Data Analytics Case Study with R programming

How Can a Wellness Technology Company Play It Smart?

1 Introduction

Hello, welcome to my Bellabeat data analytics case study. As a junior data analyst, I will show the real-world tasks I performed in this project. The data collected for this study is from Bellabeat, a wellness technology company with a high-tech manufacturer of health-focused products for women and different characters. To complete this capstone project, I will follow the steps of the data analysis process: ask, prepare, process, analyze, share, and act.

1.1 About The Company

Bellabeat was founded in 2013 by Urška Sršen and Sando Mur. They aimed to manufacture health-focused small products. Sršen used her background to design technology that inspires women around the world. Collecting data on activity, sleep, stress, and reproductive health has allowed Bellabeat to empower women with knowledge about their health and habits. By 2016, the company opened several offices in many countries and launched multiple products through online retailers and e-commerce channels.

1.2 Objective of The Study

The goal is to analyze "smart device fitness" data to gain insights into how consumers use their devices. The insights will then be helpful to make a data drive-decision making for the company. Eventually, The analysis will be presented to Bellabeat executive team to guide their marketing strategy.

2 Ask

2.1 Beginning of The Analysis Process

Ask questions that guide us to the solution and give details about the information we want to find.

- What is problem are we trying to solve?
  - Bellabeat wants to be a player in the global smart-device market; therefore, the task is to analyze smart device fitness data to check how consumers use their devices.
- What kind of data is available?

- The collected “FitBit Fitness Tracker Data” is from a public dataset obtained from the “Kaggle” website containing a personal fitness tracker from thirty FitBit users. The users’ information is about: heart rate, sleep monitoring, physical activity, steps, and daily activity.

- Who are the stakeholders?
  - The main stakeholders are Urška Sršen, Bellabeat’s co-founder and Chief Creative Officer; Sando Mur, Mathematician and Bellabeat’s co-founder. The analysis also will be presented to the marketing analytics team.

2.2 Business Task

After understanding the contents of the data, the insights will help us come up with recommendations to attract users to use Bellabeat’s devices.

3 Prepare

The data will be downloaded and imported to check its credibility and integrity

3.1 Downloading The Data

Bellabeat’s co-founder Sršen encourages the usage of the public data “FitBit Fitness Tracker Data” (CC0: Public Domain, dataset made available through Mobius) to explore smart-device users’ daily habits. It includes information about thirty users regarding their health habits. The link to the database is below:

Fitbit Fitness Tracker Data: `https://www.kaggle.com/arashnicfitbit`

The dataset came from a survey on Amazon Mechanical Turk between the period 03.12.2016 - 05.12.2016. It includes eighteen CSV files.

3.2 Data Pre-processing

3.2.1 Importing The Dataset

The available files will be imported to R and continue our analysis through RStudio since spreadsheets cannot deal with big data, and visualizations are hard on SQL. The 18 datasets have information about activities in the format of days, hours, and minutes.

3.2.2 Loading Needed Packages

First, install the mandatory packages to get the functions we need to use. The most famous packages are (readr, tidyverse, skimr, janitor, RColorBrewer, and ggplot2).

```
#Import needed packages

library(readr)

library(tidyverse)
```

library(skimr)

library(janitor)

library(RColorBrewer)

library(ggplot2)

#Importing datasets to R

Daily_Activity <- read.csv(“dailyActivity_merged.csv”)

Daily_Calories <- read.csv(“dailyCalories_merged.csv”)

Daily_Intensities <- read.csv(“dailyIntensities_merged.csv”)

Daily_Steps <- read.csv(“dailySteps_merged.csv”)

Hourly_Calories <- read.csv(“hourlyCalories_merged.csv”)

Hourly_Intensities <- read.csv(“hourlyIntensities_merged.csv”)

Hourly_Steps <- reead.csv(“hourlySteps_merged.csv”)

Sleep_Day <- read.csv(“sleepDay_merged.csv”)

Heart_Rate <- read.csv(“heartrate_seconds_merged.csv”)

Weight <- read.csv(“weightLogInfo_merged.csv”)

#Summary for the following datasets

glimpse(Daily_Activity)

glimpse(Hourly_Calories)

glimpse(Sleep_Day)

OUTPUT > glimpse(Daily_Activity)

Rows: 940

Columns: 15

$ Id < *dbl* > 1503960366, 1503960366, 1503960366, 1503960366, 1503960366, 15…

$ ActivityDate < *dbl* > “4/12/2016”, “4/13/2016”, “4/14/2016”,“4/15/2016”, “4/16/2016”…

$ TotalSteps < *dbl* > 13162, 10735, 10460, 9762, 12669, 9705, 13019, 15506, 10544, 9…

$ TotalDistance < *dbl* > 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.88, 6.68, 6.34, 8.…

$ TrackerDistance < *dbl* > 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.88, 6.68, 6.34, 8.…

$ LoggedActivitiesDistance < *dbl* > 0, 0, 0, 0, 0, 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, 3.53, 1.96, 1.34, 4.…

$ ModeratelyActiveDistance < *dbl* > 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.32, 0.48, 0.35, 1.…

$ LightActiveDistance < *dbl* > 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.03, 4.24, 4.65, 2.…

$ SedentaryActiveDistance < *dbl* > 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…

$ VeryActiveMinutes < *dbl* > 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 41, 39, 73, 31, 78…

$ FairlyActiveMinutes < *dbl* > 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21, 5, 14, 23, 11, …

$ LightlyActiveMinutes < *dbl* > 328, 217, 181, 209, 221, 164, 233, 264, 205, 211, 130, 262, 23…

$ SedentaryMinutes < *dbl* > 728, 776, 1218, 726, 773, 539, 1149, 775, 818, 838, 1217, 732,…

$ Calories < *dbl* > 1985, 1797, 1776, 1745, 1863, 1728, 1921, 2035, 1786, 1775, 18…

> glimpse(Hourly_Calories)

Rows: 22,099

Columns: 3

$ Id < *dbl* > 1503960366, 1503960366, 1503960366, 1503960366, 1503960366, 1503960366, 15…

$ ActivityHour < *chr* > “4/12/2016 12:00:00 AM”, “4/12/2016 1:00:00 AM”, “4/12/2016 2:00:00 AM”, …

$ Calories < *dbl* > 81, 61, 59, 47, 48, 48, 48, 47, 68, 141, 99, 76, 73, 66, 110, 151, 76, 83,…

> glimpse(Sleep_Day)

Rows: 413

Columns: 5

$ Id < *dbl* > 1503960366, 1503960366, 1503960366, 1503960366, 1503960366, 15039603…

$ SleepDay < *chr* > “4/12/2016 12:00:00 AM”, “4/13/2016 12:00:00 AM”, “4/15/2016 12:00:0”...

$ TotalSleepRecords < *dbl* > 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...

$ TotalMinutesAsleep < *dbl* > 327, 384, 412, 340, 700, 304, 360, 325, 361, 430, 277, 245, 366, 341...

$ TotalTimeInBed < *dbl* > 346, 407, 442, 367, 712, 320, 377, 364, 384, 449, 323, 274, 393, 354...

> str(Daily_Activity)

str(Hourly_Calories)

str(Sleep_Day)

OUTPUT > str(Daily_Activity)

spec_tbl_df [940 × 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)

$ Id : num [1:940] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...

