Bellabeat Report

Lim Wei Hern

11/22/2021

This project uses the FitBit Fitness Tracker Data from Kaggle

The Business Task

Bellabeat is a high-tech manufacturer of health-focused products for women. In this report, I will use smart device data to analyse how consumers are using their smart devices and provide recommendations for the marketing strategy for the company.

Bellabeat's list of products are: * Bellabeat App: The Bellabeat app provides users with health data related to their activity, sleep, stress, menstrual cycle, and mindfulness habits. This data can help users better understand their current habits and make healthy decisions. The Bellabeat app connects to their line of smart wellness products * Leaf: Bellabeat's classic wellness tracker can be worn as a bracelet, necklace, or clip. The Leaf tracker connects to the Bellabeat app to track activity, sleep, and stress. * Time: This wellness watch combines the timeless look of a classic timepiece with smart technology to track user activity, sleep, and stress. The Time watch connects to the Bellabeat app to provide you with insights into your daily wellness. * Spring: This is a water bottle that tracks daily water intake using smart technology to ensure that you are appropriately hydrated throughout the day. The Spring bottle connects to the Bellabeat app to track your hydration levels. * Bellabeat membership Bellabeat also offers a subscription-based membership program for users. Membership gives users 24/7 access to fully personalized guidance on nutrition, activity, sleep, health and beauty, and mindfulness-based on their lifestyle and goals

Table of Contents:

Deliverables:

- A clear summary of the business task
- A description of all data sources used
- Documentation of any cleaning or manipulation of data
- A summary of analysis
- Supporting visualizations and key findings
- Top high-level content recommendations based on the analysis

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1

## -- Conflicts ---------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(readr)
library(lubridate)

##
## Attaching package: 'lubridate'

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

library("here")

## here() starts at C:/Users/limwe/OneDrive/Desktop/Case Study 2

library("skimr")
library("janitor")

##
## Attaching package: 'janitor'

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

library(scales)

##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor

library(colorspace)
library(RColorBrewer)
#Loading Files
#daily_Activity_merged has all the necessary data
dailyActivity_merged <- read_csv("dailyActivity_merged.csv")
```

```
## Rows: 940 Columns: 15

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (1): ActivityDate
## dbl (14): Id, TotalSteps, TotalDistance, TrackerDistance,
LoggedActivitiesDi...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

heartrate_seconds_merged <- read_csv("heartrate_seconds_merged.csv")

## Rows: 2483658 Columns: 3

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr (1): Time
## dbl (2): Id, Value

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

hourlyCalories_merged <- read_csv("hourlyCalories_merged.csv")

## Rows: 22099 Columns: 3

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr (1): ActivityHour
## dbl (2): Id, Calories

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

hourlyIntensities_merged <- read_csv("hourlyIntensities_merged.csv")

## Rows: 22099 Columns: 4

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr (1): ActivityHour
## dbl (3): Id, TotalIntensity, AverageIntensity
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

hourlySteps_merged <- read_csv("hourlySteps_merged.csv")

## Rows: 22099 Columns: 3

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr (1): ActivityHour
## dbl (2): Id, StepTotal

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

minuteCaloriesNarrow_merged <- read_csv("minuteCaloriesNarrow_merged.csv")

## Rows: 1325580 Columns: 3

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr (1): ActivityMinute
## dbl (2): Id, Calories

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

minuteCaloriesWide_merged <- read_csv("minuteCaloriesWide_merged.csv")

## Rows: 21645 Columns: 62

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (1): ActivityHour
## dbl (61): Id, Calories00, Calories01, Calories02, Calories03, Calories04,
Ca...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

minuteIntensitiesNarrow_merged <-
read_csv("minuteIntensitiesNarrow_merged.csv")
```

```
## Rows: 1325580 Columns: 3

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr (1): ActivityMinute
## dbl (2): Id, Intensity

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

minuteIntensitiesWide_merged <- read_csv("minuteIntensitiesWide_merged.csv")

## Rows: 21645 Columns: 62

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (1): ActivityHour
## dbl (61): Id, Intensity00, Intensity01, Intensity02, Intensity03,
Intensity0...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

minuteMETsNarrow_merged <- read_csv("minuteMETsNarrow_merged.csv")

## Rows: 1325580 Columns: 3

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr (1): ActivityMinute
## dbl (2): Id, METs

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

minuteSleep_merged <- read_csv("minuteSleep_merged.csv")

## Rows: 188521 Columns: 4

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr (1): date
## dbl (3): Id, value, logId
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

minuteStepsNarrow_merged <- read_csv("minuteStepsNarrow_merged.csv")

## Rows: 1325580 Columns: 3

## -- Column specification ---------------------------------------------------
------
## Delimiter: ","
## chr (1): ActivityMinute
## dbl (2): Id, Steps

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

minuteStepsWide_merged <- read_csv("minuteStepsWide_merged.csv")

