Bellabeat case study with R

Mopelola

2023-02-09

Introduction

Bellabeat, a high-tech manufacturer of health-focused products for women. They produce products such as Bellabeat app, Leaf, Time and Spring that provides users with health data related to their activity, sleep, stress, menstrual cycle, and mindfulness habits.

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.

Business Question.

As a junior data Analyst, I have been asked to analyze smart device usage data in order to gain insight into how consumers use non-Bellabeat smart devices. I used one of the Bellabeat products to apply these insights into my presentation.

These questions guided my Analysis: 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?

Preparing the Data

The FitBit Fitness Tracker Data was used. A public data that explores smart device users' daily habits. The data set was downloaded , stored appropriately and imported on R studio. I Identified how the data was organized, then I proceeded to sort and filter the data.

Processing the data

Loading packages

Some R packages were installed and loaded. These packages will be used in my analysis.

installing the packages.

```
install.packages("tidyverse")

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

```r
install.packages("lubridate")

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

install.packages("dplyr")

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

install.packages("ggplot2")

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

install.packages("tidyr")

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

install.packages("here")

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

install.packages("skimr")

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

install.packages("janitor")

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

loading the packages.

```r
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✔ dplyr     1.1.1     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force
all conflicts to become errors
```

```
library(lubridate)
library(dplyr)
library(ggplot2)
library(tidyr)
library(here)

## here() starts at /cloud/project

library(skimr)
library(janitor)

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

Importing datasets

The FitBit Fitness Tracker Data was used for this analysis. The fitness Tracker Data was imported.

The FitBit Fitness Tracker Data was used for this analysis. The fitness Tracker Data was imported. Before importing this dataset, the dataset was downloaded and was viewed in excel.

The Fitness Tracker Data has 18 merged csv files(Data frames) that can be used to carry our this analysis. For this analysis I used the: * DailyActivity * DailyCalories * SleepDay * Dailysteps.

importing and renaming the dataframes

```
Activity <- read.csv("dailyActivity_merged.csv")
Calories <- read.csv("dailyCalories_merged.csv")
Sleep <- read.csv("sleepDay_merged.csv")
Steps <- read.csv("dailySteps_merged.csv")
```

Previewing the data: All the 4 dataframes were viewed to ensure that the files were properly imported. head(), was used to view the first 6rolls of the data.

```
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
```

```
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
```

```
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
```

```
head(Steps)
```

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

```
view(Activity)
view(Calories)
view(Sleep)
view(Steps)
```

Cleaning the dataset

The dataset was cleaned and organized to be ready for analysis. While cleaning the data, I checked for some unique ids, checking for duplicates, removing duplicates, made sure the headers are clean and also made sure that the format types of the data are consistent.

Counting unique IDs:

```
n_unique(Activity$Id)

## [1] 33

n_unique(Calories$Id)

## [1] 33

n_unique(Sleep$Id)

## [1] 24

n_unique(Steps$Id)

## [1] 33
```

There are 33, 33, 24 and 33 unique IDs in the activity, Calories, Sleep and Step datasets respectively. Though the sleep dataframe does not meet the minimum sample size of 30, I presumed that the sample size to be sufficient for the purpose of completing this project as that is the amount of data provided for the sleep dataset.

Checking for duplicates:

```
sum(duplicated(Activity))

## [1] 0

sum(duplicated(Calories))

## [1] 0

sum(duplicated(Sleep))

## [1] 3

sum(duplicated(Steps))
```

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

We found out there were duplicates in the sleep dataset,The part of the cleaning process is to remove duplicate. This duplicate was removed by running this code:

```
Sleep <- Sleep %>% distinct() %>% drop_na()
```

To confirm that the duplicate has been removed, I checked for duplicates, again.