$ ActivityDate : chr [1:940] “4/12/2016” “4/13/2016” “4/14/2016” “4/15/2016” ...

$ TotalSteps : num [1:940] 13162 10735 10460 9762 12669 ...

$ TotalDistance : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...

$ TrackerDistance : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...

$ LoggedActivitiesDistance: num [1:940] 0 0 0 0 0 0 0 0 0 0 ...

$ VeryActiveDistance : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...

$ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...

$ LightActiveDistance : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...

$ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...

$ VeryActiveMinutes : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...

$ FairlyActiveMinutes : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...

$ LightlyActiveMinutes : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...

$ SedentaryMinutes : num [1:940] 728 776 1218 726 773 ...

$ Calories : num [1:940] 1985 1797 1776 1745 1863 ...

- attr(*, “spec”)=

.. cols(

.. Id = col_double(),

.. ActivityDate = col_character(),

.. TotalSteps = col_double(),

.. TotalDistance = col_double(),

.. TrackerDistance = col_double(),

.. LoggedActivitiesDistance = col_double(),

.. VeryActiveDistance = col_double(),

.. ModeratelyActiveDistance = col_double(),

.. LightActiveDistance = col_double(),

.. SedentaryActiveDistance = col_double(),

.. VeryActiveMinutes = col_double(),

.. FairlyActiveMinutes = col_double(),

.. LightlyActiveMinutes = col_double(),

.. SedentaryMinutes = col_double(),

.. Calories = col_double()

..)

- attr(*, “problems”)=<externalptr>

> str(Hourly_Calories)

spec_tbl_df [22,099 × 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)

$ Id : num [1:22099] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...

$ ActivityHour: chr [1:22099] “4/12/2016 12:00:00 AM” “4/12/2016 1:00:00 AM” “4/12/2016

2:00:00 AM” “4/12/2016 3:00:00 AM” ...

$ Calories : num [1:22099] 81 61 59 47 48 48 48 47 68 141 ...

- attr(*, “spec”)=

.. cols(

.. Id = col_double(),

.. ActivityHour = col_character(),

.. Calories = col_double()

..)

- attr(*, “problems”)=<externalptr>

> str(Sleep_Day)

spec_tbl_df [413 × 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)

$ Id : num [1:413] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...

$ SleepDay : chr [1:413] “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” ...

$ TotalSleepRecords : num [1:413] 1 2 1 2 1 1 1 1 1 1 ...

$ TotalMinutesAsleep: num [1:413] 327 384 412 340 700 304 360 325 361 430 ...

$ TotalTimeInBed : num [1:413] 346 407 442 367 712 320 377 364 384 449 ...

- attr(*, “spec”)=

.. cols(

.. Id = col_double(),

.. SleepDay = col_character(),

.. TotalSleepRecords = col_double(),

.. TotalMinutesAsleep = col_double(),

.. TotalTimeInBed = col_double()

..)

- attr(*, "problems")=<externalptr>

4 Process

4.1 Verification

The following step shows the number of available participants and how many times were they recorded in each dataset.

#Count the number of IDs repeated in the datasets

Activity_ID_Count <- Daily_Activity %>%

group_by(Id) %>%

summarize(count = n())

Activity_ID_Count

A tibble: 33 × 2

Id count

$< dbl > < int >$

1 1503960366 31

2 1624580081 31

3 1644430081 30

4 1844505072 31

5 1927972279 31

6 2022484408 31

7 2026352035 31

8 2320127002 31

9 2347167796 18

10 2873212765 31

… with 23 more rows

Calories_ID_Count <- Hourly_Calories %>%

group_by(Id) %>%

summarize(count = n())

Calories_ID_Count

A tibble: 33 × 2

Id count

$< dbl > < int >$

1 1503960366 717

2 1624580081 736

3 1644430081 708

4 1844505072 731

5 1927972279 736

6 2022484408 736

7 2026352035 736

8 2320127002 735

9 2347167796 414

10 2873212765 736

… with 23 more rows

Sleep_ID_Count <- Sleep_Day %>%

group_by(Id) %>%

summarize(count = n())

Sleep_ID_Count

A tibble: 24 × 2

Id count

$< dbl > < int >$

1 1503960366 25

2 1644430081 4

3 1844505072 3

4 1927972279 5

5 2026352035 28

6 2320127002 1

7 2347167796 15

8 3977333714 28

9 4020332650 8

10 4319703577 26

... with 14 more rows

HeartRate_ID_Count <- Heart_Rate %>%

group_by(Id) %>%

summarize(count = n())

HeartRate_ID_Count #Only 14 records are available

A tibble: 14 × 2

Id count

$< dbl > < int >$

1 2022484408 154104

2 2026352035 2490

3 2347167796 152683

4 4020332650 285461

5 4388161847 249748

6 4558609924 192168

7 5553957443 255174

8 5577150313 248560

9 6117666160 158899

10 6775888955 32771

11 6962181067 266326

12 7007744171 133592

13 8792009665 122841

14 8877689391 228841

Weight_ID_Count <- Weight %>%

group_by(Id) %>%

summarize(count = n())

Weight_ID_Count #Only 8 participants had their weights recorded

A tibble: 8 × 2

Id count

$< dbl > < int >$

1 1503960366 2

2 1927972279 1

3 2873212765 2

4 4319703577 2

5 4558609924 5

6 5577150313 1

7 6962181067 30

8 8877689391 24

For the “Heart Rate” dataset, only 14 participants had their records available. For the “Weight” dataset, only eight records were in the data. For that reason, it is better not to consider these datasets in the analysis phase.
Next, the date and time in the datasets has to be separated into individual columns to work better with the data.

#Split the date column into “date & time” columns in the following datasets.

Hourly_Intense <- Hourly_Intensities %>%

extract(ActivityHour, c(“Date”, “Hour”), “([∧]+) (.*)”)

Hourly_Calorie <- Hourly_Calories %>%

extract(ActivityHour, c(“Date”, “Hour”), “([∧]+) (.*)”)

Hourly_Step <- Hourly_Steps %>%

extract(ActivityHour, c(“Date”, “Hour”), “([∧]+) (.*)”)

5 Analyze

In this phase, the recorded time for each data needs to have the same unit; therefore, there is a newly created dataset to show how many hours participants sleep and stay in bed. The conversion will happen from minutes into hours and set the results in a new table.

#Analyze phase, conversion of minutes asleep into hours

Hours_sleep <- Sleep_Day %>%

mutate(Total_Hours_Asleep = TotalMinutesAsleep/60,

Total_Hours_in_Bed = TotalTimeInBed/60) %>%

extract(SleepDay, c(“Date”, “Hour”), “([∧]+) (.*)”) %>%

select(-TotalSleepRecords, -TotalMinutesAsleep, -TotalTimeInBed, -Hour)

To better understand the recordings, new tables will be created for each dataset to show the mean values of their habits, consumed calories, sleep, taken steps, etc. The new recorded values will illustrate all the ID numbers and record the average of people’s habits.

