Bellabeat_case

Aurelija Petrauskaitė

2022-09-21

Introduction

Welcome to my Bellabeat case study. In this real world case I will follow the Case Study Roadmap, which details the steps of the data analysis process: ask, prepare, process, analyze, share, and act.

About the company

Bellabeat was founded by Urška Sršen and Sando Mur in 2013. It is a high-tech company that manufactures health-focused smart products for women. By 2016, Bellabeat had grown - they opened offices around the world and launched multiple products(Leaf, smart watch, smart Spring water bottle). Bellabeat products became available through a growing number of online retailers in addition to their own e-commerce channel on their website. 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.

ASK

Business task

Find insights what can help guide marketing strategy for the company.

Stakeholders:

Urška Sršen and Sando Mur, Bellabeat marketing analytics team.

PREPARE

Dataset what I used is located on Kaggle. This Kaggle data set contains personal fitness tracker from thirty fitbit users. It was collected from 2016.04.12 to 2016.05.12. The users consented to the submission of personal tracker data such as: minute-level output for physical activity(daily, monthly and steps), heart rate, sleep monitoring.. I think data is valid because it is placed in well known page such as Kaggle.

FitBit Fitness Tracker Data dataset is organized into 18 .csv files in long format (the first column is repeated). Also this data set have some limitations, because it is a small dataset(only 33 users and only one month of tracking), with not much information about users(age, gender, city). I chose to analyze those 4 .csv files:

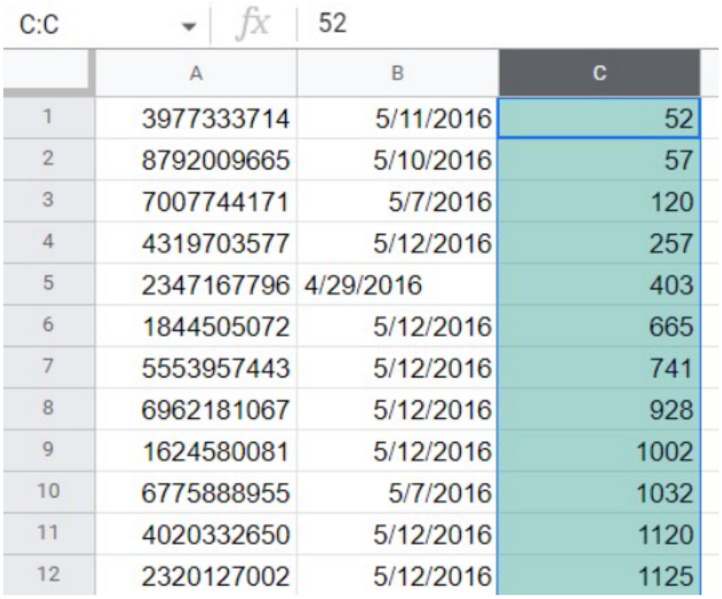
dailyCalories,

dailySteps,

sleepDay,

dailyActivity.

Before processing I filtered data to check for blank spaces and zero values. Also I sorted all my .csv files be ascending order(daily calories, daily steps, daily sleep, daily activity). So it is easier to see the smallest value.