## Rows: 21645 Columns: 62

## -- Column specification ---------------------------------------------------
------
## Delimiter: ","
## chr  (1): ActivityHour
## dbl (61): Id, Steps00, Steps01, Steps02, Steps03, Steps04, Steps05,
Steps06,...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

sleep_day_data <- read_csv("sleepDay_merged.csv")

## Rows: 413 Columns: 5

## -- Column specification ---------------------------------------------------
------
## Delimiter: ","
## chr (1): SleepDay
## dbl (4): Id, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

weight_data <- read_csv("weightLogInfo_merged.csv")
```

```
## Rows: 67 Columns: 8

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr (1): Date
## dbl (6): Id, WeightKg, WeightPounds, Fat, BMI, LogId
## lgl (1): IsManualReport

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

Prepare Data

This Kaggle data set contains personal fitness tracker data from thirty fitbit users. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and heart rate that can be used to explore users' habits.

To analyze the data, the following ROCCC framework was used:

- **Reliable:** This data is reliable as there is unlikely to be sampling bias given that I am using the data wholesale from Bellabeat. However, given that the data was sampled from 30 users, there might be bias as the type of consumers who buy the trackers might not be the same as those who buy other products such as Spring and the Membership
- **Original:** This data is considered original as it is downloaded directly from Bellabeat users (first-party data).
- **Comprehensive:** The data is comprehensive given that it contains many details throughout the day for each user.
- **Current:** The data may be slightly outdated as it was taken 5 years ago in 2016. This is however a by-product of the Kaggle dataset used.
- **Cited:** The data is first-hand data.

Process Data

```
daily_data <- dailyActivity_merged
rm(dailyActivity_merged)
hourly_data <- full_join(hourlyCalories_merged,hourlyIntensities_merged)

## Joining, by = c("Id", "ActivityHour")

hourly_data <- full_join(hourly_data,hourlySteps_merged)

## Joining, by = c("Id", "ActivityHour")
```

```
rm(hourlyCalories_merged,hourlyIntensities_merged,hourlySteps_merged)
minutes_data_narrow <- full_join(minuteCaloriesNarrow_merged,
                                 minuteIntensitiesNarrow_merged)

## Joining, by = c("Id", "ActivityMinute")

minutes_data_narrow <- full_join(minutes_data_narrow,minuteMETsNarrow_merged)

## Joining, by = c("Id", "ActivityMinute")

minutes_data_narrow <-
full_join(minutes_data_narrow,minuteStepsNarrow_merged)

## Joining, by = c("Id", "ActivityMinute")

rm(minuteCaloriesNarrow_merged,minuteIntensitiesNarrow_merged,
   minuteStepsNarrow_merged,minuteMETsNarrow_merged)

#Note that there is only narrow METS data
minutes_data_wide <- full_join(minuteCaloriesWide_merged,
                               minuteIntensitiesWide_merged)

## Joining, by = c("Id", "ActivityHour")

minutes_data_wide <- full_join(minutes_data_wide,minuteStepsWide_merged)

## Joining, by = c("Id", "ActivityHour")

rm(minuteCaloriesWide_merged,minuteIntensitiesWide_merged,minuteStepsWide_mer
ged
   )

daily_data$ActivityDate <- as.Date(daily_data$ActivityDate, "%d/%m/%y")
heartrate_seconds_merged$Time <-
lubridate::mdy_hms(heartrate_seconds_merged$Time)
hourly_data$ActivityHour <- lubridate::mdy_hms(hourly_data$ActivityHour)
minutes_data_narrow$ActivityMinute <-
lubridate::mdy_hms(minutes_data_narrow$ActivityMinute)
minutes_data_wide$ActivityMinute <-
lubridate::mdy_hms(minutes_data_wide$ActivityHour)
minuteSleep_merged$date <- lubridate::mdy_hms(minuteSleep_merged$date)
weight_data$Date <- lubridate::mdy_hms(weight_data$Date)
sleep_day_data$SleepDay <- lubridate::mdy_hms(sleep_day_data$SleepDay)
```

I first started by merging all the data together (daily data, narrow minutes data, and wide minutes data). It is worth pointing out that dailyActivity_merged.csv contains all the relevant daily data from the get go. Further, there is no METs data in a wide format.

I then converted the necessary data into their appropriate formats. In the *daily_data* tibble, ActivityDate was converted to Date. In the *heartrate_seconds_merged* tibble, time was converted to Date-Time Format. In the *hourly_data* tibble, ActivityHour was converted to Date-Time format. In the *minutes_data_narrow* and *minutes_data_wide* tibble,

ActivityMinute and ActivityHour were respectively converted to Date-Time format as well. The date columns in *minutesSleep_merged* and *weight_data* as well as the SleepDay column in *sleep_day_data* were all converted from characters to Date-Time format.

I also noticed that there appeared to be a BMI recording of 47.5. This appeared to be an outlier as the second highest recording was 28. Since this was likely a result of a misfunction, I removed this from my dataset.

