Data Analytics Case Study with R

G NAFISA MASHITHA

Introduction

Welcome to my Bellabeat data analysis case study! In this case study, I will perform real-world tasks of a data analyst. And in order to answer some key business questions, I will follow the steps of the data analysis process: ask, prepare, process, analyze, share, and act.

About the company Bellabeat

Bellabeat, is a high-tech manufacturer of health-focused products for women. Bellabeat is a successful small company, and they have the potential to become a larger player in the global smart device market. Urška Sršen and Sando Mur founded Bellabeat, a high-tech company that manufactures health-focused smart products. 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.

Scenario of the study

In this study, I will 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 will then help guide marketing strategy for the company

Questions for the Analysis (Ask phase)

In this phase, I tried to better undestand the data and the problem I'm trying to solve. And to do that, I had to do more research and ask more questions.

- What are some trends in smart device usage? How could these trends help influence Bellabeat marketing strategy ? So first, the company need to better target their marketing efforts into their customer's needs based on their usage of their fitness smart devices. And then, make high-level recommendations for how these trends can inform Bellabeat marketing strategy.
- Who are the main stakeholders ? The main stakeholders are Urška Sršen, Bellabeat's co-founder and Chief Creative Officer; Sando Mur, Mathematician and Bellabeat's cofounder; And also, we need to think about and work with the rest of the Bellabeat marketing analytics team.

Business Task

Now, after getting answers to all of my questions (during the ask phase), I'm able to define clearly the business task which is: To analyse how do Bellabeat customers use their smart devices? And then, Identify potential opportunities for growth and recommendations for the Bellabeat marketing team based on trends in smart device usage.

Preparing the Data (Prepare Phase)

In this phase, I will download and Import the dataset. Then make sure all the data is organized and credible. And I will sort and filter the data.

Downloading the data

Bellabeat encourages me to use public data that explores smart device users' daily habits from FitBit Fitness Tracker Data. FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius). 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. And It includes information about daily activity, steps, and heart rate that can be used to explore users' habits. So, here is the link to download the dataset:

- FitBit Fitness Tracker Data : https://www.kaggle.com/arashnic/fitbit.

About the dataset: The dataset was generated by respondents to a distributed survey via Amazon Mechanical Turk between 03.12.2016 and 05.12.2016. And include 18 CSV files.

Loading packages

Now, I'm going to Install some R packages that will help me in my analysis. And I'm using in my code the options `message=FALSE` and `warning=FALSE`, to save space. And to prevent printing of the execution of the R code generated and the warning messages. And I will add some data cleaning packages as well (last 3 packages)

```
install.packages("tidyverse")
install.packages("lubridate")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("tidyr")
install.packages("here")
install.packages("skimr")
install.packages("janitor")
```

Now, I'm going to load these packages. And I'm using in my code the options `message=FALSE` and `warning=FALSE`, to save space. And to prevent printing of the execution of the R code generated and the warning messages.

```
library(tidyverse)
library(lubridate)
library(dplyr)
library(ggplot2)
library(tidyr)
library(here)
library(skimr)
library(janitor)
```

Importing dataset

Now, I'm going to Import all dataset. Then VIEW, CLEAN, FORMAT, and ORGANIZE the data. After reviewing all the dataset, I decided to make some asumptions and work only with these data for my analysis:

- dailyActivity_merged.csv

```
Activity <- read.csv("dailyActivity_merged.csv")
head(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
```

```
colnames(Activity)
```

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

```
str(Activity)
```

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

- dailyCalories_merged.csv

```
Calories <- read.csv("dailyCalories_merged.csv")
head(Calories)
```

```
##           Id ActivityDay Calories
## 1 1503960366   4/12/2016     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
```

```
colnames(Calories)
```

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

```
str(Calories)
```

```
## 'data.frame':    940 obs. of  3 variables:
##  $ Id         : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDay: chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ Calories   : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

- dailyIntensities_merged.csv

