, 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("#85e085","#e6e600", "#ffd480", "#ff8080")) +
  geom_text(aes(label = labels),
            position = position_stack(vjust = 0.5))+
  labs(title="User Activity",
       subtitle = "Categories according to the number of steps taken per day")+
  theme(plot.subtitle = element_text(hjust = 0.55))
```

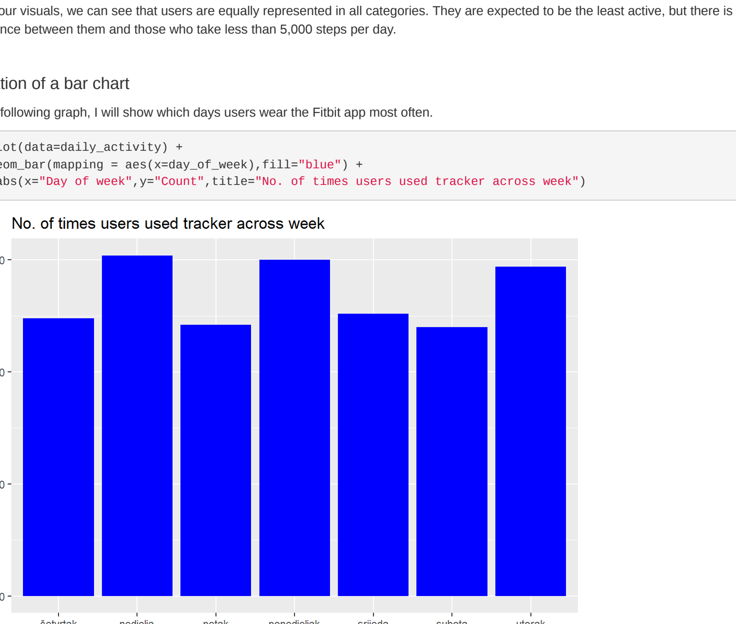



From our visuals, we can see that users are equally represented in all categories. They are expected to be the least active, but there is no big difference between them and those who take less than 5,000 steps per day.

### Creation of a bar chart

In the following graph, I will show which days users wear the Fitbit app most often.

```
ggplot(data=daily_activity) +
  geom_bar(mapping = aes(x=day_of_week),fill="blue") +
  labs(x="Day of week",y="Count",title="No. of times users used tracker across week")
```



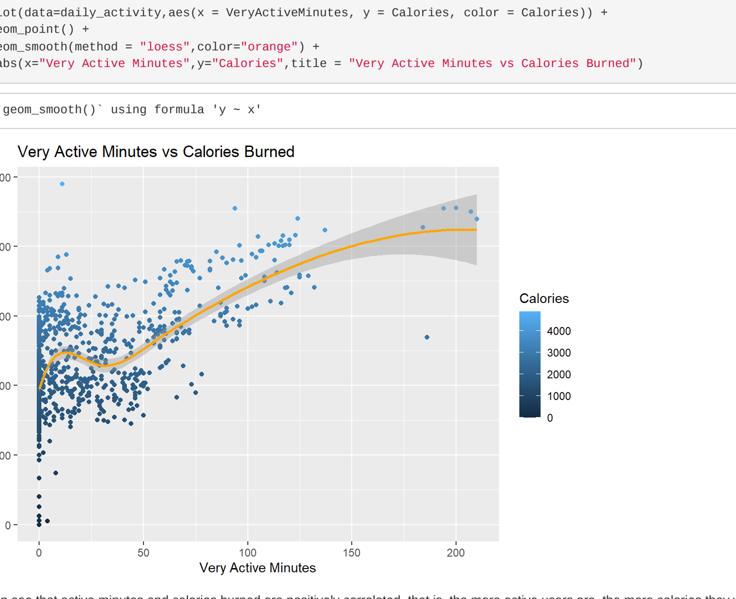

The days when the application is used the most are Sunday, Monday, and Tuesday.

### Creation of scatter plots

Furthermore, the following graph will show the relationship between the user's active minutes and calorie consumption.