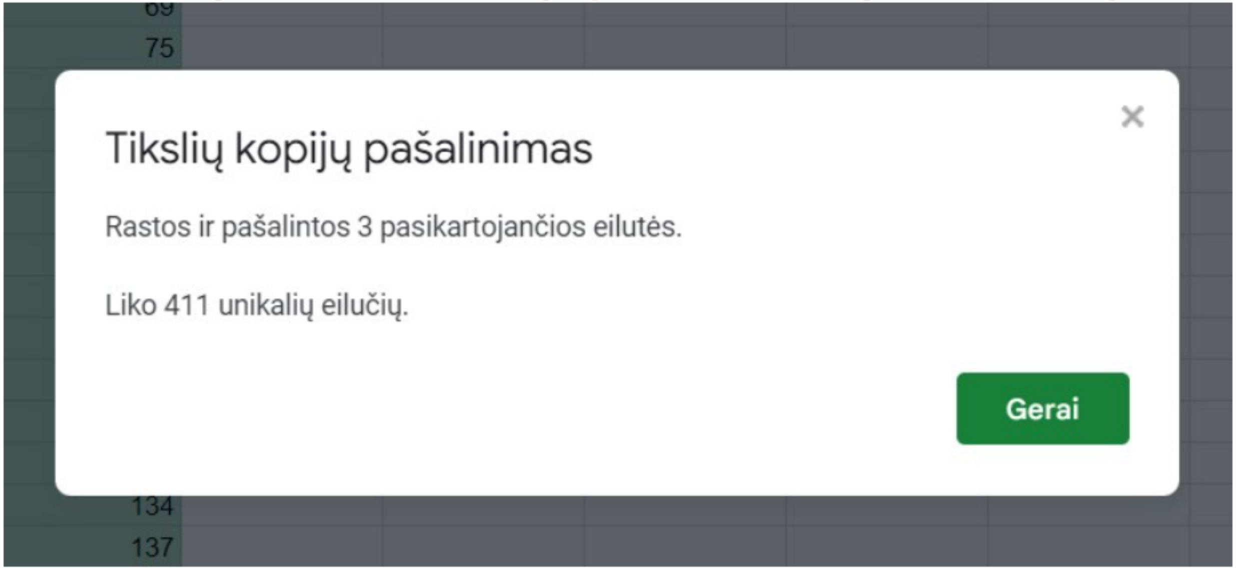
52

	A	B	C
1	3977333714	5/11/2016	52
2	8792009665	5/10/2016	57
3	7007744171	5/7/2016	120
4	4319703577	5/12/2016	257
5	2347167796	4/29/2016	403
6	1844505072	5/12/2016	665
7	5553957443	5/12/2016	741
8	6962181067	5/12/2016	928
9	1624580081	5/12/2016	1002
10	6775888955	5/7/2016	1032
11	4020332650	5/12/2016	1120
12	2320127002	5/12/2016	1125

Figure 1: *1.Daily calories in ascending order*

PROCESS

For this part I was using Google Sheet, RStudio, so I can see results faster. I've used naming conventions for clarity. First, I used "remove duplicates" function and at *sleepDay_clean.csv* I found 3 repeated rows. 411 unique rows left.



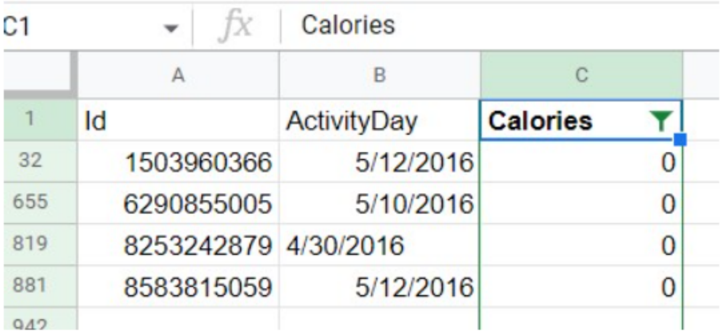
Tikslių kopijų pašalinimas

Rastos ir pašalintos 3 pasikartojančios eilutės.

Liko 411 unikalių eilučių.

Then, I checked the other files and they all had unique data. Also, I checked for empty cells in Google spreadsheet and found 4 zero values in *dailyCalories_clean.csv*.