#Mean values of the following datasets

```
AVG_Daily_Calories <- Daily_Calories %>%

group_by(Id) %>%

summarize(AverageBurnedCaloriesDaily = mean(Calories))

AVG_Daily_Steps <- Daily_Steps %>%

group_by(Id) %>%

summarize(AverageTakenStepsDaily = mean(StepTotal))

AVG_Hourly_Calories <- Hourly_Calorie %>%

group_by(Id, Hour) %>%

summarize(AverageBurnedCaloriesHourly = mean(Calories))

‘summarise()‘ has grouped output by ’Id’. You can override using the ‘.groups‘

argument.

>

AVG_Hourly_Steps <- Hourly_Step %>%

group_by(Id, Hour) %>%

summarize(AeravgeStepsHourly = mean(StepTotal))

‘summarise()‘ has grouped output by ’Id’. You can override using the ‘.groups‘

argument.

>

AVG_Hourly_Intense < Hourly_Intense %>%

group_by(Id, Hour) %>%

summarize(AverageIntenseHourly = mean(AverageIntensity))

‘summarise()‘ has grouped output by ’Id’. You can override using the ‘.groups‘

argument.
```

```
>

AVG_Hours_Asleep <- Hours_sleep %>%

group_by(Id) %>%

summarize(AverageHoursSleeping = mean(Total_Hours_Asleep))

AVG_Sleep_Dates <- Hours_sleep %>%

group_by(Date) %>%

summarize(AverageHoursSleeping = mean(Total_Hours_Asleep))

AVG_BPM <- Heart_Rate %>%

group_by(Id) %>%

summarize(BPM = mean(Value))
```

Note that, the heart rate dataset has missing values, as not all participants had records for their BPM.

5.1 Analyze and Share Phase (Data Analysis and Visualization)

Next is an important information to know about, which is for how many days the users did not use their BellaBeat product and calculate their days off in percentage term.

```
#% of how many days users were not using the product

Percentage_days_off <- Daily_Activity %>%

filter(SedentaryMinutes == 1440) %>%

group_by(Id) %>%

summarize(days_off = n(),

percent_days_off = (days_off/31)*100) %>%

print()

# A tibble: 17 × 3

Id days_off percent_days_off
```

$< dbl >< int >< dbl >$

1 1503960366 1 3.23

2 1844505072 9 29.0

3 1927972279 13 41.9

4 4020332650 14 45.2

5 4057192912 1 3.23

6 4319703577 1 3.23

7 4388161847 1 3.23

8 4702921684 1 3.23

9 5577150313 2 6.45

10 6117666160 5 16.1

11 6290855005 4 12.9

12 6775888955 9 29.0

13 7007744171 1 3.23

14 7086361926 1 3.23

15 8253242879 1 3.23

16 8583815059 6 19.4

17 8792009665 9 29.0

ggplot(Percentage_days_off, aes(x = percent_days_off)) +

geom_histogram(bins = 15, fill = “cyan”, color = “black”) +

xlab(“Days off taken by participants in %”) +

ylab(“Number of participants”) +

labs(title = “Percentage of days off for participants”)

Starting from here, the results of the analysis are visualized

#Visualizing the results

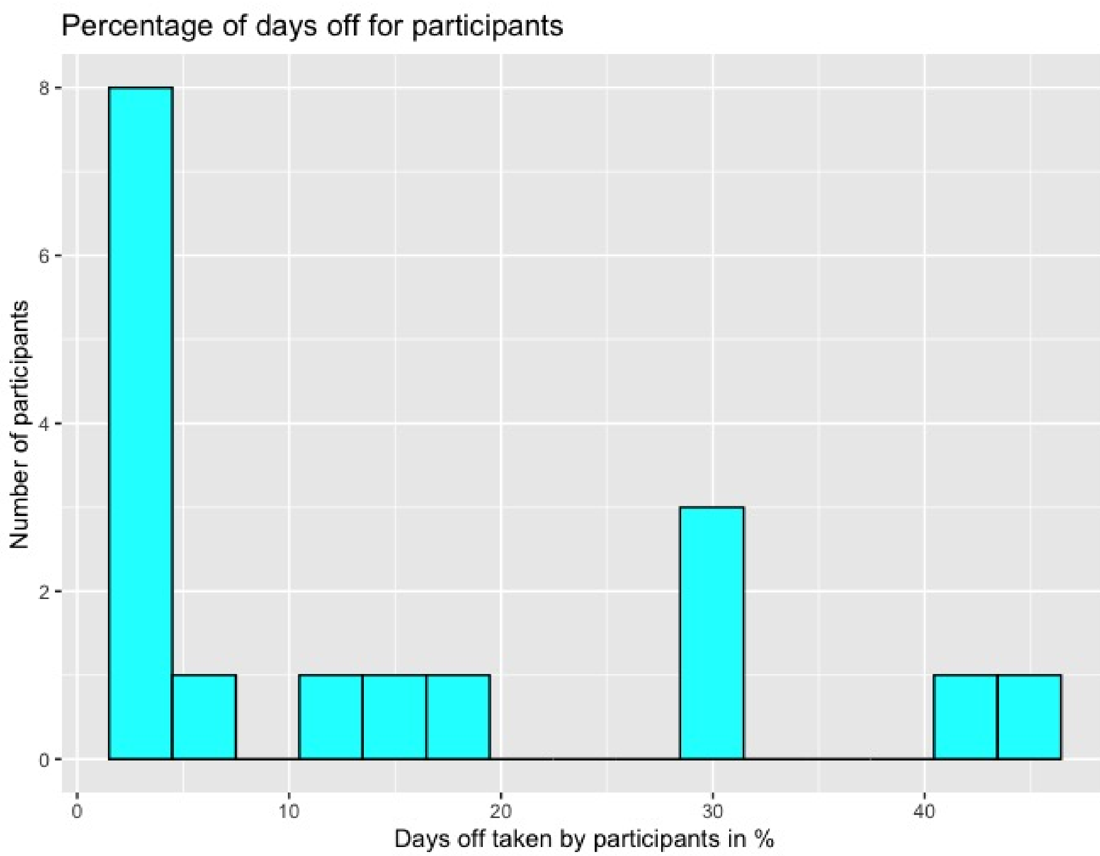

Figure 1: Percentage of Days Off.

#Creating a pie chart

piechart <- Percentage_days_off %>%

group_by(days_off) %>%

summarize(num_of_participants = n())

Plot of users activity

slices <- c(48, 24, 12, 16)

lbls <- c(“Used the product daily”, “Did not use for 1 day”, “Did not use for 1-7 days”, “Did not use for more than 7 days”)

pct <- round(slices/sum(slices)*100)

lbls <- paste(lbls, pct) # add percents to labels

lbls <- paste(lbls,“%”,sep=“ ”) # add % to labels

pie(slices, labels = lbls, col = blues9,

main = “Fitbit Usage”)