The Analysis

The first order of business is to check the number of users in each of the datasets. To do so, I queried for all distinct ID values.

```
n_distinct(daily_data$Id)

## [1] 33

n_distinct(heartrate_seconds_merged$Id)

## [1] 14

n_distinct(hourly_data$Id)

## [1] 33

n_distinct(minutes_data_narrow$Id)

## [1] 33

n_distinct(minutes_data_wide$Id)

## [1] 33

n_distinct(minuteSleep_merged$Id)

## [1] 24

n_distinct(weight_data$Id)

## [1] 8
```

Given that there were 3 studies, there were a varying number of distinct users. There were 33 sessions recorded for the Activity Data, 14 for Heart Rate Data, 24 for Sleep Data and 8 for Weight Data. As such, the smaller and differing sample sizes will be accounted for as a limitation when giving recommendations towards the end of the report.

```
mean(daily_data[["TotalSteps"]])

## [1] 7637.911

mean(daily_data[["TotalDistance"]])

## [1] 5.489702
```

```
mean(daily_data[["Calories"]])
```

```
## [1] 2303.61
```

```
mean(daily_data[["SedentaryActiveDistance"]])
```

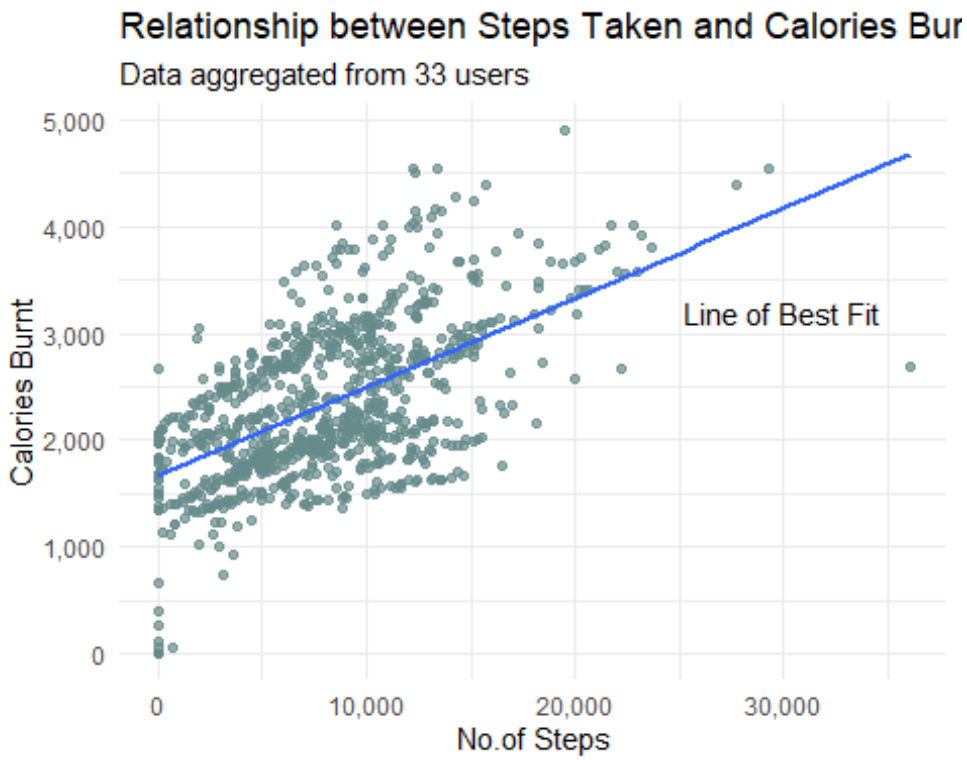
```
## [1] 0.001606383
```

From the daily_data tibble, We discover that the average steps taken by users is *7637* steps at an average of *5.4* miles each day. The CDC is recommending the average adult to clock 10,000 steps a day. The average amount of calories burned each day is *2303.61kcal* as recorded by the trackers. According to healthline, it is recommended to reduce calorie intake by 500kcal per day to achieve weight loss.

As evidenced from the graphs below, there is a clear positive relationship between calories burnt and steps taken as well as calories burnt and distance covered in a day.