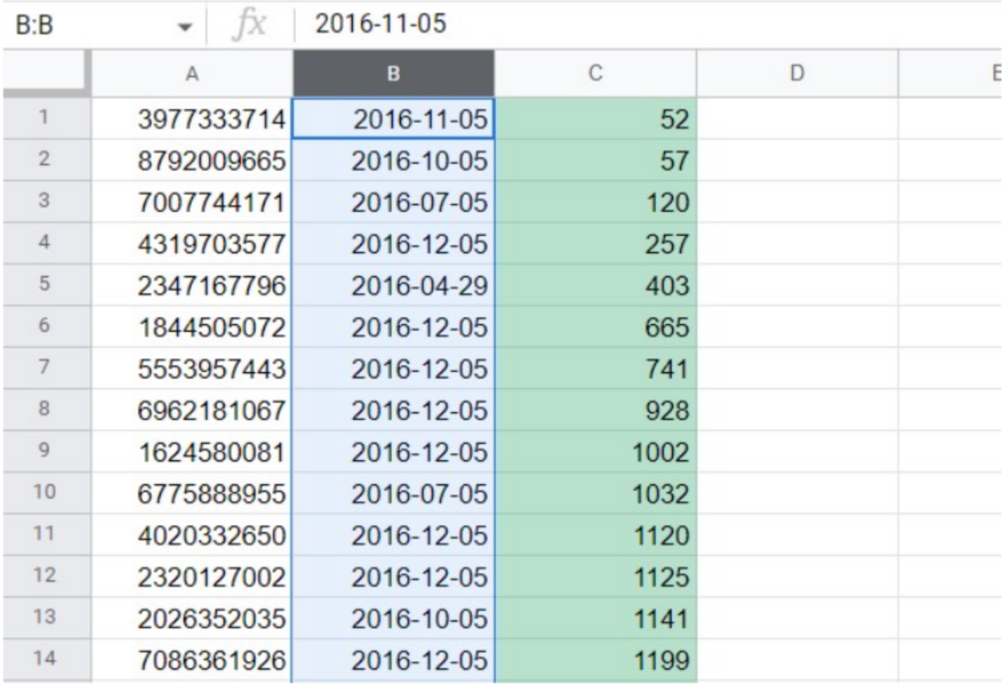
At last, I convert date type to yyyy-mm-dd format in daily calories file, what all the date would be in the



C1 | Calories

	A	B	C
1	Id	ActivityDay	Calories
32	1503960366	5/12/2016	0
655	6290855005	5/10/2016	0
819	8253242879	4/30/2016	0
881	8583815059	5/12/2016	0
942			

Figure 2: *3.Daily calories with zero value*



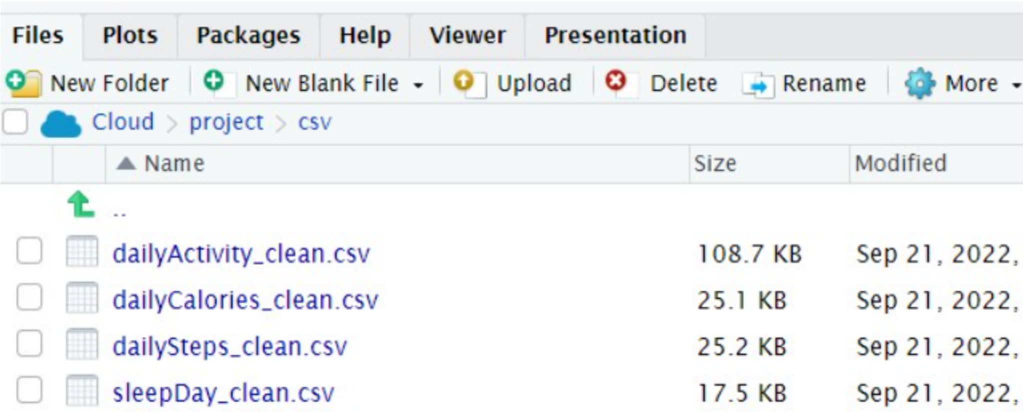
B:B | 2016-11-05

	A	B	C	D	E
1	3977333714	2016-11-05	52		
2	8792009665	2016-10-05	57		
3	7007744171	2016-07-05	120		
4	4319703577	2016-12-05	257		
5	2347167796	2016-04-29	403		
6	1844505072	2016-12-05	665		
7	5553957443	2016-12-05	741		
8	6962181067	2016-12-05	928		
9	1624580081	2016-12-05	1002		
10	6775888955	2016-07-05	1032		
11	4020332650	2016-12-05	1120		
12	2320127002	2016-12-05	1125		
13	2026352035	2016-10-05	1141		
14	7086361926	2016-12-05	1199		

same format.