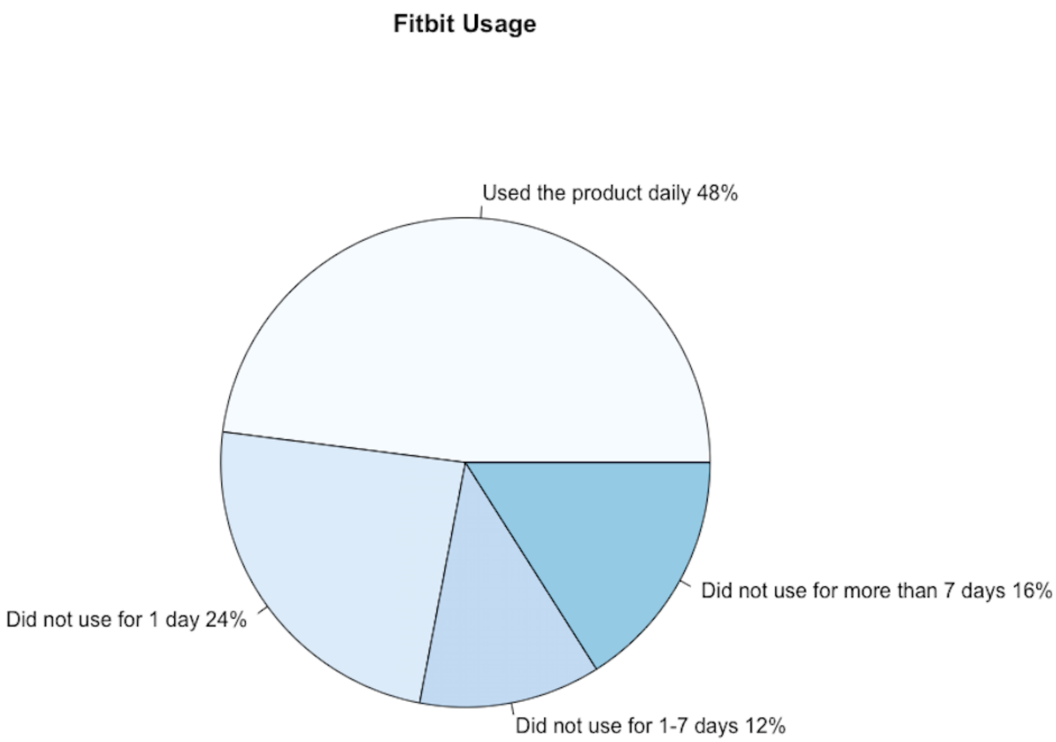

Figure 2: Users’ Activity for FitBit Product

The next created table will sum up the Daily_Activity dataset and will show the total amount of participants’ activity according to each category.

#Create a new table to categorize participants activity in total and percentage terms

Activity_of_Participants <- Daily_Activity %>%

filter(SedentaryMinutes != 1440) %>%

group_by(Id) %>%

summarize(Total_Very_Active_Minutes = sum(VeryActiveMinutes),

Total_Fairly_Active_Minutes = sum(FairlyActiveMinutes),

```
Total_Lightly_Active_Minutes = sum(LightlyActiveMinutes),

Total_Sendentary_Minutes = sum(SedentaryMinutes),

Total_Minutes = sum(VeryActiveMinutes, FairlyActiveMinutes,

LightlyActiveMinutes, SedentaryMinutes),

Percentage_Very_Active = (Total_Very_Active_Minutes/Total_Minutes)*100,

Percentage_Fairly_Active = (Total_Fairly_Active_Minutes/Total_Minutes)*100,

Percent_Lightly_Active =(Total_Lightly_Active_Minutes/Total_Minutes)*100,

Percent_Sendentary_Active = (Total_Sendentary_Minutes/Total_Minutes)*100)
```

A new column is introduced to the new dataset to distinguish between participants according to their activity type and habit.

```
#Create a new table to differentiate between participants intense workout

Participants_Activity <- Activity_of_Participants %>%

mutate(Intensity =

case_when(Percentage_Very_Active > mean(Percentage_Very_Active) ~“Very Active”,

Percentage_Fairly_Active > mean(Percentage_Fairly_Active) ~“Fairly Active”,

Percent_Lightly_Active > mean(Percent_Lightly_Active) ~“Lightly Active”,

Percent_Sendentary_Active > mean(Percent_Sendentary_Active) ~“Sendentary Active”))
```

The final table will sum up the results and show how many participants were very, fairly, lightly, or sedentary active.

```
#New table to show how many participant in each category

Activity <- Participants_Activity %>%

group_by(Intensity) %>%

summarize(count = n())
```

5.2 Results of The Comparison

The following visualizations will show the results of the analysis according to people's activity, eating and sleeping habits daily and hourly.

#Comparing participants according to daily activity

ggplot(Activity, aes(x = Intensity, y = count)) +

geom_histogram(stat = "identity", col = "red", fill = "lightblue") +

ylab("Number of Participants") +

xlab("Types of Intensity") +

labs(title = "Number of Participants by Intensity")

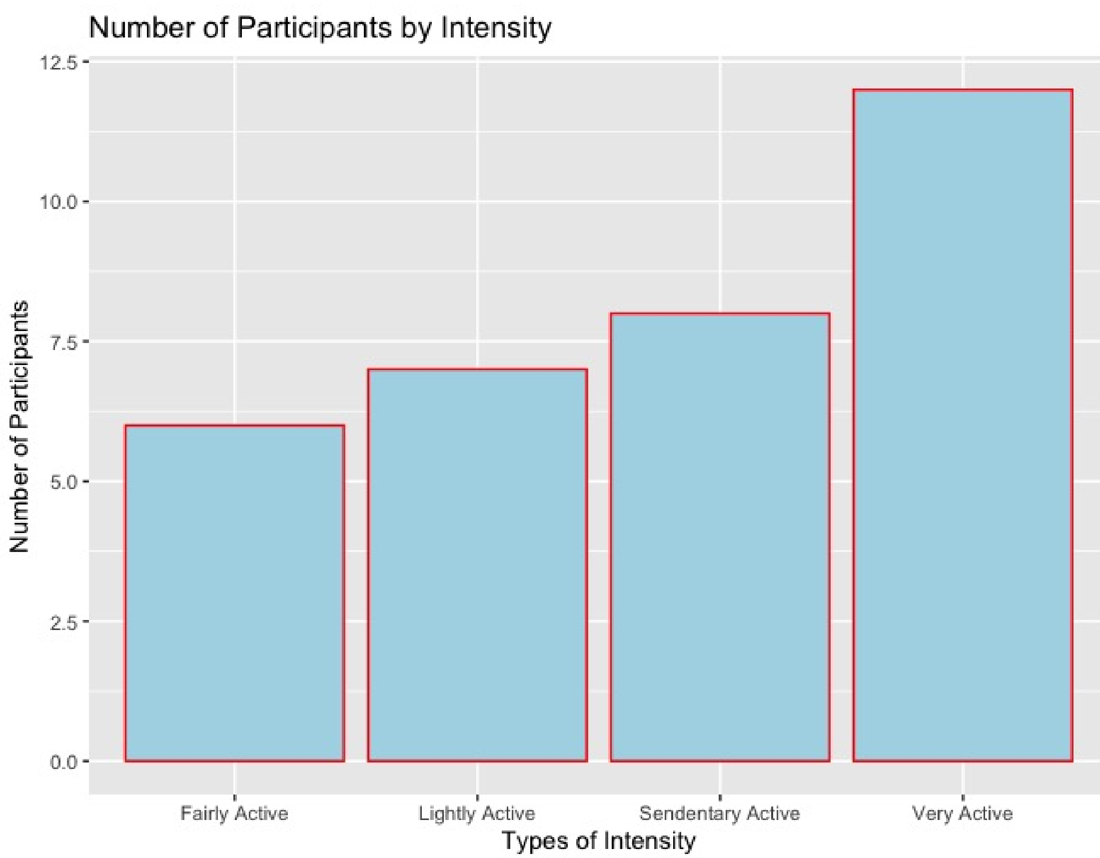

Figure 3: Participants According to Intensity Type

ggplot(AVG_Daily_Calories, aes(x = AverageConsumedCaloriesDaily)) +

geom_histogram(bins = 8, fill = "magenta", color = "black") +

ylab("Number of Participants") +

xlab("Average Consumed Calories") +

labs(title = "Average Calories Daily")