```
Intensities <- read.csv("dailyIntensities_merged.csv")
head(Intensities)
```

```
##           Id ActivityDay SedentaryMinutes LightlyActiveMinutes
## 1 1503960366   4/12/2016              728                  328
## 2 1503960366   4/13/2016              776                  217
## 3 1503960366   4/14/2016             1218                  181
## 4 1503960366   4/15/2016              726                  209
## 5 1503960366   4/16/2016              773                  221
## 6 1503960366   4/17/2016              539                  164
##   FairlyActiveMinutes VeryActiveMinutes SedentaryActiveDistance
## 1                  13                25                       0
## 2                  19                21                       0
## 3                  11                30                       0
## 4                  34                29                       0
## 5                  10                36                       0
## 6                  20                38                       0
##   LightActiveDistance ModeratelyActiveDistance VeryActiveDistance
## 1                6.06                     0.55               1.88
## 2                4.71                     0.69               1.57
## 3                3.91                     0.40               2.44
## 4                2.83                     1.26               2.14
## 5                5.04                     0.41               2.71
## 6                2.51                     0.78               3.19
```

```
colnames(Intensities)
```

```
##  [1] "Id"                       "ActivityDay"
##  [3] "SedentaryMinutes"         "LightlyActiveMinutes"
##  [5] "FairlyActiveMinutes"      "VeryActiveMinutes"
##  [7] "SedentaryActiveDistance"  "LightActiveDistance"
##  [9] "ModeratelyActiveDistance" "VeryActiveDistance"
```

```
str(Intensities)
```

```
## 'data.frame':    940 obs. of  10 variables:
##  $ Id                       : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDay              : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ SedentaryMinutes         : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ LightlyActiveMinutes     : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ FairlyActiveMinutes      : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ VeryActiveMinutes        : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LightActiveDistance      : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ VeryActiveDistance       : num  1.88 1.57 2.44 2.14 2.71 ...
```

- heartrate_seconds_merged.csv

```
Heartrate <- read.csv("heartrate_seconds_merged.csv")
head(Heartrate)
```

```
##           Id                  Time Value
## 1 2022484408 4/12/2016 7:21:00 AM    97
## 2 2022484408 4/12/2016 7:21:05 AM   102
## 3 2022484408 4/12/2016 7:21:10 AM   105
## 4 2022484408 4/12/2016 7:21:20 AM   103
## 5 2022484408 4/12/2016 7:21:25 AM   101
## 6 2022484408 4/12/2016 7:22:05 AM    95
```

```
colnames(Heartrate)
```

```
## [1] "Id"    "Time"  "Value"
```

```
str(Heartrate)
```

```
## 'data.frame':    2483658 obs. of  3 variables:
##  $ Id   : num  2.02e+09 2.02e+09 2.02e+09 2.02e+09 2.02e+09 ...
##  $ Time : chr  "4/12/2016 7:21:00 AM" "4/12/2016 7:21:05 AM" "4/12/2016 7:21:10 AM" "4/12/2016 7:21:1
##  $ Value: int  97 102 105 103 101 95 91 93 94 93 ...
```

- sleepDay_merged.csv

```
Sleep <- read.csv("sleepDay_merged.csv")
head(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
```

```
colnames(Sleep)
```

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

```
str(Sleep)
```

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

- weightLogInfo_merged.csv