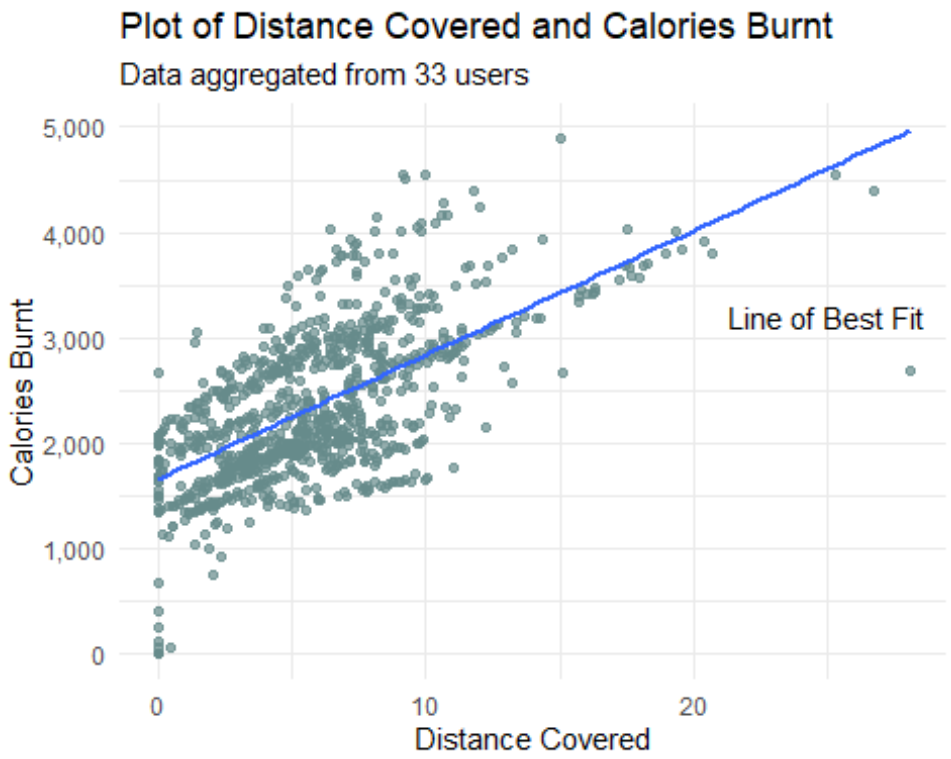
```
ggplot(data=daily_data, aes(x=TotalSteps,y=Calories))+
  geom_point(color="paleturquoise4",
             alpha = 0.7, position = position_jitter()) +
  geom_smooth(method='lm', se=FALSE)+
  scale_y_continuous(label=comma)+
  scale_x_continuous(label=comma)+
  labs(x="No.of Steps", y="Calories Burnt",
       title="Relationship between Steps Taken and Calories Burnt",
       subtitle="Data aggregated from 33 users")+
  theme_minimal()+
  annotate("text", x= 30000, y =3200, label="Line of Best Fit")
```

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




```
ggplot(data=daily_data, aes(x=TotalDistance,y=Calories))+
  geom_point(color="paleturquoise4",
             alpha = 0.7, position = position_jitter()) +
  geom_smooth(method='lm', se=FALSE)+
  scale_y_continuous(label=comma)+
  scale_x_continuous(label=comma)+
  labs(x="Distance Covered", y="Calories Burnt",
       title="Plot of Distance Covered and Calories Burnt",
       subtitle="Data aggregated from 33 users")+
  theme_minimal()+
  annotate("text", x= 25, y =3200, label="Line of Best Fit")
```

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



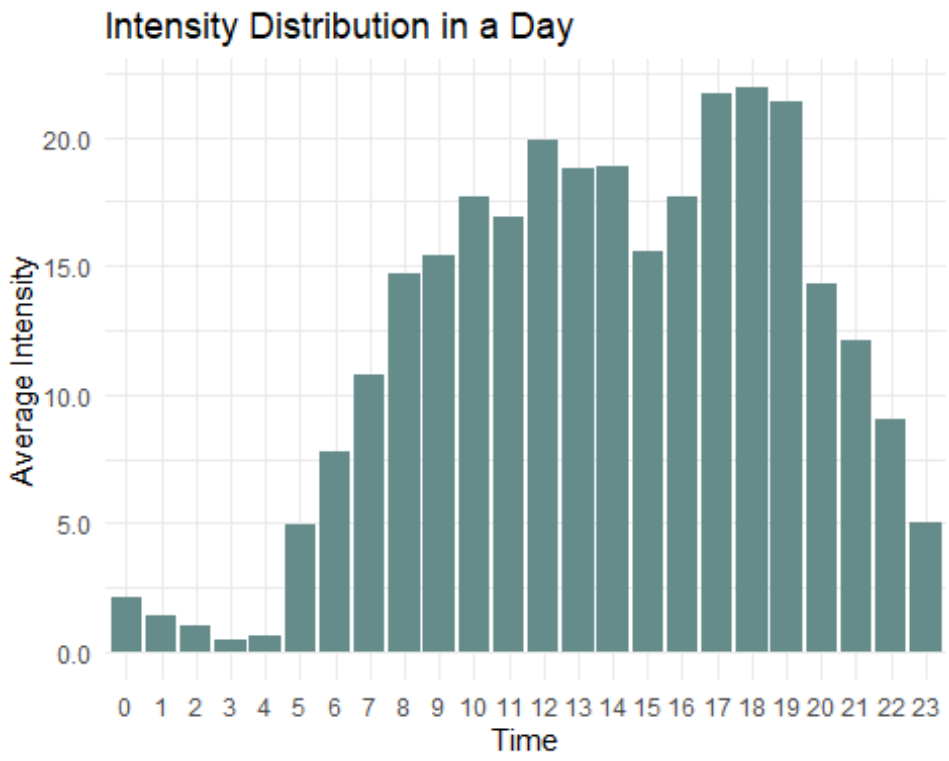
Next, by taking the sum of Fairly Active Minutes and Very Active Minutes, I obtained the total active minutes each day. On average, users spend 34 minutes exercising moderately each day which is above the recommended 150 per week.

```
daily_data <- mutate(daily_data,ExerciseMinutes=
FairlyActiveMinutes+VeryActiveMinutes)
mean(daily_data[["ExerciseMinutes"]])

## [1] 34.72979

mean(daily_data[["SedentaryMinutes"]])

## [1] 991.2106

mean(sleep_day_data[["TotalMinutesAsleep"]])

## [1] 419.4673
```