```
ggplot(data=daily_activity,aes(x = VeryActiveMinutes, y = Calories, color = Calories)) +
  geom_point() +
  geom_smooth(method = "loess",color="orange") +
  labs(x="Very Active Minutes",y="Calories",title = "Very Active Minutes vs Calories Burned")
```

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

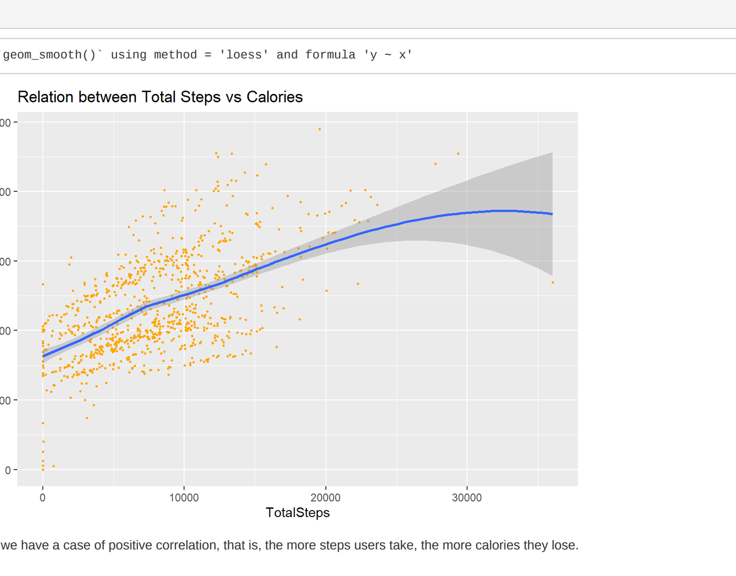



We can see that active minutes and calories burned are positively correlated, that is, the more active users are, the more calories they will burn. The geom_smooth function allows this correlation to be displayed on the graph.

The following graph will show the relationship between the total number of steps and the calories burned.

```
daily_activity %>%
  ggplot() +
  aes(x = TotalSteps, y = Calories) +
  geom_point(shape = "circle", size = 0.5, colour = "#ffa600") +
  geom_smooth(span = 0.75) +
  labs(
    title = "Relation between Total Steps vs Calories"
  )
```

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

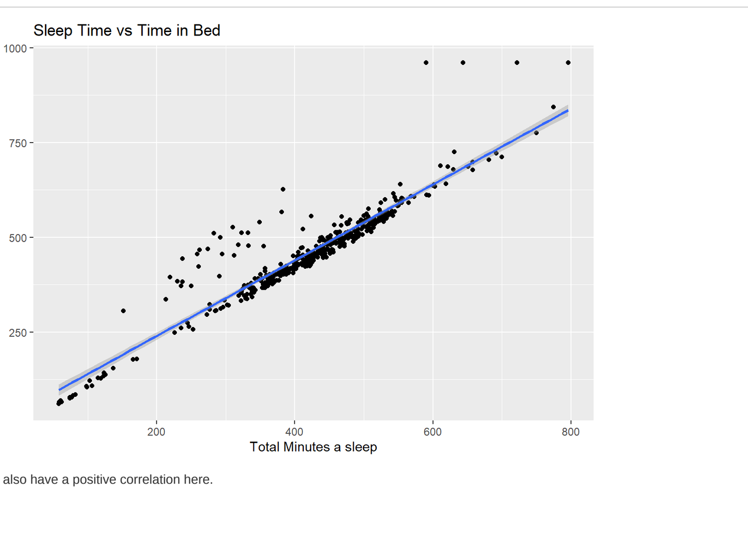



Again we have a case of positive correlation, that is, the more steps users take, the more calories they lose.

The following graph shows the relationship between time spent in bed and minutes of sleep.

```
ggplot(data=daily_sleep, aes(x=TotalMinutesAsleep, y=TotalTimeInBed)) +
  geom_point() +
  stat_smooth(method = lm) +
  labs(x="Total Minutes a sleep", y="Total Time in Bed", title = "Sleep Time vs Time in Bed")
```

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




We also have a positive correlation here.

## 6. Act

At this stage, my goal is to summarize the main conclusions that I reached in this analysis and to recommend the stakeholders take certain actions.

First of all, I must point out that the data analyzed are limited due to many factors. They are not current, they were collected on a very small sample, which is why it is difficult to claim with certainty that they reflect the real situation in the population we are interested in.

The analysis showed that Fitbit users most often use the application on the weekend, that is, on Sundays, and then on Mondays and Tuesdays. Sunday is expected to be the strongest day, and Monday and Tuesday could be interpreted as the beginning of the week when users start with more motivation in monitoring their activity, but as the week goes by, most of them falter.

My proposal for Bellabeat would be to make a certain division of activity monitoring into weekends and weekdays, i.e. that weekends and weekdays have a certain norm that the user must meet (a certain percentage of calories burned or steps taken), and not that most of the activities are carried out for the weekend, or only on some weekdays. Furthermore, the analysis showed that there is no major difference in the representation of users by activity, that is, there is an equal number of users who, according to the steps achieved on a daily basis, can be classified as slightly active, somewhat active, active and very active.

The goal of the application should be to increase the percentage of active people, but that limit of 10,000 steps on a daily basis seems too heavy for most users. Even the limit of 7,500 steps, which I declared to be the beginning of a solid activity, is quite a challenge. So I would suggest to Bellabeat that the activity is not determined only on the basis of the steps achieved, but that the focus is on the trend. For example, if a user of their app takes an average of 5,000 steps per day one week, and 10 percent more the next, that would be considered a very active class, regardless of whether the number of steps would not be 10,000 per day.

I would definitely suggest Bellabeat to conduct a new user survey to increase the number of users who would be willing to provide their data on the use of a certain device.