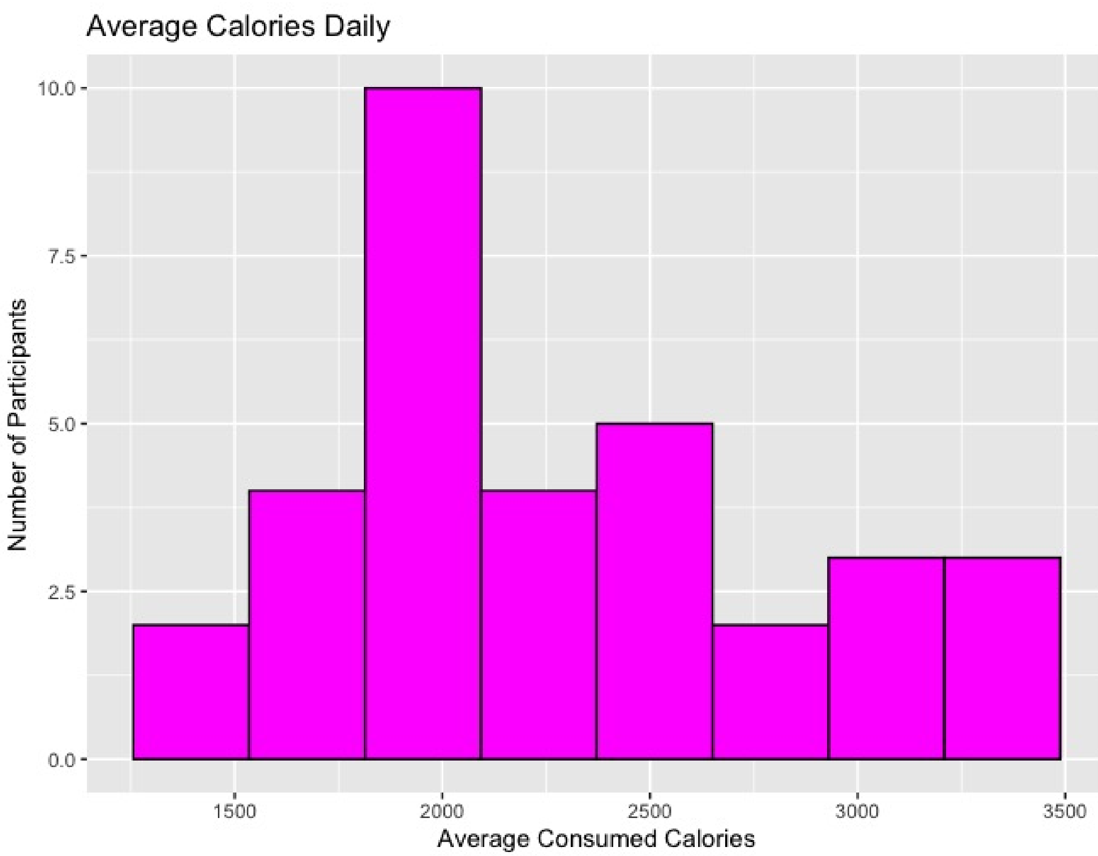

Figure 4: Average Calories Daily

ggplot(AVG_Daily_Steps, aes(x = AverageTakenStepsDaily)) +

geom_histogram(bins = 8, fill = "purple", color = "black") +

ylab("Number of Participants") +

xlab("Average Taken Steps") +

labs(title = "Average Taken Steps Daily")

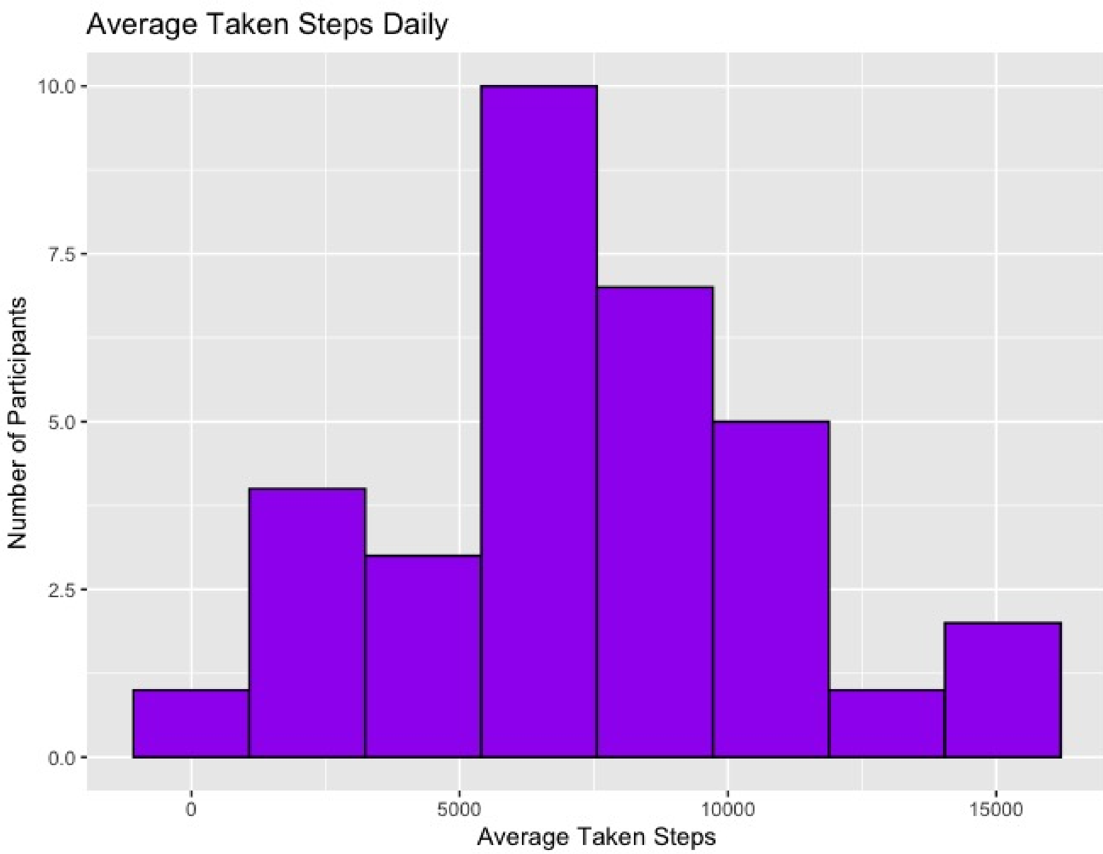

Figure 5: Average Steps Daily

Previous plots indicate the activities done daily by people, next graphs will show how participants' activities differ hourly during the day. The graphs provide us with information about which hours participants are active and their daily habits.

```
#Showing the activities in each hour per day

ggplot(AVG_Hourly_Calories, aes(x = AverageConsumedCaloriesHourly)) +

geom_histogram(bins = 10, fill = "blue", color = "black") +

ylab("Number of Participants") +

xlab("Average Consumed Calories") +

labs(title = "Average Calories per Hour") +

facet_wrap(~Hour)
```