Further, given that the average Sedentary Active Distance is 0.0016 miles, we can assume that sedentary minutes are hence time spent sitting down. According to this Washington Post Article, adults should begin to stand, move and take breaks for at least two out of eight hours at work. From our data, the average user clocks 991.2 Sendentary minutes each day. Factoring in the average minutes asleep of 419, it is safe to assume that the average user spends 571.75 minutes or 9 hours and 30 minutes sitting.

It is also worth pointing out that the average of 419 minutes a sleep each night or approximately 7 hours a night falls right along the National Sleep Foundation guidelines of between 7 and 9 hours a night.

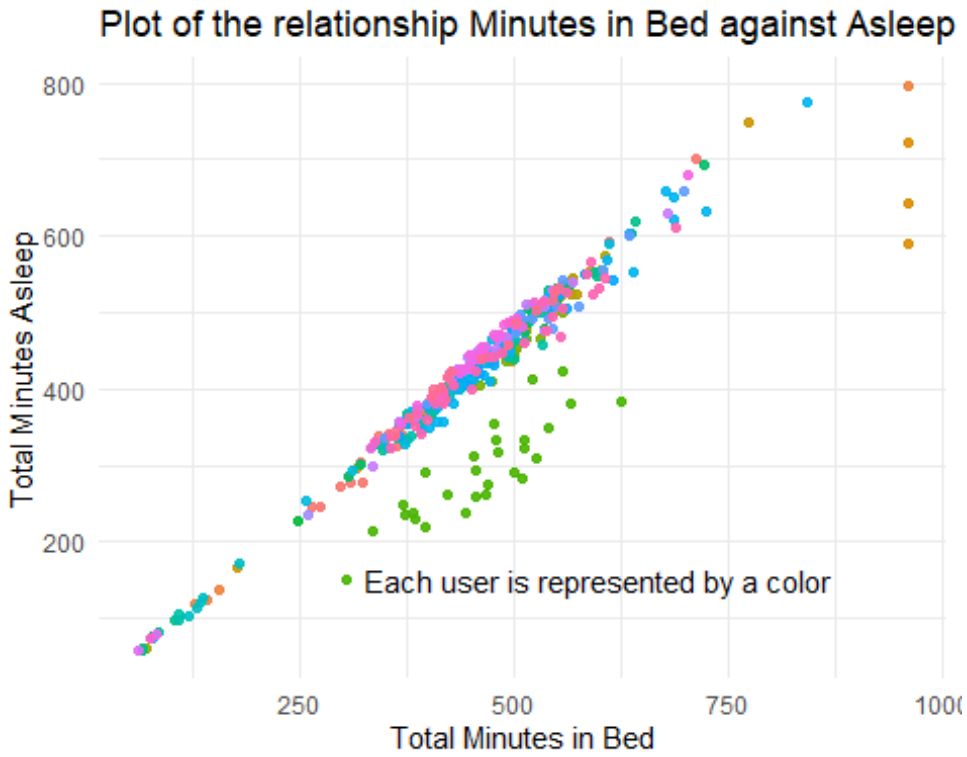
```
tmp <-hourly_data
tmp <-separate(tmp, ActivityHour, into =c("Date","Time"),sep=" ")
tmp$Date = as.Date(tmp$Date, "%dm/%d/%y")

tmp_labels <- c(0:23)
ggplot(tmp,aes(x=Time, y=TotalIntensity))+
  geom_bar(fill="paleturquoise4",stat="summary",
           fun="mean")+
  scale_x_discrete(labels=tmp_labels)+
  scale_y_continuous(labels=comma)+
  labs(X="Hours of the Day", y="Average Intensity",
      title="Intensity Distribution in a Day")+
  theme_minimal()
```



As we can see, the intensity peaks between 17:00 and 19:00 which might be due to most people wanting to exercise during the evening.