```
sum(duplicated(Activity))

## [1] 0

sum(duplicated(Calories))

## [1] 0

sum(duplicated(Sleep))

## [1] 0

sum(duplicated(Steps))

## [1] 0
```

Now that there are no more duplicates, I proceeded to ensure the data sets headers are consistent.

Checking for missing Values:

I checked for missing Values.

```
sum(is.na(Activity))

## [1] 0

sum(is.na(Calories))

## [1] 0

sum(is.na(Steps))

## [1] 0

sum(is.na(Sleep))

## [1] 0
```

There are no missing Values in the dataset.

Cleaning Headers:

```
activity <- clean_names(Activity)
Calories <- clean_names(Calories)
Sleep <- clean_names(Sleep)
Steps <- clean_names(Steps)
```

Formatting dates in the datasets:

I noticed some inconsistencies with the time stamp data. I converted the time stamp data into date- time format. Since none of the data frames I chose for this project has hourly data, I decided to use the "as_date" function because the data frames used consist of daily data.

To check the variable type/data type of the date to know if there's going to be a need to change the formatting, I used the str() function to check the variable type of the data sets and found out that in was stored in the character "chr". Also because its a csv file the Date variable is always mostly in Character "Chr" so we need to transform into a date variable type

```
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 ...

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 ...
##  $ activity_day: 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
...

str(Steps)

## 'data.frame':    940 obs. of  3 variables:
##  $ id          : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ activity_day: chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
```

```
## $ step_total : int 13162 10735 10460 9762 12669 9705 13019 15506 10544
9819 ...
```

```
str(Sleep)
```

```
## 'data.frame':    410 obs. of  5 variables:
##  $ id                  : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ sleep_day           : chr  "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00
AM" "4/15/2016 12:00:00 AM" "4/16/2016 12:00:00 AM" ...
##  $ total_sleep_records : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ total_minutes_asleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ total_time_in_bed   : int  346 407 442 367 712 320 377 364 384 449 ...
```

```
 Activity <- Activity %>%
  rename(date = ActivityDate) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))

Calories <- Calories %>%
  rename(date = activity_day) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))

Sleep <- Sleep %>%
  rename(date = sleep_day) %>%
  mutate(date = as_date(date,format ="%m/%d/%Y %I:%M:%S %p"))

Steps <- Steps %>%
  rename(date = activity_day) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))
```

Previewing the cleaned dataframes :

```
head(Activity)
```

```
##           Id       date TotalSteps TotalDistance TrackerDistance
## 1 1503960366 2016-04-12      13162          8.50            8.50
## 2 1503960366 2016-04-13      10735          6.97            6.97
## 3 1503960366 2016-04-14      10460          6.74            6.74
## 4 1503960366 2016-04-15       9762          6.28            6.28
## 5 1503960366 2016-04-16      12669          8.16            8.16
## 6 1503960366 2016-04-17       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(Calories)
```

```
##           id       date calories
## 1 1503960366 2016-04-12     1985
## 2 1503960366 2016-04-13     1797
## 3 1503960366 2016-04-14     1776
## 4 1503960366 2016-04-15     1745
## 5 1503960366 2016-04-16     1863
## 6 1503960366 2016-04-17     1728
```

```
head(Sleep)
```

```
##           id       date total_sleep_records total_minutes_asleep
## 1 1503960366 2016-04-12                   1                  327
## 2 1503960366 2016-04-13                   2                  384
## 3 1503960366 2016-04-15                   1                  412
## 4 1503960366 2016-04-16                   2                  340
## 5 1503960366 2016-04-17                   1                  700
## 6 1503960366 2016-04-19                   1                  304
##   total_time_in_bed
## 1               346
## 2               407
## 3               442
## 4               367
## 5               712
## 6               320
```

```
head(Steps)
```

```
##           id       date step_total
## 1 1503960366 2016-04-12      13162
## 2 1503960366 2016-04-13      10735
## 3 1503960366 2016-04-14      10460
## 4 1503960366 2016-04-15       9762
## 5 1503960366 2016-04-16      12669
## 6 1503960366 2016-04-17       9705
```

Merging Data

To visualize the data, I merged 2 dataframes , the Activity and the Sleep data frame on column id and I also merged the calories and steps. The data will be merged with the id and date fields as their primary

I had to change the colnames of the Activity data frames to lower case to allow easy merging of the dataframes. All other dataframes were in lowercase.