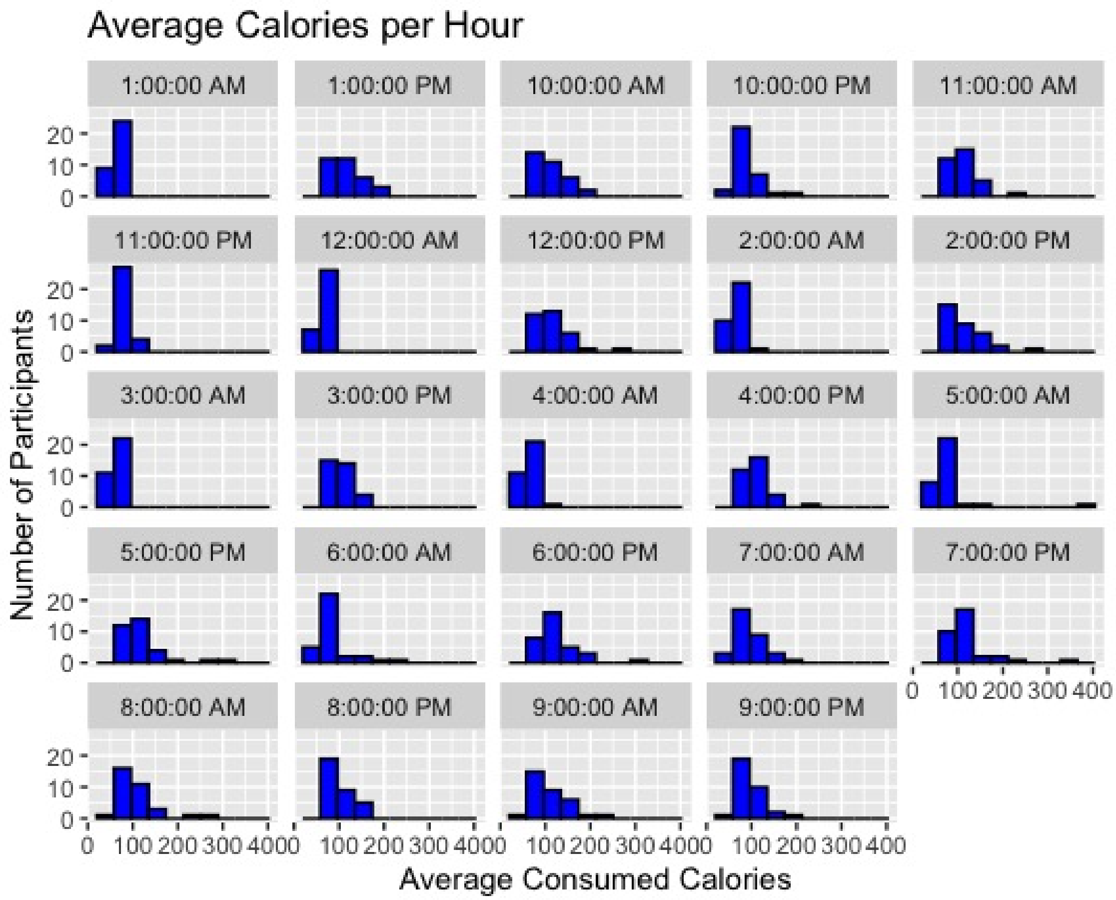

Figure 6: Average Calories Hourly

ggplot(AVG_Hourly_Steps, aes(x = AeravgeStepsHourly)) +

geom_histogram(bins = 10, fill = “red”, color = “black”) +

ylab(“Number of Participants”) +

xlab(“Average Taken Steps”) +

labs(title = “Average Taken Steps Each Hour”) +

facet_wrap(~Hour)

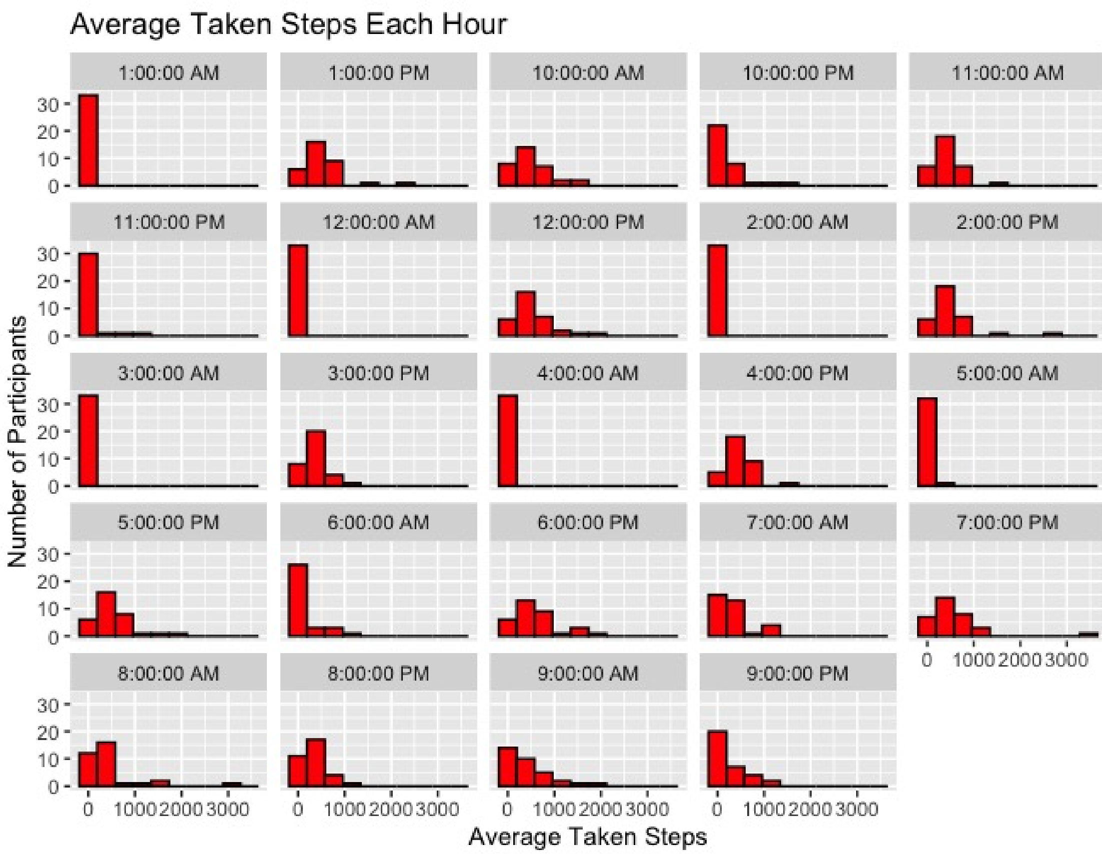

Figure 7: Average Steps Hourly

ggplot(AVG_Hourly_Intense, aes(x = AverageIntenseHourly)) +

geom_histogram(bins = 10, fill = “green”, color = “black”) +

ylab(“Number of Participants”) +

xlab(“Average Intense Workout”) +

labs(title = “Average of Intense Workout Each Hour”) +

facet_wrap(~Hour)