ANALYZE

And now it's time to check if all the data is clean, and make some reviews and calculations and create visualizations.



Name	Size	Modified
..		
dailyActivity_clean.csv	108.7 KB	Sep 21, 2022,
dailyCalories_clean.csv	25.1 KB	Sep 21, 2022,
dailySteps_clean.csv	25.2 KB	Sep 21, 2022,
sleepDay_clean.csv	17.5 KB	Sep 21, 2022,

First I uploaded four files I will need.

Next step is to install the packages:

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
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   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Then, load my daily calories csv file and create dataframe *daily_calories*.

```
daily_calories <- read.csv("dailyCalories_clean.csv")
```

Next, I will create another two dataframes for steps and sleep data.

```
daily_steps <- read.csv("dailySteps_clean.csv")

daily_sleep <- read.csv("sleepDay_clean.csv")

daily_activity <- read.csv("dailyActivity_clean.csv")
```

Then take a quick look at the *daily_calories* data:

```
head(daily_calories)
```

```
##           Id ActivityDay Calories
## 1 1503960366  2016-12-04     1985
## 2 1503960366   4/13/2016     1797
## 3 1503960366   4/14/2016     1776
## 4 1503960366   4/15/2016     1745
## 5 1503960366   4/16/2016     1863
## 6 1503960366   4/17/2016     1728
```

To find out the existing files I used *colnames*:

```
colnames(daily_calories)
```

```
## [1] "Id"          "ActivityDay" "Calories"
```

Also I take a look at steps data:

```
head(daily_steps)
```

```
##           Id ActivityDay StepTotal
## 1 1503960366  2016-12-04     13162
## 2 1503960366   4/13/2016     10735
## 3 1503960366   4/14/2016     10460
## 4 1503960366   4/15/2016      9762
## 5 1503960366   4/16/2016     12669
## 6 1503960366   4/17/2016      9705
```

Daily sleep data :

```
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
```

Sleep data column names:

```
colnames(daily_sleep)
```

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

Next I counted how many unique users there are in each dataframe.

```
n_distinct(daily_calories$Id)
```

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

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

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

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

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

```
n_distinct(daily_steps$Id)
```

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

From this we can see that there are fewer sleep data users than other datasets(9 less).

I wanted to know how many rows are in the dataframes I used:

```
nrow(daily_activity)
```

```
## [1] 940
```

```
nrow(daily_sleep)
```

```
## [1] 410
```

```
nrow(daily_steps)
```

```
## [1] 940
```

```
nrow(daily_calories)
```

```
## [1] 940
```

From this data (number of rows) we can also see that there is much less sleep data - more than two times less. Therefore, it can be concluded that people either forget to charge the smart watch the night sleep before, or do not wear the watch when they sleep.

Some summary of *calories* dataframe:

```
daily_sleep%>%
  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
```

The data shows that people sleep about 419,2 minutes(almost 7 hours).So,this is a good sign, as this is the number of hours(7 to 9 hours) recommended by health professionals.

And quick look to all daily activity:

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

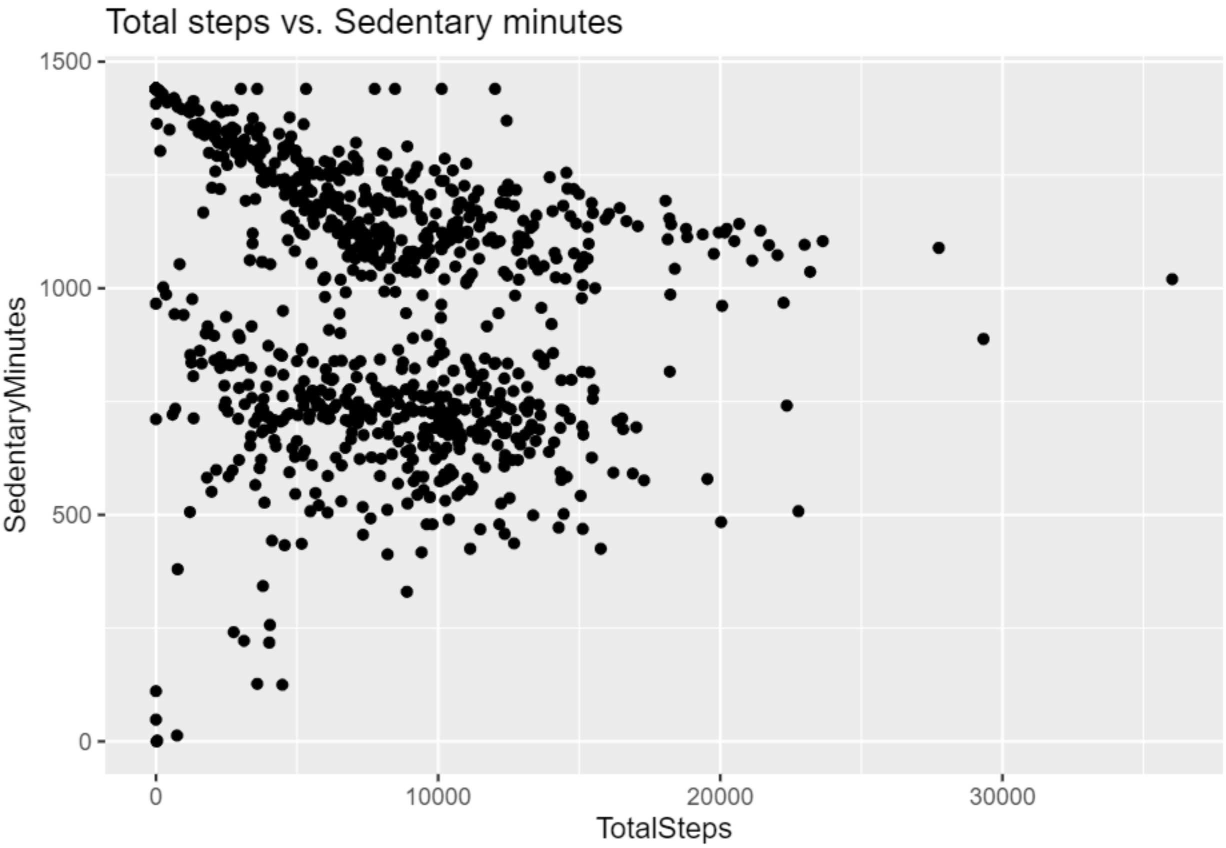
```
##    TotalSteps    TotalDistance    SedentaryMinutes
##  Min.   :    0   Min.   : 0.000   Min.   :   0.0
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 729.8
##  Median : 7406   Median : 5.245   Median :1057.5
##  Mean   : 7638   Mean   : 5.490   Mean   : 991.2
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:1229.5
##  Max.   :36019   Max.   :28.030   Max.   :1440.0
```

From this we can see that users are less active than recommended by the World Health Organization. The average user walks 7,638 steps, and the recommended step is 10,000 steps.