```
names(Activity) <- tolower(names(Activity))

Activity_Sleep <- merge(Activity, Sleep, by = c ("id", "date"))
n_distinct(Activity_Sleep$id)

## [1] 24

head(Activity_Sleep)

##           id       date totalsteps totaldistance trackerdistance
## 1 1503960366 2016-04-12      13162          8.50            8.50
## 2 1503960366 2016-04-13      10735          6.97            6.97
## 3 1503960366 2016-04-15       9762          6.28            6.28
## 4 1503960366 2016-04-16      12669          8.16            8.16
## 5 1503960366 2016-04-17       9705          6.48            6.48
## 6 1503960366 2016-04-19      15506          9.88            9.88
##   loggedactivitiesdistance veryactivedistance moderatelyactivedistance
## 1                        0               1.88                     0.55
## 2                        0               1.57                     0.69
## 3                        0               2.14                     1.26
## 4                        0               2.71                     0.41
## 5                        0               3.19                     0.78
## 6                        0               3.53                     1.32
##   lightactivedistance sedentaryactivedistance veryactiveminutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                2.83                       0                29
## 4                5.04                       0                36
## 5                2.51                       0                38
## 6                5.03                       0                50
##   fairlyactiveminutes lightlyactiveminutes sedentaryminutes calories
## 1                  13                  328              728     1985
## 2                  19                  217              776     1797
## 3                  34                  209              726     1745
## 4                  10                  221              773     1863
## 5                  20                  164              539     1728
## 6                  31                  264              775     2035
##   total_sleep_records total_minutes_asleep total_time_in_bed
## 1                   1                  327               346
## 2                   2                  384               407
## 3                   1                  412               442
## 4                   2                  340               367
```

```
## 5                   1                  700                712
## 6                   1                  304                320
```

```
Calories_Steps <- merge(Calories, Steps, by = c ("id", "date"))
n_distinct(Calories_Steps$id)
```

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

```
head(Calories_Steps)
```

```
##           id       date calories step_total
## 1 1503960366 2016-04-12     1985      13162
## 2 1503960366 2016-04-13     1797      10735
## 3 1503960366 2016-04-14     1776      10460
## 4 1503960366 2016-04-15     1745       9762
## 5 1503960366 2016-04-16     1863      12669
## 6 1503960366 2016-04-17     1728       9705
```

Analysing and visualising the data

Activity

I analyzed some trends of the FITBit users to provide some insights for the BellaBeat App in order to help in the marketing strategies and decision making.

According to the instruction given in the business question, I was encouraged to use other resources to help in my Analysis. So I decided to use resources from MedicineNet and article by Community Access Network to categorize and to know how many steps in a day are considered active.

Clasification of Activity.

- Sedentary: 5,000 or fewer
- Semi-Active: 5,000-7,499
- Somewhat Active: 7,500-9,999
- Active: 10,000-12,499
- Very Active: 12,500 or more

To make the above classifications, I will compute the average daily steps for each user

```
Average_Steps <- Activity_Sleep %>%
  group_by(id) %>%
  summarise(Average_Steps = mean(totalsteps))

head(Average_Steps)
```

```
## # A tibble: 6 × 2
##           id Average_Steps
##        <dbl>         <dbl>
## 1 1503960366        12406.
## 2 1644430081         7968.
```