```
Weight <- read.csv("weightLogInfo_merged.csv")
head(Weight)
```

```
##           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
```

```
colnames(Weight)
```

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

```
str(Weight)
```

```
## 'data.frame':    67 obs. of  8 variables:
##  $ Id            : num  1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...
##  $ Date          : chr  "5/2/2016 11:59:59 PM" "5/3/2016 11:59:59 PM" "4/13/2016 1:08:52 AM" "4/21/2
##  $ WeightKg      : num  52.6 52.6 133.5 56.7 57.3 ...
##  $ WeightPounds  : num  116 116 294 125 126 ...
##  $ Fat           : int  22 NA NA NA NA 25 NA NA NA NA ...
##  $ BMI           : num  22.6 22.6 47.5 21.5 21.7 ...
##  $ IsManualReport: chr  "True" "True" "False" "True" ...
##  $ LogId         : num  1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
```

So now, we can see that everything were imported correctly.

Cleaning the dataset (Process Phase)

Basics cleaning:

Now, I'm going to Process, Clean and Organize the dataset for analysis. I used functions like `glimpse()`, `skim_without_charts` to quickly review the data. I also clean the names of the data using `clean_names()`.

And here some cleaning steps I did with the data :

- For Dataset (Activity, Calories and Intensities): For the data cleaning steps, I did NOT FOUND in this data (Spelling errors, Misfield values, Missing values, Extra and blank space, no duplicated found). For formatting, I used clear formatting. For Data types, some data were converted to numeric and Dates columns will be converted to date type.
- For Sleep data : 3 duplicates were found and removed.
- For Weight data : too many missing values were found in one column. And I decided to remove that column.

Fixing formatting

I spotted some problems with the timestamp data. So before analysis, I need to convert it to date time format and split to date and time.

```
# Activity
Activity$ActivityDate=as.POSIXct(Activity$ActivityDate, format="%m/%d/%Y", tz=Sys.timezone())
Activity$date <- format(Activity$ActivityDate, format = "%m/%d/%y")
Activity$ActivityDate=as.Date(Activity$ActivityDate, format="%m/%d/%Y", tz=Sys.timezone())
Activity$date=as.Date(Activity$date, format="%m/%d/%Y")


# Intensities
Intensities$ActivityDay=as.Date(Intensities$ActivityDay, format="%m/%d/%Y", tz=Sys.timezone())


# Sleep
Sleep$SleepDay=as.POSIXct(Sleep$SleepDay, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
Sleep$date <- format(Sleep$SleepDay, format = "%m/%d/%y")
Sleep$date=as.Date(Sleep$date, "% m/% d/% y")
```

Now that everything is ready, I can start exploring and analyzing the data sets.

Summarizing the dataset (Analyze Phase)

Now that all the data is stored appropriately and has been prepared for analysis, I can start putting it to work.

Let's look at the total number of participants in each data sets:

```
Activity %>%
  summarise(Activity_participants = n_distinct(Activity$Id))
```

```
##   Activity_participants
## 1                    33
```

```
n_distinct(Calories$Id)
```

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

```
n_distinct(Intensities$Id)
```

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

```
n_distinct(Heartrate$Id)
```

```
## [1] 14
```

```
n_distinct(Sleep$Id)
```

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

```
n_distinct(Weight$Id)
```

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

So, there are 33 participants in the activity, calories and intensities data sets. 24 participants in the Sleep data. And only 14 participants for Heartrate, and only 8 in the weight data set. 8 and 14 participants are not significant to make any recommendations and conclusions based on these dataset.

So I will focus on these datasets for my analysis: Activity, Calories, Intensities and Sleep.

Here are some quick summary statistics about each data frame.

For the Activity dataframe:

```
# Activity
Activity %>%
  select(TotalSteps,
         TotalDistance,
         SedentaryMinutes, Calories) %>%
  summary()
```

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

Exploring the number of Intense active participants :

```
# Explore number of active minutes per category
Intensities %>%
  select(VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, SedentaryMinutes) %>%
  summary()
```

```
##  VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :  0.0        Min.   :   0.0
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:127.0        1st Qu.: 729.8
##  Median :  4.00    Median :  6.00      Median :199.0        Median :1057.5
##  Mean   : 21.16    Mean   : 13.56      Mean   :192.8        Mean   : 991.2
##  3rd Qu.: 32.00    3rd Qu.: 19.00      3rd Qu.:264.0        3rd Qu.:1229.5
##  Max.   :210.00    Max.   :143.00      Max.   :518.0        Max.   :1440.0
```

For the Calories dataframe:

```
# Calories
Calories %>%
  select(Calories) %>%
  summary()
```

```
##     Calories
```

```
##  Min.   :   0
##  1st Qu.:1828
##  Median :2134
##  Mean   :2304
##  3rd Qu.:2793
##  Max.   :4900
```

For the Sleep dataframe:

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