SHARE

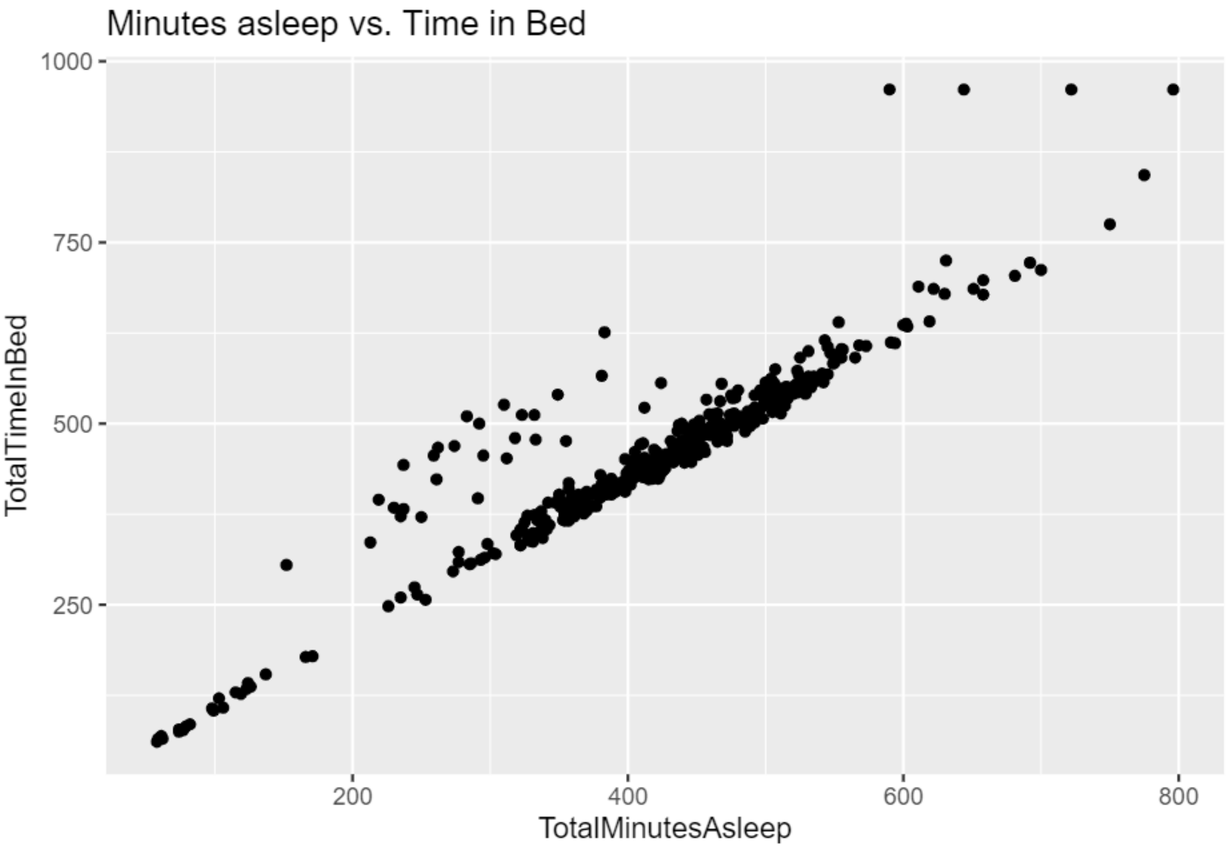
To see insights or patterns I used ggplot function. First, I examined how sedentary minutes are related to daily steps:

```
ggplot(data=daily_activity, aes(TotalSteps, y=SedentaryMinutes))+ geom_point()+
  labs(title="Total steps vs. Sedentary minutes")
```



So I can see what there is some correlation between that. Next, I wanted to know if the time spent in bed is related to the hours(minutes) of sleep.

```
ggplot(data=daily_sleep, aes(x=TotalMinutesAsleep, y=TotalTimeInBed))+ geom_point()+
  labs(title="Minutes asleep vs. Time in Bed")
```



From this we can see that most users spend time in bed and sleep duration is similar.

Next, I wanted to know if those who sleep the most are the most active?For that I needed to merge two datasets(daily steps and daily sleep).

```
sleep_steps <- merge(daily_sleep, daily_steps, by = "Id")
```

Quick look at how many users are currently in merged dataset.

```
n_distinct(sleep_steps$Id)
```

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

When we merged the two datasets we lost some users, because there is only 24 users in *daily_sleep* data.But *dailySteps* has 33 users. So the data can be not accurate.