```
## 3 1844505072         3477
## 4 1927972279         1490
## 5 2026352035         5619.
## 6 2320127002         5079
```

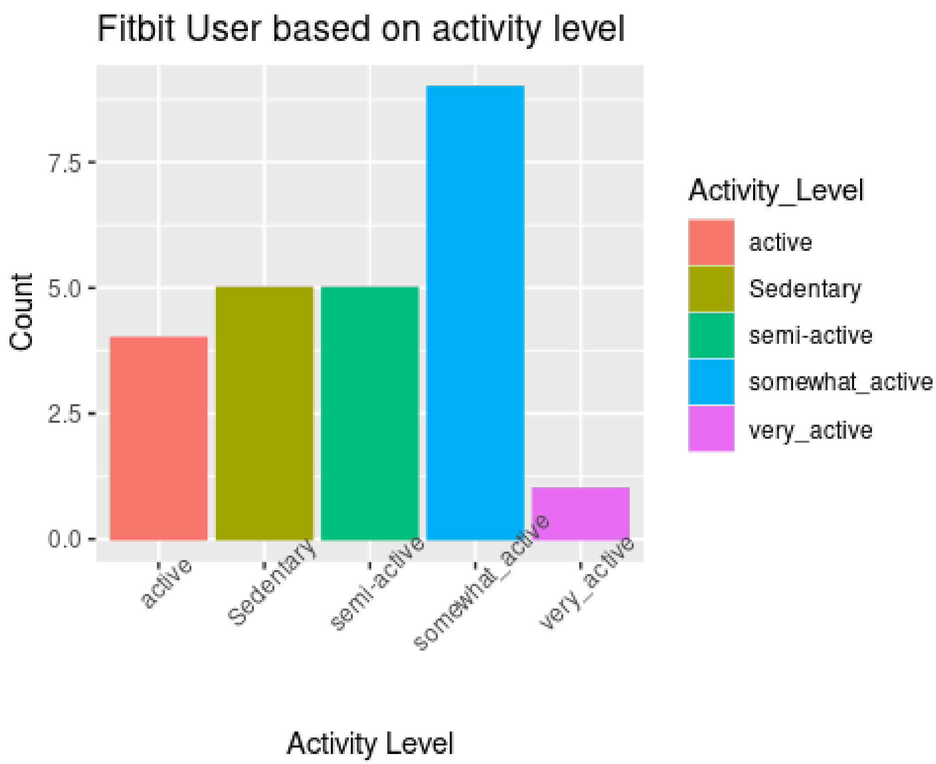
Categorizing each user by activity level.

```
Activity_Level <- Average_Steps %>%
  mutate(Activity_Level = case_when(
    Average_Steps < 5000 ~ "Sedentary",
    Average_Steps >= 5000 & Average_Steps < 7500 ~ "semi-active",
    Average_Steps >= 7500 & Average_Steps < 9999 ~ "somewhat_active",
    Average_Steps >= 10000 & Average_Steps < 12500 ~ "active", Average_Steps
>= 12500 ~ "very_active",

  ))

head(Activity_Level)
```

```
## # A tibble: 6 × 3
##           id Average_Steps Activity_Level
##        <dbl>         <dbl> <chr>
## 1 1503960366        12406. active
## 2 1644430081         7968. somewhat_active
## 3 1844505072         3477  Sedentary
## 4 1927972279         1490  Sedentary
## 5 2026352035         5619. semi-active
## 6 2320127002         5079  semi-active
```

I proceeded to visualize the classification of users via a barchart.