```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.000     Min.   : 58.0      Min.   : 61.0
##  1st Qu.:1.000     1st Qu.:361.0      1st Qu.:403.0
##  Median :1.000     Median :433.0      Median :463.0
##  Mean   :1.119     Mean   :419.5      Mean   :458.6
##  3rd Qu.:1.000     3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :3.000     Max.   :796.0      Max.   :961.0
```

For the Weight dataframe:

```
# Weight
Weight %>%
  select(WeightKg, Fat) %>%
  summary()
```

```
##     WeightKg           Fat
##  Min.   : 52.60   Min.   :22.00
##  1st Qu.: 61.40   1st Qu.:22.75
##  Median : 62.50   Median :23.50
##  Mean   : 72.04   Mean   :23.50
##  3rd Qu.: 85.05   3rd Qu.:24.25
##  Max.   :133.50   Max.   :25.00
##                   NA's   :65
```

Key findings from this analysis :

- The average sedentary time is too high (more than 16 hours). And definitely needs to be reduced with a good marketing strategy.
- The majority of the participants are lightly active. With a high sedentary time.
- Participants sleep 1 time for an average of 7 hours.
- Average total steps per day (which is 7638) is a little bit less than recommended by the CDC. According to the CDC research, taking 8,000 steps per day was associated with a 51% lower risk for all-cause mortality (or death from all causes). And taking 12,000 steps per day was associated with a 65% lower risk compared with taking 4,000 steps.

Merging some data :

Before beginning to visualize the data, I'm going to merge two data sets : Activity and Sleep data on columns Id. Note that there are more participant Ids in the Activity dataset than in the Sleep dataset. So if I use the merge option `inner_joint`, then I will have the number of participants from the Sleep data set.

Take a look :

```
Combined_data_inner <- merge(Sleep, Activity, by="Id")
n_distinct(Combined_data_inner$Id)
```

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

So for analysis, I will consider using 'outer_join' to keep all participants in the in the dataset. And I can do that by adding in my code chunk the extra argument `all=TRUE`.

```
Combined_data_outer <- merge(Sleep, Activity, by="Id", all = TRUE)
n_distinct(Combined_data_outer$Id)
```

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

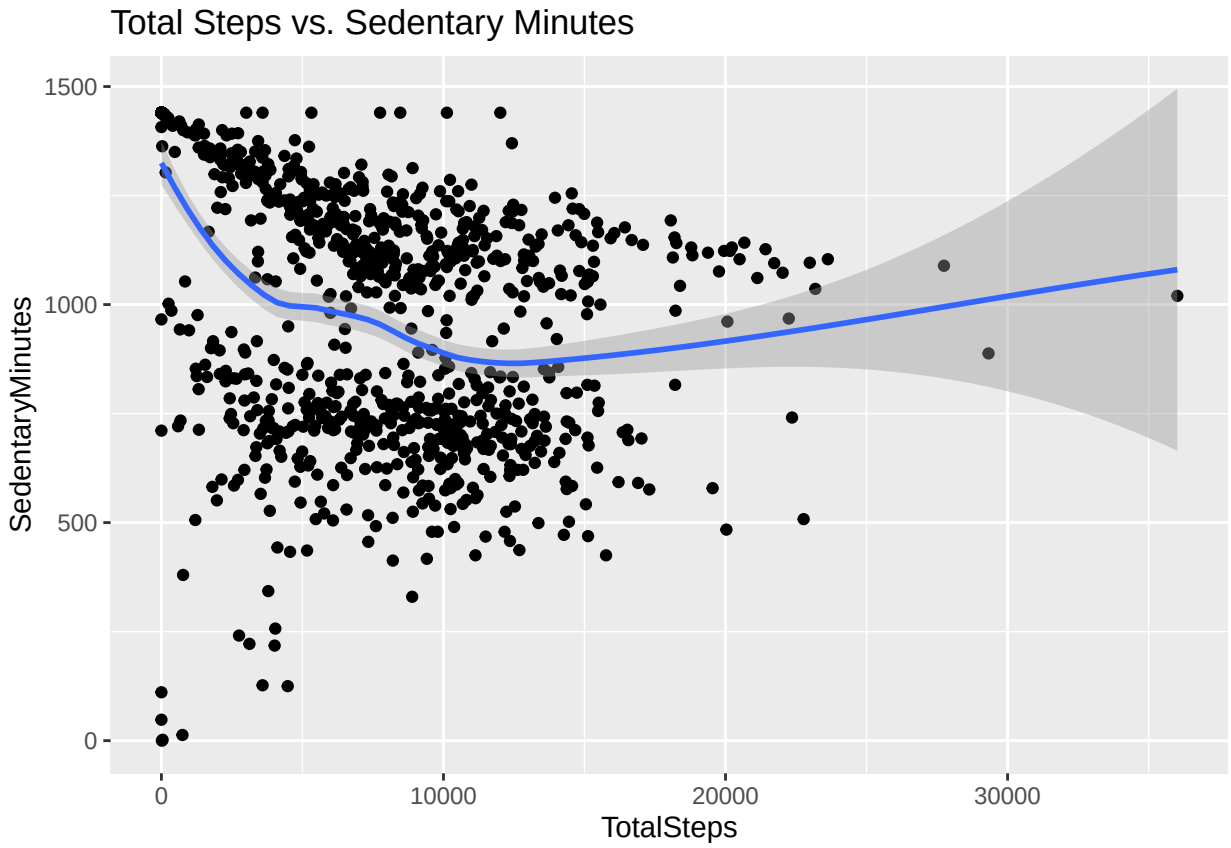
Data visualization (Share and Act Phases)

Now let's visualize some key explorations.

Relationship between Steps and Sedentary time

What's the relationship between steps taken in a day and sedentary minutes?