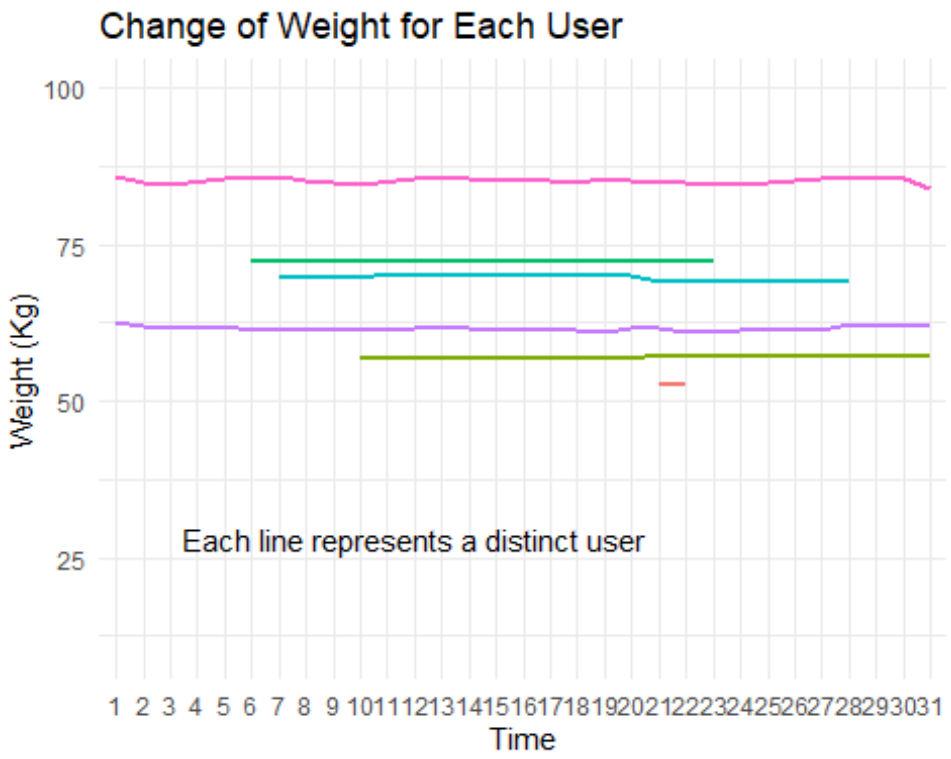
```
ggplot(sleep_day_data,aes(x=TotalTimeInBed,
y=TotalMinutesAsleep,color=as.factor(Id)))+
  scale_shape_manual(values=c(3, 16, 17, 5, 9, 20, 25, 30))+
  geom_point(alpha = 0.9,  position = position_jitter()) +
  theme_minimal()+
  theme(legend.position = "none")+
```

```
  labs(x="Total Minutes in Bed", y ="Total Minutes Asleep",
  title="Plot of the relationship Minutes in Bed against Asleep")+
  annotate("text", x=600, y=150, label="Each user is represented by a color")
```



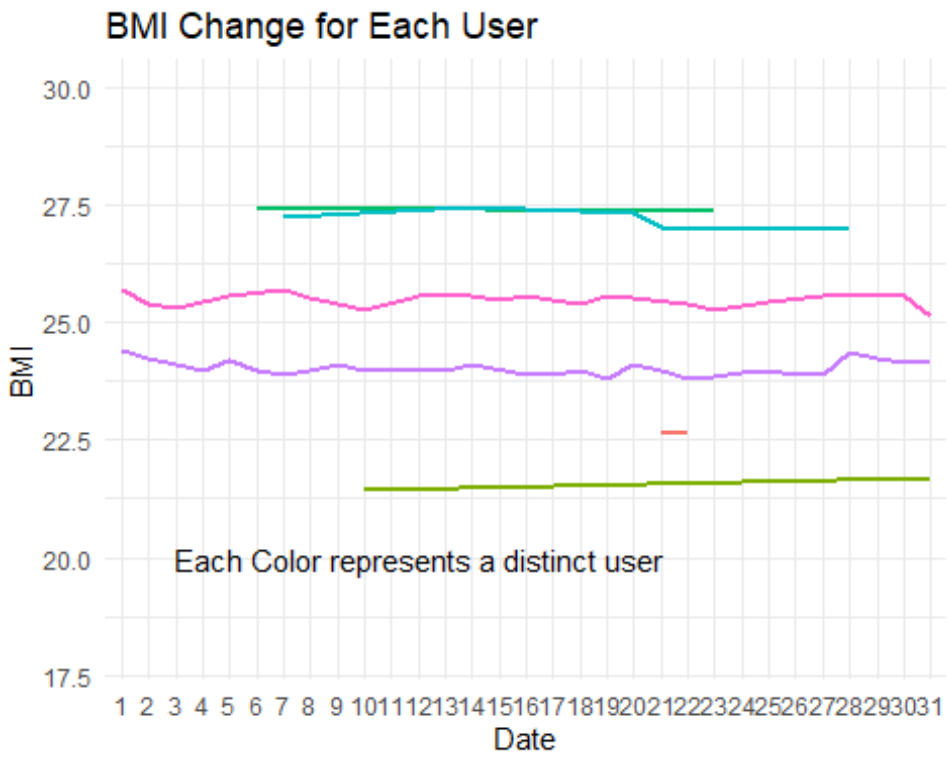
The graph above demonstrates the ease of sleeping for each user. The closer their point is towards the central diagonal, the faster they fall asleep. We can see that user "green" has the most difficulty sleeping while user "orange" sleeps the longest.

```
tmp <-weight_data
tmp <-separate(weight_data, Date, into =c("Date","Time"),sep=" ")
tmp_labels <-c(1:31)

ggplot(tmp)+
  geom_line(mapping =aes(x=Date,y=WeightKg,group=as.factor(Id),
                         color=as.factor(Id)),size=1)+
  scale_shape_manual(values=c(3, 16, 17, 5, 9, 20, 25, 30))+
  scale_x_discrete(labels=tmp_labels)+
  ylim(10,100)+
  labs(x="Time", y="Weight (Kg)",title="Change of Weight for Each User")+
  theme_minimal()+
  theme(legend.position = "none")+
  annotate("text",x=12,y=28,label="Each line represents a distinct user" )

## Warning: Removed 1 row(s) containing missing values (geom_path).
```




```
ggplot(tmp)+
  geom_line(mapping =aes(x=Date,y=BMI,group=as.factor(Id),
                         color=as.factor(Id)),size=1)+
  scale_shape_manual(values=c(3, 16, 17, 5, 9, 20, 25, 30))+
  scale_x_discrete(labels=tmp_labels)+
  ylim(18,30)+
  labs(x="Date", y="BMI",title="BMI Change for Each User")+
  theme_minimal()+
  theme(legend.position = "none")+
  annotate("text",x=12,y=20,label="Each Color represents a distinct user" )
```