```
ggplot(data=Activity_Level, mapping=aes(x = Activity_Level, colour =
Activity_Level, fill = Activity_Level)) +geom_bar() + theme(axis.text.x =
element_text(angle = 45)) +labs( x = "Activity Level", y = "Count", title =
"Fitbit User based on activity level")
```

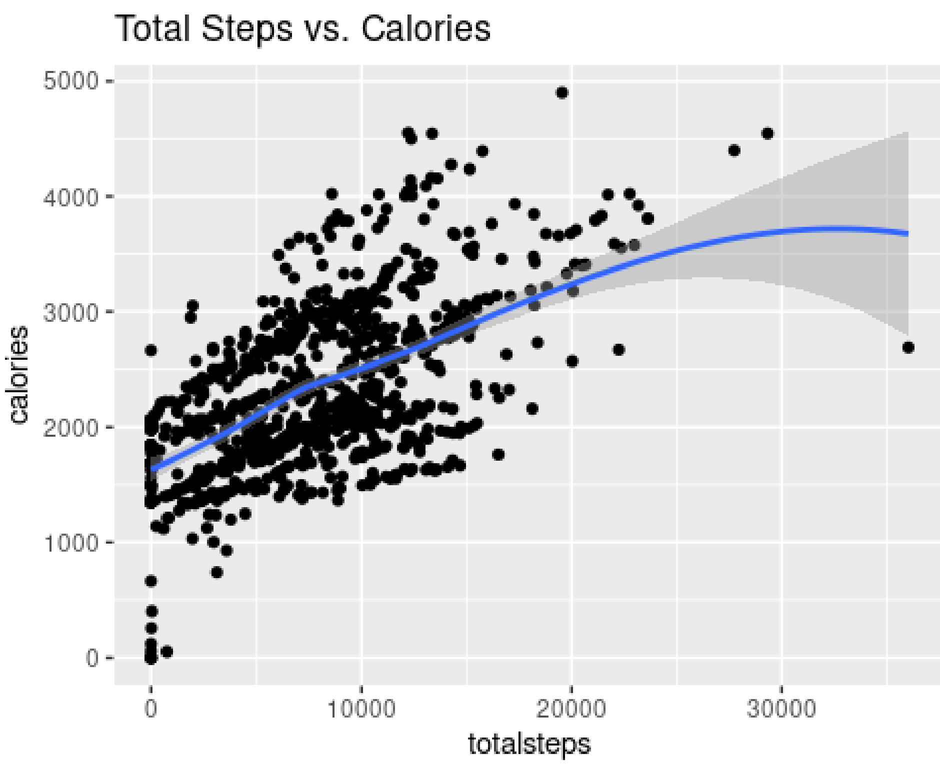


The chart above tells us that even though we have different types of users using the smart device, the "some-what active" users with (7,500-9,999 steps) are the most/highest users of the smart device.

Relationship between Steps and Sedentary time

The relationship between total steps taken in a day and Calories.

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

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



This plot clearly states that the more the steps the more the calories burnt, the lesser the steps taken, the lesser the calories burnt.

Sleep

I categorized users based on sleep minute totals and how they can be defined according tohealth.org. According to the article, It's important to get enough sleep. Sleep helps keep your mind and body healthy. Most adults need 7 or more hours of good-quality sleep on a regular schedule each night. Getting enough sleep isn't only about total hours of sleep. It's also important to get good-quality sleep on a regular schedule so you feel rested when you wake up. The National Sleep Foundation recommends 450 minutes of sleep average for a good night's rest.

```
Average_Sleep <- Activity_Sleep %>%
  group_by(id) %>%
  summarise(Average_Sleep = mean(total_minutes_asleep))

head(Average_Sleep)