```
ggplot(data=Activity, aes(x=TotalSteps, y=SedentaryMinutes)) + geom_point() + geom_smooth() + labs(titl
```

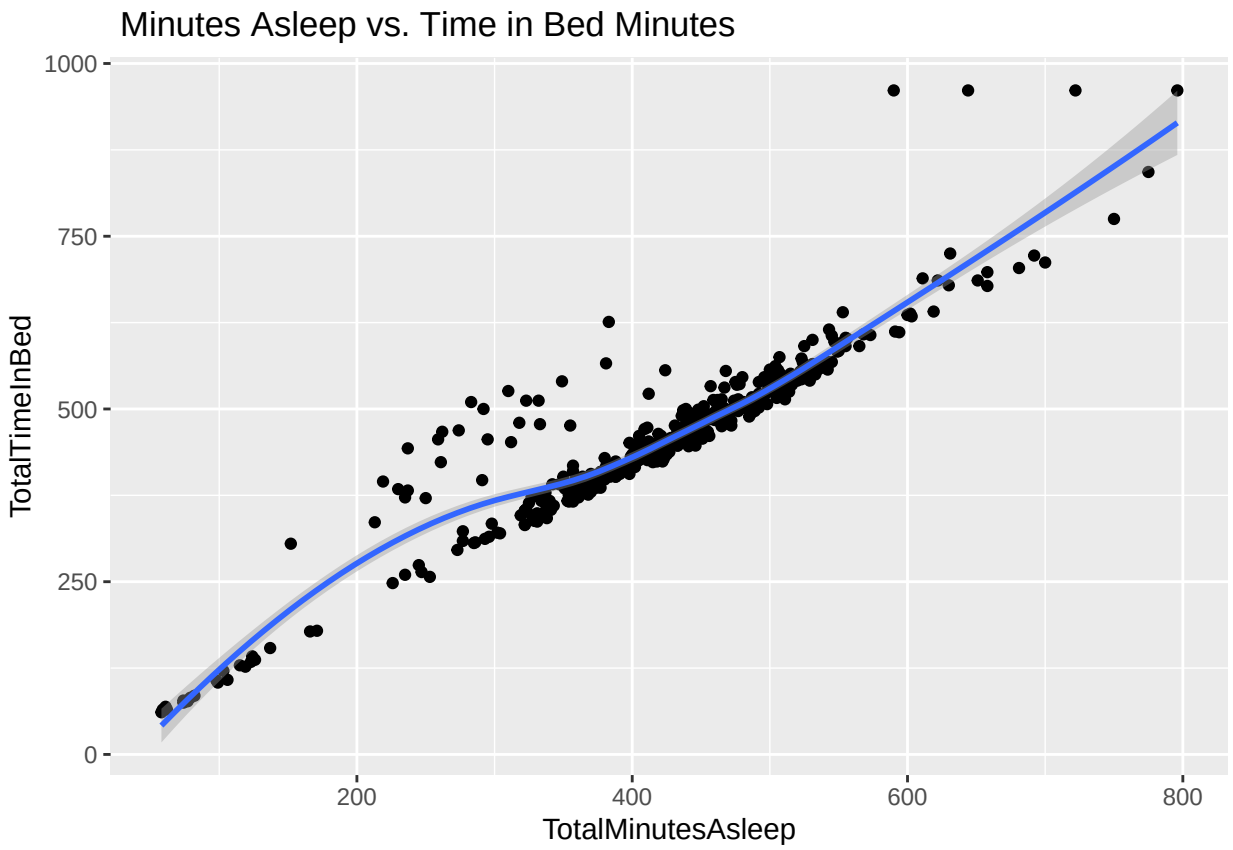


I can see here a negative correlation between Steps and Sedentary time. The more Sedentary time you have, the less Steps you're taking during the day. This data shows that the company need to market more the customer segments with high Sedentary time. And to do that, the company needs to find ways to get customers get started in walking more and also measure their daily steps.

Relationship between Minutes Asleep and Time in Bed

What's the relationship between minutes asleep and time in bed?