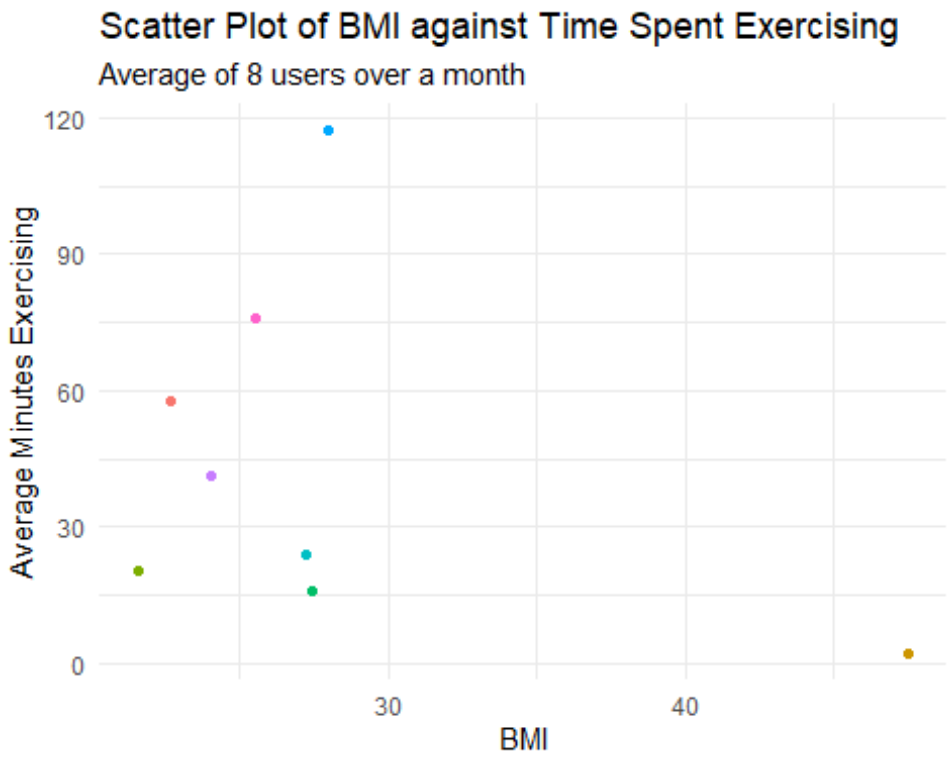
```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



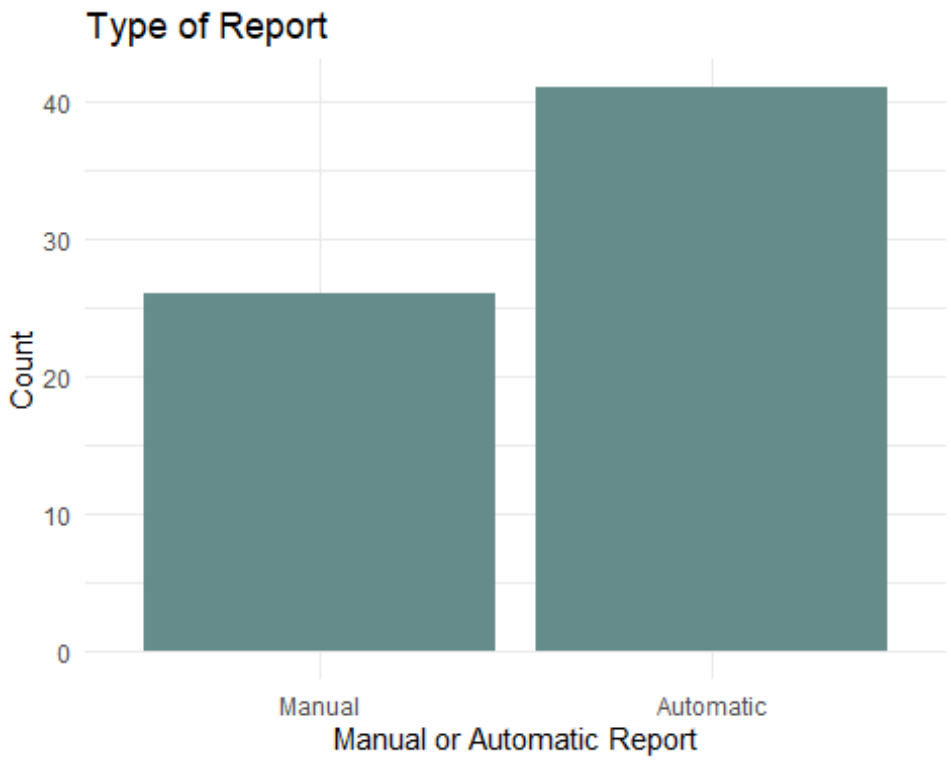
From the graphs above, we can see that there is practically no change in both BMI and Weight of each user. However, it is worth pointing out that only 3 users fall in the healthy BMI range of between 18.5 - 24.9. This could mean that the use of a tracker does not prompt a lifestyle change.

However, if we were to look at the relationship between time exercised and BMI, we notice a slight negative relationship. The higher the amount of time spent exercising, the lower the BMI. However, the trend is not fully clear and the sample size of 8 makes it less reliable as well.