## # A tibble: 6 × 2
##            id Average_Sleep
##         <dbl>         <dbl>
## 1 1503960366          360.
## 2 1644430081          294
## 3 1844505072          652
## 4 1927972279          417
```

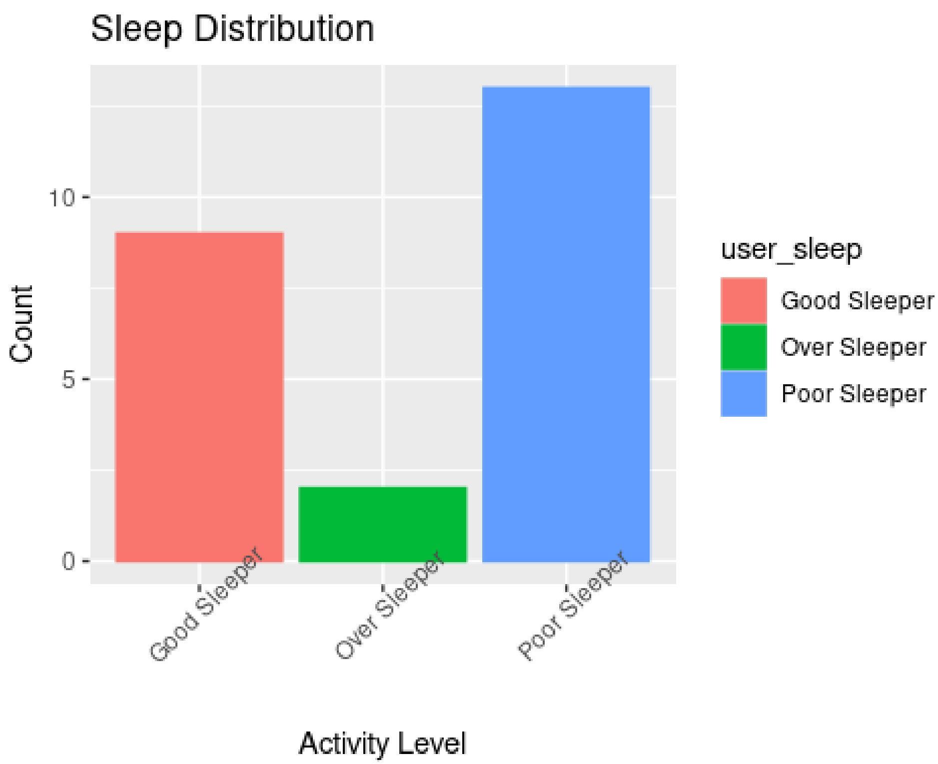
```
## 5 2026352035          506.
## 6 2320127002           61
```

With the average sleep calculated per user, I categorized the users based on their average sleep times.

```
Sleep_Activity <- Average_Sleep %>%
  mutate(user_sleep = case_when(
    Average_Sleep < 420 ~ "Poor Sleeper",
    Average_Sleep >= 420 & Average_Sleep <= 480 ~ "Good Sleeper",
    Average_Sleep > 480 ~ "Over Sleeper",
  ))

head(Sleep_Activity)
```

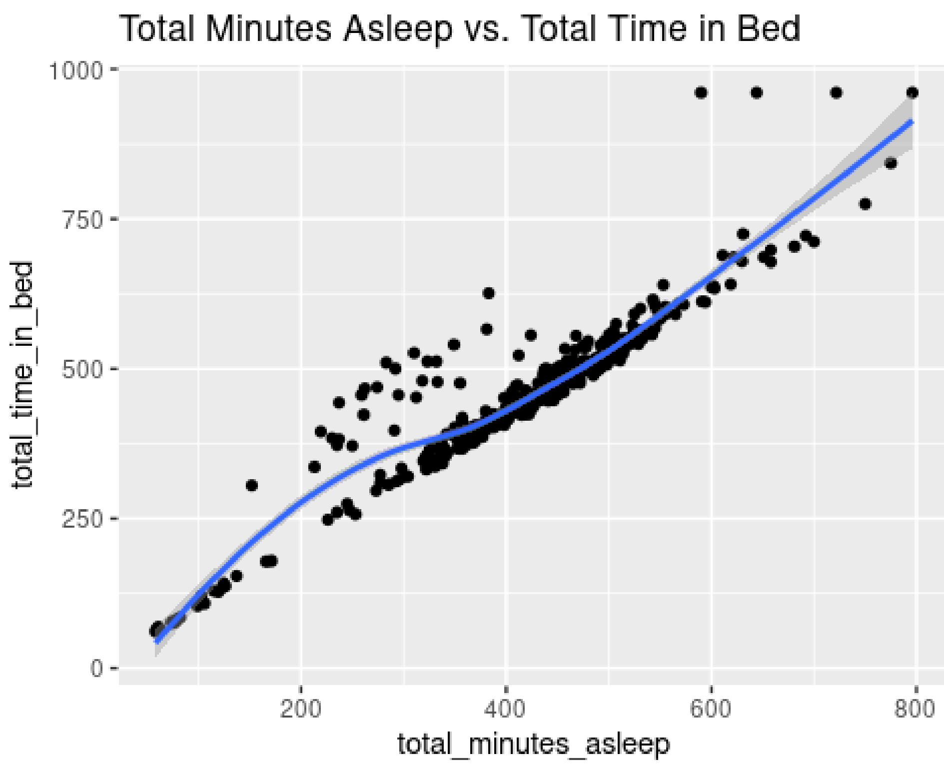
```
## # A tibble: 6 × 3
##           id Average_Sleep user_sleep
##        <dbl>         <dbl> <chr>
## 1 1503960366          360. Poor Sleeper
## 2 1644430081          294  Poor Sleeper
## 3 1844505072          652  Over Sleeper
## 4 1927972279          417  Poor Sleeper
## 5 2026352035          506. Over Sleeper
## 6 2320127002           61  Poor Sleeper
```