```
ggplot(data=Sleep, aes(x=TotalMinutesAsleep, y=TotalTimeInBed)) + geom_point()+ geom_smooth() + labs(ti
```

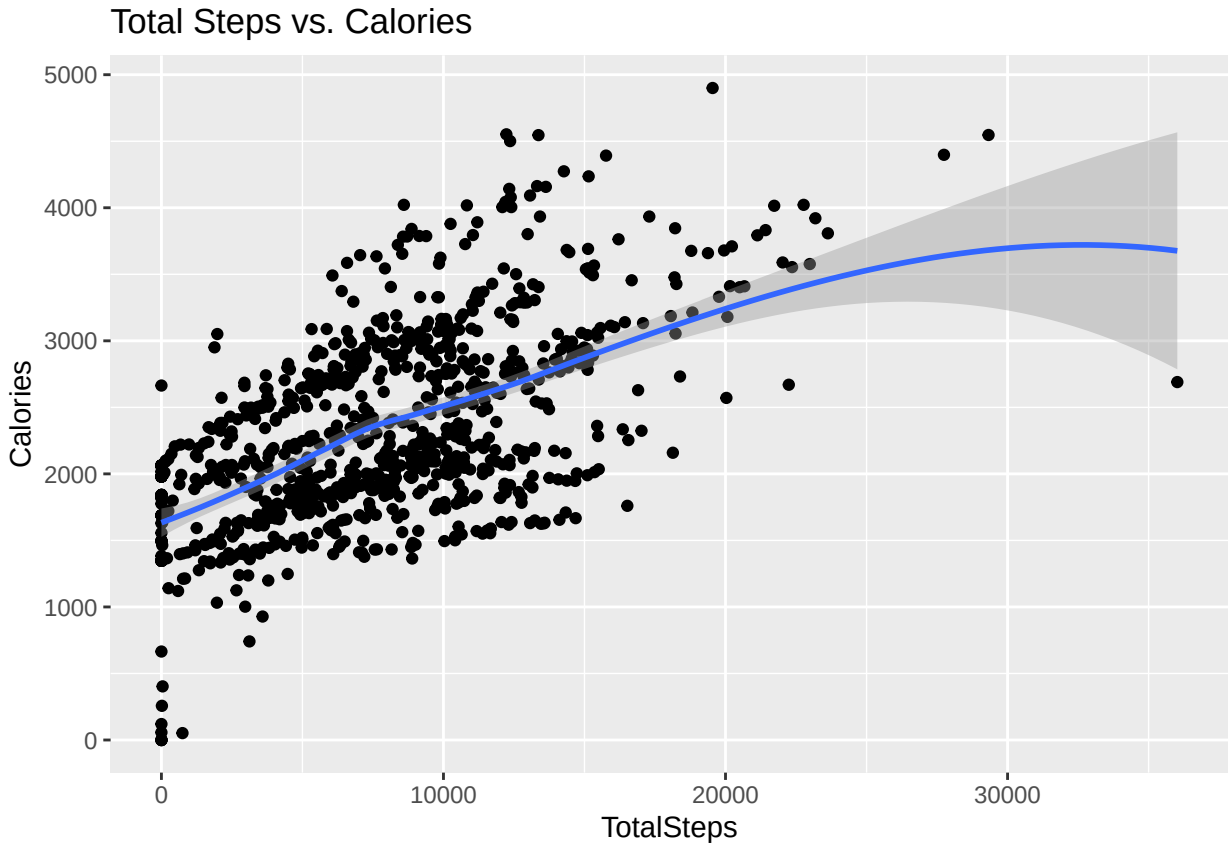


As we might expect, we can see here an almost completely linear trend between Minutes Asleep and Time in Bed. So to help users improve their sleep, the company should consider using notification to go to sleep.

Relationship between Steps and Calories

What's the relationship between steps taken and Calories ?