```
tmp <- full_join(weight_data,daily_data)

## Joining, by = "Id"

tmp <- aggregate(tmp, by=list(tmp$Id),FUN =mean) %>%
select(Id,BMI,ExerciseMinutes) %>%drop_na()

ggplot(data=tmp)+
  geom_point(aes(x=BMI,y=ExerciseMinutes,color=as.factor(Id)))+
  theme_minimal()+
  labs(x="BMI", y="Average Minutes Exercising",
       title = "Scatter Plot of BMI against Time Spent Exercising",
       subtitle="Average of 8 users over a month")+
  theme(legend.position = "none")
```




```
tmp <- weight_data
tmp_labels <- c("Manual","Automatic")
ggplot(tmp)+
  geom_bar(mapping =aes(x=IsManualReport), fill ="paleturquoise4")+
  labs(x="Manual or Automatic Report", y="Count",title="Type of Report")+
  scale_x_discrete(labels=tmp_labels)+
  theme_minimal()
```



As evidenced by the graph above, there are almost twice as many automatic reports compared to manual reports.

Lastly, Bellabeat also offers heart rate monitoring products. We can see the statistical summary of the heart rate data below.

```
summary(heartrate_seconds_merged$Value)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   36.00   63.00   73.00   77.33   88.00  203.00
```

The average heart rate is 77.33bpm which falls in the normal range of 60-100bpm. The min of 36 is likely to be data collected when the user was asleep. However, a maximum recorded value of 203bpm is high and thus should be looked at given that the average bpm during exercise is between 100 and 160bpm.

Findings Summary

- The average user is taking fewer steps and sitting more than recommended
- There is a clear positive relationship between steps taken a day and calories burnt
- There is a clear positive relationship between distance covered and calories burnt
- Users tend to exercise in the evenings
- However, with there being no BMI or weight changes, it is unlikely that the tracker prompts a lifestyle change

Recommendations

To encourage a healthier lifestyle, Bellabeat's products can give calories and steps notifications relative to the recommended numbers. This can cover factors such as steps taken in a day, distance covered and time spent sitting.

Secondly, reminders can be sent out between 17:00 and 19:00 in the evenings. Given that most users tend to exercise during this period, such reminders are more likely to be heeded and thus users can build a habit of exercising during this time.

In today's world, proper sleep time is becoming increasingly scarce. Hence, Bellabeat can focus on marketing its sleep tracking features to attract users who are seeking to better understand their sleep schedules.

Lastly, most people are averse to using trackers where they have to key in their data themselves. Given that the majority of reports were automatically entered in this data set, this convenience can be leveraged in Bellabeat's marketing efforts.

Change Log

- Merged all daily data to form *daily_data* tibble of dimensions 940 x 15
- Merged narrow minutes data to form *minute_data_narrow* tibble of dimensions 1,325,580 x 6
- Merged wide minutes data to form *minute_data_wide* tibble of dimensions 21,645 x 182
- In the *daily_data* tibble, ActivityDate was converted to Date.
- In the *heartrate_seconds_merged* tibble, time was converted to Date-Time Format.
- In the *hourly_data* tibble, ActivityHour was converted to Date-Time format.
- In the *minutes_data_narrow* and *minutes_data_wide* tibble, ActivityMinute and ActivityHour were respectively converted to Date-Time format as well.
- The date columns in *minutesSleep_merged* and *weight_data* as well as the SleepDay column in *sleep_day_data* were all converted from characters to Date-Time format.
- Created ExereciseMinutes column in *daily_data* which now has dimensions of 940 x 16