```
ggplot(data=Sleep_Activity, mapping=aes(x = user_sleep, colour = user_sleep,
fill = user_sleep)) +geom_bar() + theme(axis.text.x = element_text(angle =
45)) +labs( x = "Activity Level", y = "Count", title = "Sleep Distribution")
```



This chart shows that there are more users with poor sleeping patterns.

```
ggplot(data=Sleep, aes(x=total_minutes_asleep, y=total_time_in_bed)) +
geom_point()+ geom_smooth() + labs(title="Total Minutes Asleep vs. Total Time
in Bed")

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



This charts shows the longer you stay in bed, the longer you sleep, which is not enetirely true. I recommend that there should be a reminder to remind users when to sleep, instead of staying in bed and not sleeping.

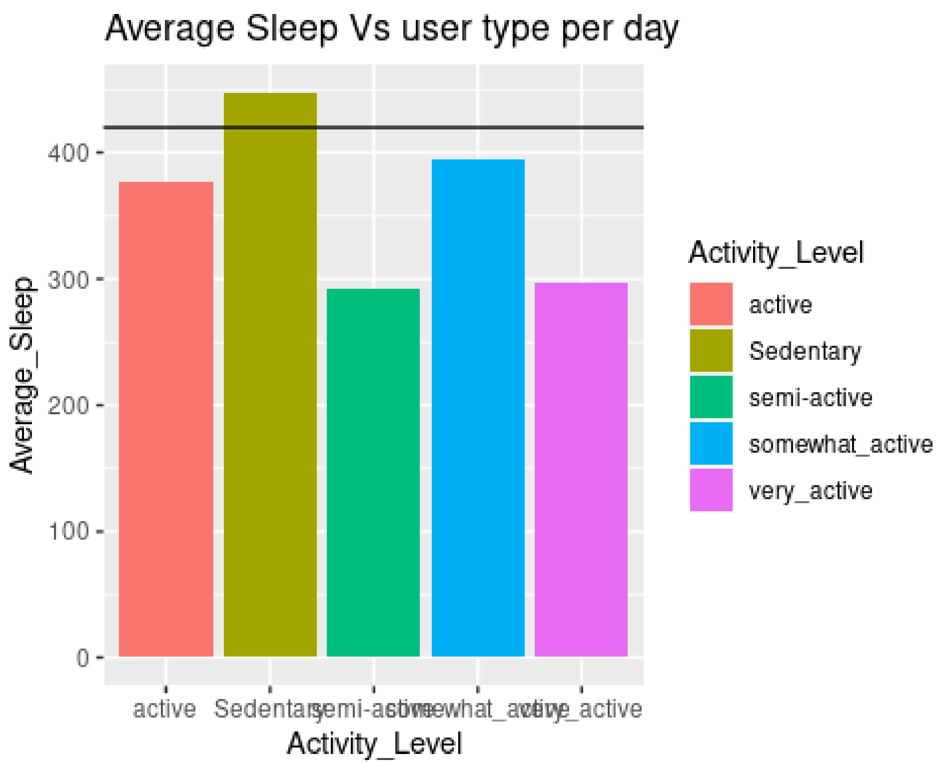
Average sleep per activity level

I proceeded to check the average sleep per activity level

```
Sleep_Analytics <- merge(Average_Sleep, Activity_Level, by = c ("id"))
n_distinct(Sleep_Analytics$id)

## [1] 24

head(Sleep_Analytics)

##           id Average_Sleep Average_Steps  Activity_Level
## 1 1503960366      360.2800     12405.680          active
## 2 1644430081      294.0000      7967.750 somewhat_active
## 3 1844505072      652.0000      3477.000       Sedentary
## 4 1927972279      417.0000      1490.000       Sedentary
## 5 2026352035      506.1786      5618.679     semi-active
## 6 2320127002       61.0000      5079.000     semi-active

Sleep_Actv_Analytics <- Sleep_Analytics %>% group_by(Activity_Level) %>%
summarise(Average_Sleep = mean(Average_Sleep))

head(Sleep_Actv_Analytics)
```

```
## # A tibble: 5 × 2
##   Activity_Level  Average_Sleep
##   <chr>                   <dbl>
## 1 Sedentary                448.
## 2 active                   377.
## 3 semi-active              292.
## 4 somewhat_active          395.
## 5 very_active              297
```

```
ggplot()+ geom_col(Sleep_Actv_Analytics, mapping = aes(x=Activity_Level,
y=Average_Sleep , fill = Activity_Level)) + labs(title="Average Sleep Vs user
type per day") + geom_hline(yintercept = 420)
```



From the chart above, it shows that Sedentary users have the most amount of sleep while the semi_active users have the least amount of sleep followed by the very active users. By normal standard the very_active users should have more amount of sleep because people get tired by activities done. Factors that might hinder this could be, the users are very used to their way of life, thereby being very active daily, could also be due to caffeine consumption, anxiety, sleeping-disorders, etc.