```
ggplot(data=Activity, aes(x=TotalSteps, y=Calories)) +
  geom_point() + geom_smooth() + labs(title="Total Steps vs. Calories")
```



We can see here a positive correlation between Total Steps and Calories. The more active we are, the more calories we will burn.

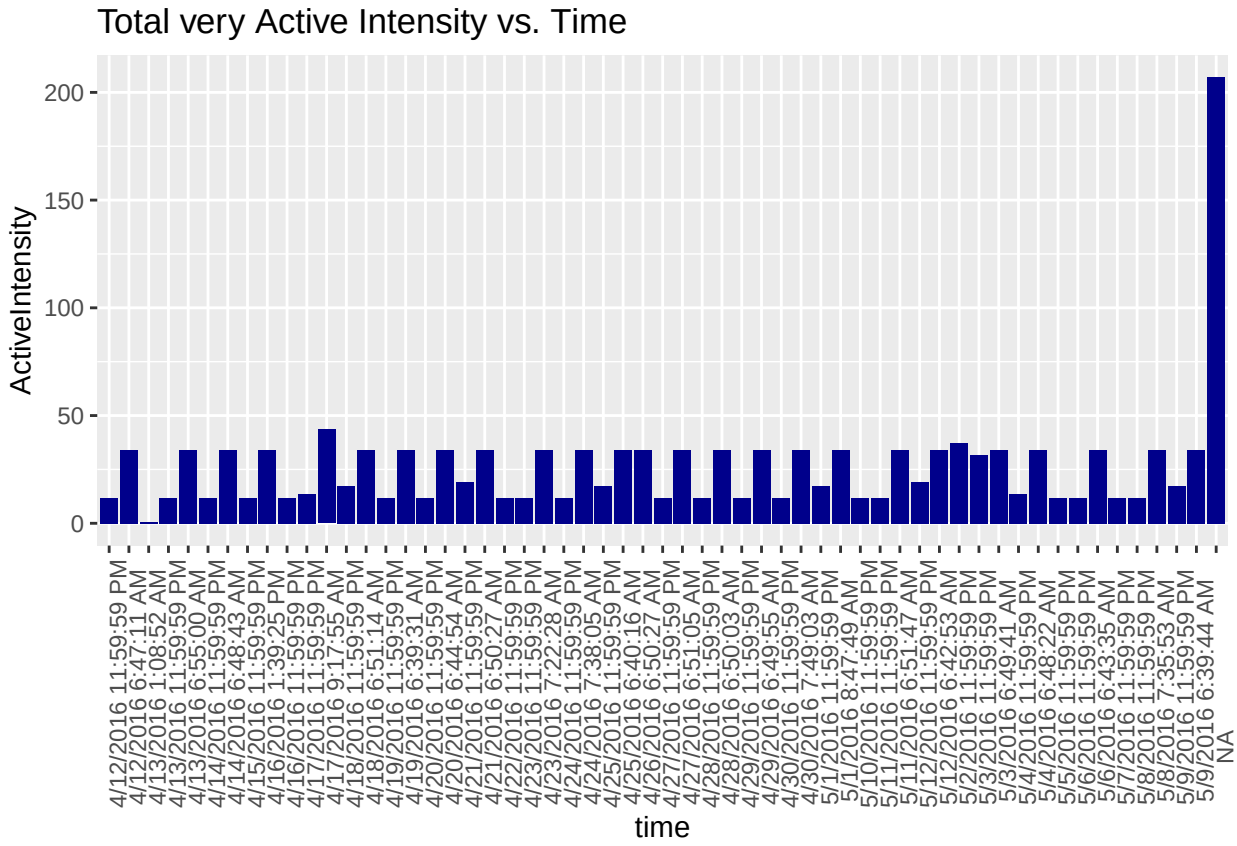
Intensities data

Now, let's look at some Intensities data over time.