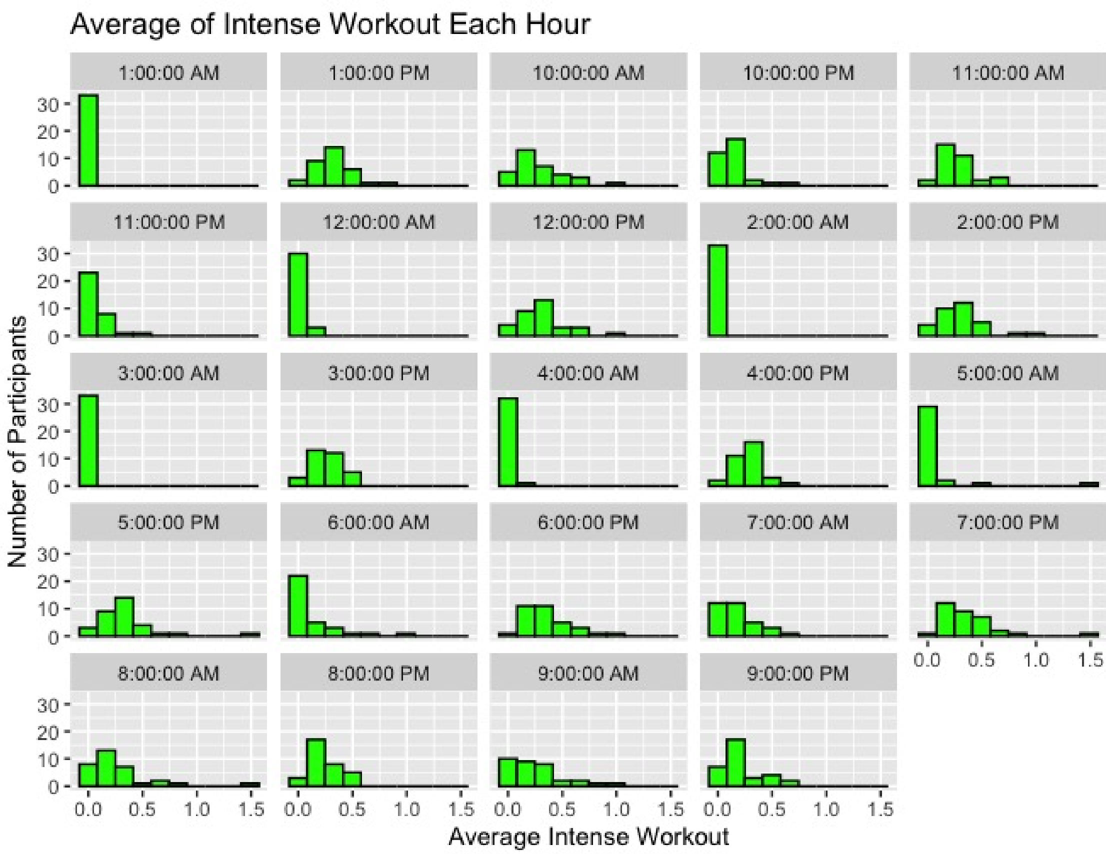

Figure 8: Average Intensity Hourly

Dates <- AVG_Sleep_Dates

AVG_Sleep_Dates$Date <- as.Date(AVG_Sleep_Dates$Date, “%m/%d/%Y”)

ggplot(AVG_Sleep_Dates, aes(x = Date, y = AverageHoursSleeping)) +

geom_point(alpha = .6) +

geom_line(color = “black”) +

ylab(“Average Hours of Sleep”) +

xlab(“Date”) +

labs(title = “Average Hours of Sleep per day for participants”)

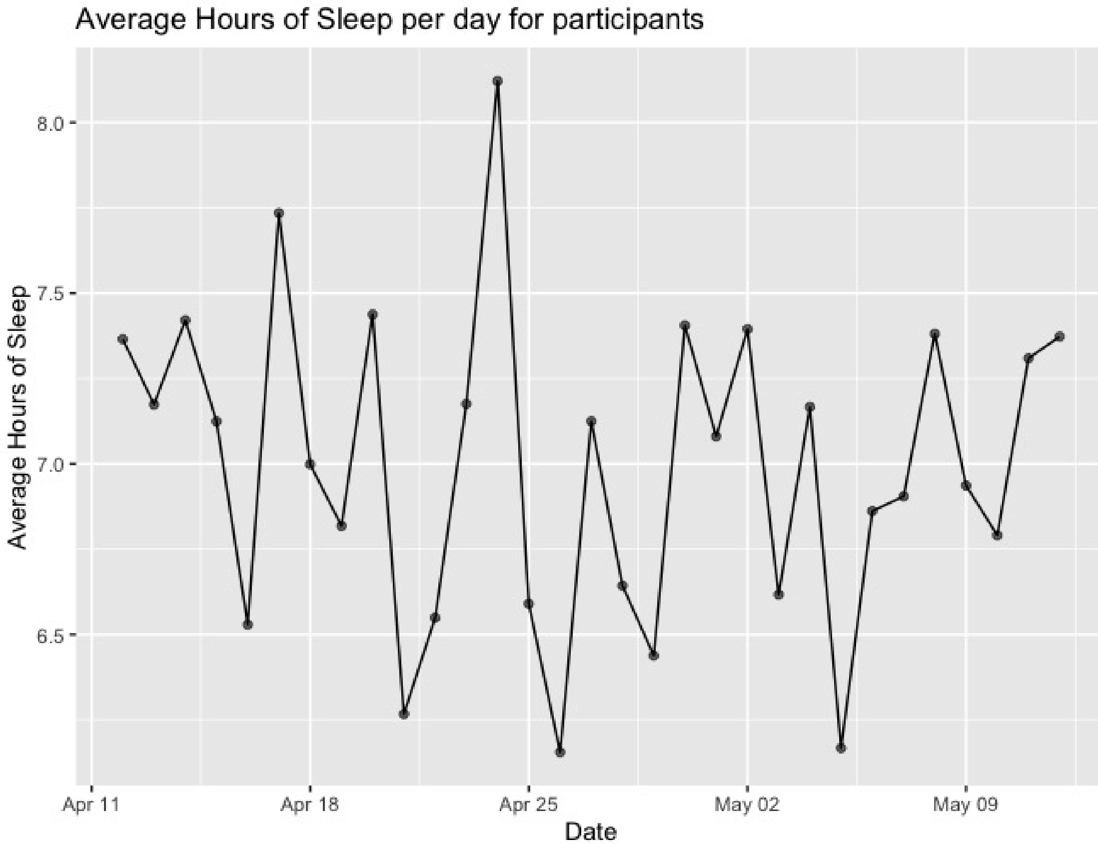

Figure 9: Average Sleeping Hours Per Day

Now there are available visualizations to show participants' habits of sleep, activities, consumed calories, and intense workout. The final graph will plot the participants' IDs to show how many hours each person sleep per day. Also, a new dataset will be created to demonstrate which days of the week were recorded for people's sleeping hours.