Recommendation.

Since we are not given enough datasets. Some research into exercise, diet, age and menstrual cycle could have given some insight which can help with better understanding of the project, Which could lead to more insights. However, with the datasets provided and the analysis done I was able to find some insight which the Bellabeat App can be built upon.

I recommend that Bellabeat should introduce some of this features to improve the *Bellabeat App* and market the app using the features suggested below :

- Calories and steps notification: User should set an activity goal and reminders at intervals. With this the user will be more committed towards this goal. These features should be voluntary though, but the App should send motivating notifications, compelling enough to make users committed to this Activity. The notification reminders should also show the daily progress of the user in terms of steps taken and calories burnt.

- Sleep Reminders/trackers : Users should be asked to set a sleep goal which includes bedtime and duration of sleep. The essence of having an effective Sleep and quality should be mentioned as the Sleep notifications pops up. Users can be asked to rate how fresh they feel every morning so that they can relate it to the sleep quality of the previous night.

- Activities Challenge: There should be a Challenge on the App in which users can participate. At the end of this challenge (this will be based on the decision made by the leadership and stakeholder of the App), a price(Free products and or free subscription should be given to user who complete the activity goals/ records set by the company to compel user to use the Bellabeat App which can also be used as a Marketing Strategies.

Conclusively, I believe from the above analysis and recommendations suggested, the Bellabeat App and its customers will benefit greatly from this. Bellabeat should propose Marketing strategies towards its target audience, which should users with least Activity level and the poor and over sleepers the features should also be introduced to new features to existing and potential customers.