```
Intensities$ActiveIntensity <- (Intensities$VeryActiveMinutes)/60

Combined_data <- merge(Weight, Intensities, by="Id", all=TRUE)
Combined_data$time <- format(Combined_data$Date, format = "%H:%M:%S")

ggplot(data=Combined_data, aes(x=time, y=ActiveIntensity)) + geom_histogram(stat = "identity", fill='da
  theme(axis.text.x = element_text(angle = 90)) +
  labs(title="Total very Active Intensity vs. Time ")
```



By analysing some Intensity data over time. The company will have a good idea on how customers are using their product during the day. Most users are actif before and after work, I suppose. The company can use this time in the Bellabeat app to remind and motivate users to go for a run or for a walk.

Conclusions & Recommandations for the Business

So, collecting data on activity, sleep, stress, etc. will allow the company Bellabeat to empower the customers with knowledge about their own health and daily habits. The company Bellabeat is growing rapidly and quickly positioned itself as a tech-driven wellness company for their customers.

By analyzing the FitBit Fitness Tracker Data set, I found some insights that would help influence Bellabeat marketing strategy.

Target Audience:

People working full-time jobs and spending a lot of time at the computer and in the office and need fitness and daily activities to be in shape.

The users are doing some light activity to stay healthy (according to the activity type analysis). And they need to improve their everyday activity to have more health benefits. And they might need some knowledge about developing healthy habits and motivation to keep them going.

Message to the Company

The Bellabeat app need to be a unique fitness activity app. By becoming a companion guide (like a friend) to its users and customers and help them balance their personal and professional life with healthy habits.

Recommendations to the Bellabeat Marketing team

- The average sedentary time is too high for the users of the app (more than 16 hours). And definitely needs to be reduced with a good marketing strategy. So, the data shows that the company need to market more to the customer segment with a high Sedentary time. And to do that, the company needs to find ways to get customers started in walking more by measuring their daily steps (+ notifications).
- Participants sleep 1 time for an average of 7 hours. To help users improve their sleep, Bellabeat should consider using app notifications to go to bed. And also, the Bellabeat app can recommend reducing sedentary time for its customers.
- The average total steps per day (which is 7638) is a little bit less than recommended by the CDC. According to the CDC research, taking 8,000 steps per day was associated with a 51% lower risk for all-cause mortality (or death from all causes). And taking 12,000 steps per day was associated with a 65% lower risk compared with taking 4,000 steps. So, Bellabeat can encourage people to take at least 8,000 steps per day by explaining the healthy benefits of doing that.
- By analysing the Intensity data over time. The company will have a good idea on how their customers are using their app during the day. Most users are actif before and after work. The company can use this time in the Bellabeat app to remind and motivate users to go for a run or for a walk.
- For customers who want to lose weight, it can be a good idea to control daily calorie consumption. And Bellabeat can suggest some ideas for low-calorie healthy food (for lunch and dinner).

Thank you very much for your interest in my Bellabeat Case Study!

And I would appreciate any comments and recommendations for improvement!

Please stay well and bye bye !