#Show number of hours participants sleep each day

AVG_Sleep_Days <- Dates %>%

mutate(Day =

case_when(Date == "4/17/2016" |Date == "4/24/2016" |Date == "5/1/2016"

|Date == "5/8/2016" ~"Sunday",

Date == "4/18/2016" |Date == "4/25/2016" |Date == "5/2/2016"

|Date == "5/9/2016" ~"Monday",

Date == "4/19/2016" |Date == "4/26/2016" |Date == "5/3/2016"

|Date == “4/12/2016” ~“Tuesday”,

Date == “4/20/2016” |Date == “4/27/2016” |Date == “5/4/2016”

|Date == “4/13/2016” ~“Wednesday”,

Date == “4/21/2016” |Date == “4/28/2016” |Date == “5/5/2016”

|Date == “4/14/2016” ~“Thursday”,

Date == “4/22/2016” |Date == “4/29/2016” |Date == “5/6/2016”

|Date == “4/15/2016” ~“Friday”,

Date == “4/23/2016” |Date == “4/30/2016” |Date == “5/7/2016”

|Date == “4/16/2016” ~“Saturday”,)) %>%

select(-Date)

head(AVG_Sleep_Days)

A tibble: 6 × 2

AverageHoursSleeping Day

$< dbl > < chr >$

1 7.37 Tuesday

2 7.17 Wednesday

3 7.42 Thursday

4 7.12 Friday

5 6.53 Saturday

6 7.73 Sunday

Activity_Sleep <- Participants_Activity %>%

inner_join(AVG_Hours_Asleep)

Joining, by = “Id”

#Plot every participant’s sleeping hours ggplot(AVG_Hours_Asleep, aes(x = AverageHoursSleeping, y = as.factor(Id))) +

geom_histogram(stat = “identity”, fill = “lightpink”, color = “black”) +

ylab(“Participants by ID”) +

xlab(“Average Sleeping Hours”) +

labs(title = “Participants’ Sleeping Hours”)

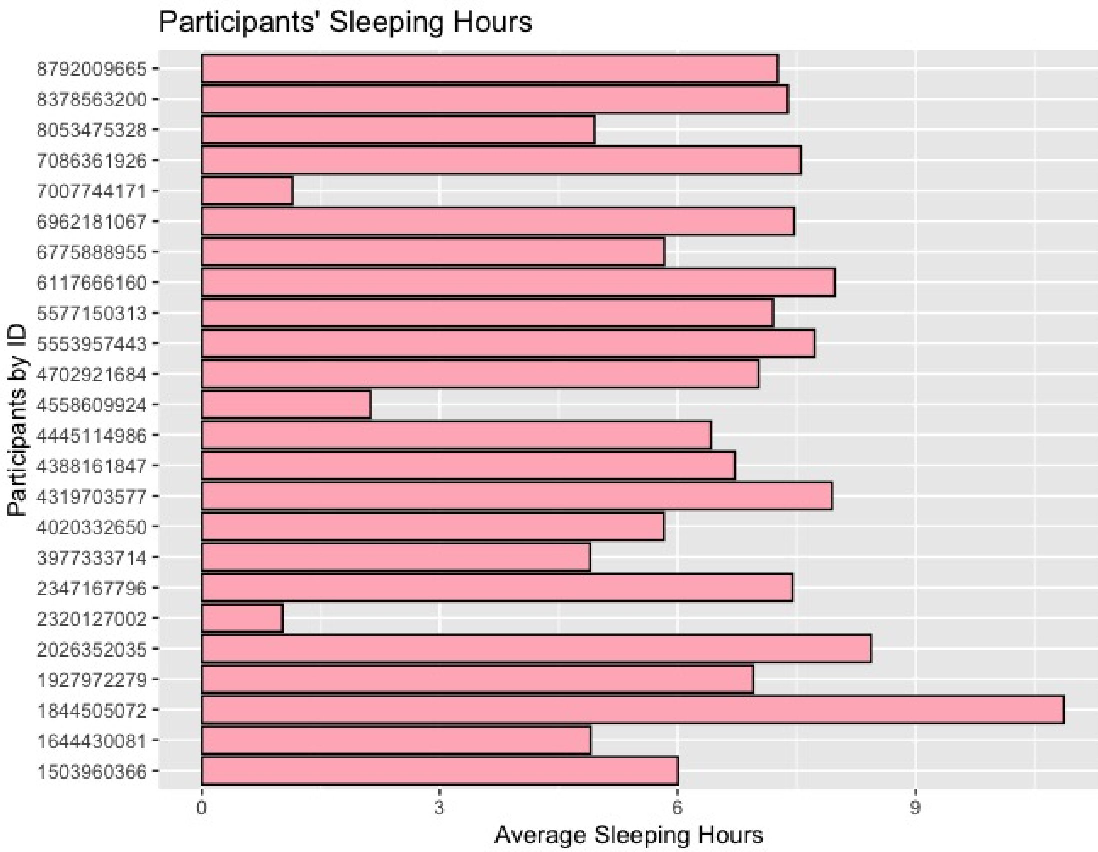

Figure 10: Participants’ Sleeping Hours

6 Act

6.1 Conclusion: Recommendations for Bellabeat

The collected database provides the company information about the daily habits of their consumers and their health information. Bellabeat company keeps growing due to its investment in Google Search and the work on social media pages active on Facebook, Instagram, and Twitter. The company also runs Youtube ads to support campaigns. The analysis of the Fitbit Tracker database allowed us to come up with some recommendations for Bellabeat company to strengthen its marketing strategy.

6.2 Recommendations to the marketing team

1. From the first two visualizations, more than half of the consumers tend to take a minimum of one day off, and about 16% take more than seven days off, which is a large number. It seems that people are not interested in using their Fitbit devices. To fight this problem, the company can enhance and modify their products to attract consumers with new features that support people better, such as distinguishing between when the people should resume their activity and when to take a day off and have rest.

2. The following graph also shows that there are only 12 very active participants and the rest vary between lightly and Fairly active. Most participants were inactive as they should be. As a recommendation, the company can let the product notify their consumers of their daily active habits and whether they should be more or less active during the day.

3. Ten consumers are following their calorie intake correctly at 2000 kcal daily. Others are under or above that average; most participants' daily intake is more than 2000 kcal. In addition, most of them tend to eat late at night and after midnight. Solving this problem relies on the product they are using. There are two options, first is to implement a reminder in the product to remind the users not to eat later than a specified hour and notify them when they complete their daily calorie intake. Secondly, provide food menus within people's devices to support users with healthy food options and choose their preferences.

4. Most participants are taking more than 5000 steps daily. For motivation, people's smart devices can give daily reports about how many steps did they walk and if they need to walk more or not.

5. Finally, most participants tend to sleep between 7-9 hours a day; however, some suffer from lack of sleep. To overcome this issue, participants can set a sleeping hours goal within their smart devices to modify their sleeping habits until they get a fixed schedule.

6.3 Limitations of The Study

1. The collected dataset was from between 03.12.2016 and 05.12.2016. That means the data is outdated, and a new dataset should be made available.

2. There are a lot of missing values within the data which limited the analysis phase with the available options. First, the "Weight" dataset had only 8 participants' weight records with several missing values that prevented us from analyzing the data. Secondly, the heart rate dataset had records for half of the participants only, which limited our options to consider the dataset for our analysis.

To sum up, the company can alter their devices by considering notifications for each daily habit the users have on their devices to be aware of their daily activities and use their devices regularly.