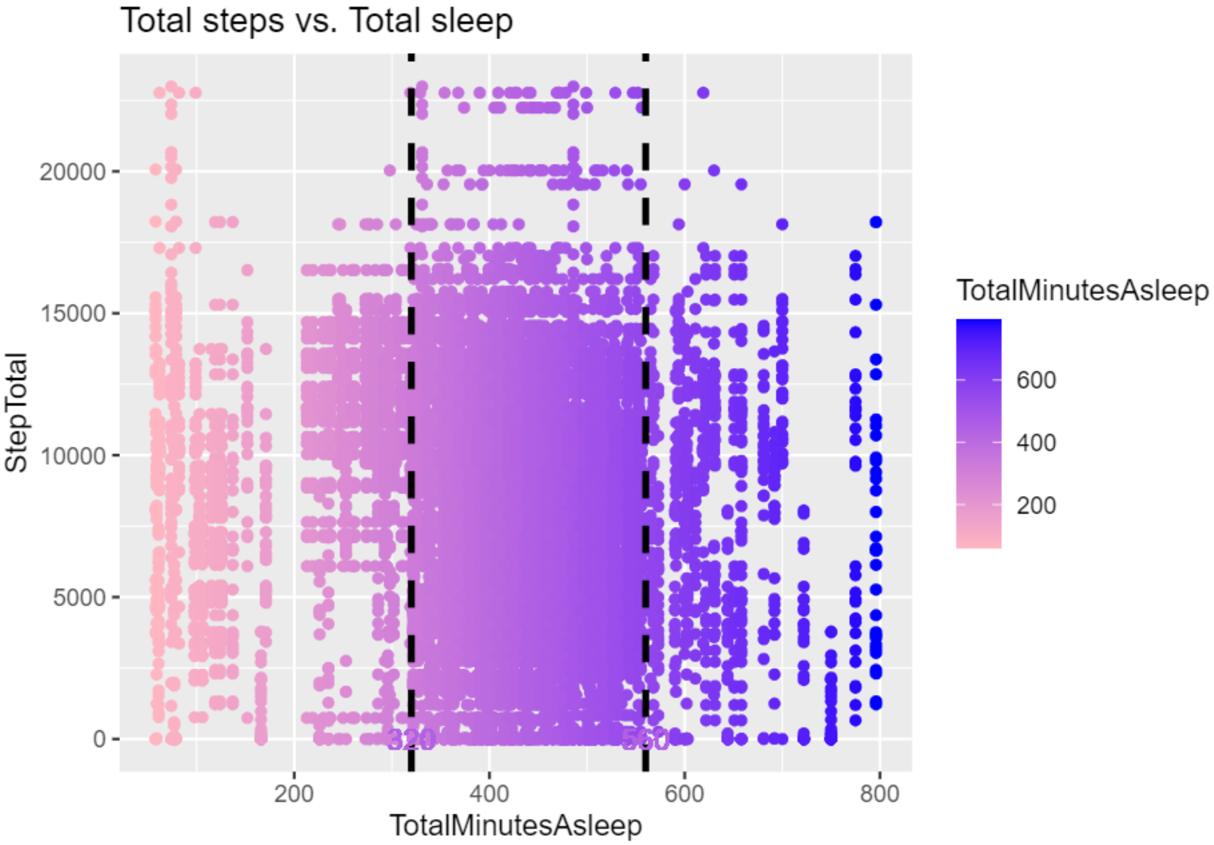
It would be good to know what the columns in the merged dataset are now:

```
colnames(sleep_steps)
```

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

With a merged data set, we can test whether there is a relationship between sleep and steps.

```
ggplot(data=sleep_steps, aes(x=TotalMinutesAsleep, y=StepTotal, color=TotalMinutesAsleep))+
  geom_point()+
  geom_vline(xintercept = 320, linetype="dashed", color = "black", size = 1.2)+
  geom_text(aes(x=320, y=0, label="320"))+
  geom_vline(xintercept = 560, linetype="dashed", color = "black", size = 1.2)+
  geom_text(aes(x=560, y=0, label="560"))+
labs(title="Total steps vs. Total sleep")+
scale_color_gradient(low="lightpink", high="blue")
```



From the available data, we can see that most users who sleep between 320 minutes(6 hours) and 560 minutes(9hours) are most active during the day. I separated the most active period with two dashed lines. Therefore, we can do conclusion that those who sleep more walk more steps per day.

ACT

Recommendation for Bellabeat: *1. Recommend users to wear the watch more often at night.*

2. Also to remind users about the decreasing battery when it will be left, for example, 40%, 30%, 20%, 10%.

3. Send newsletters about health, benefits, power of movement.

4. Remind users to be active.

5. Collect more data about user habits for further analysis.