BellaBeat Case Study Notes

Olajuwon Aina

2022-09-12

1. About the Company

Bellabeat is a high-tech manufacturer of women's health products. Bellabeat is a successful little business with the potential to grow into a major player in the global smart device industry. Urka Sren, cofounder and Chief Creative Officer of Bellabeat, believes that examining smart device fitness data could help the company discover new development prospects.

bellabeat

2. The Ask Phase

Where I get to ask the right questions to understand the business questions and also identify key stakeholders on the project.

Business Questions

- What are some trends in smart device usage?
- How could these trends apply to Bellabeat customers?
- How could these trends help influence Bellabeat marketing strategy?

The Business Task

How consumers use non-Bellabeat smart devices to gain insights

Stakeholders Involved

- Urka Sren - The cofounder and Chief Creative Officer of Bellabeat.
- Sando Mur - Bellabeat cofounder and key member of Bellabeat executive team
- The Marketing Analytics team at Bellabeat

3. The Prepare Phase

Here's where I get to gather the dataset to use, identify the source, the security, credibility and integrity.

Dataset used

The fitbit fitness tracker public data will be used for this analysis. Here

Data Accessibility and Data Privacy

By verifying the metadata of our dataset, we can confirm that it is open-source. The owner has dedicated the work to the public domain by waiving all of his or her rights to the work worldwide under copyright law, including all related and neighboring rights, to the extent permitted by law. You may copy, modify, distribute, and perform the work without asking permission.

Key Information About Our Dataset

These datasets were created by respondents to a distributed survey via Amazon Mechanical Turk between December 3rd and December 5th, 2016. Thirty (30) Fitbit users agreed to submit personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. The variation in output represents the use of various Fitbit trackers and individual tracking behaviors/preferences.

Credibility and Integrity of Data

This Kaggle data set contains thirty fitbit users' personal fitness trackers. Thirty Fitbit users agreed to submit personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. It contains data on daily activity, steps, and heart rate that can be used to investigate users' habits.

4. The Process Phase

In this phase we will carryout some data cleaning and formatting tasks to ensure the data variables are thorough and ready for visualization.

Setting Up My Environment

Setting up my R environment by loading the 'tidyverse' and other needed packages

```
library(tidyverse)  # Data import and wrangling
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6     ✔ purrr   0.3.4
## ✔ tibble  3.1.8     ✔ dplyr   1.0.9
## ✔ tidyr   1.2.0     ✔ stringr 1.4.1
## ✔ readr   2.1.2     ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)    # For data Visualization
library(dplyr)
library(tidyr)
library(scales)   # For transforming numbers in percentage
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

Get To Know Our Working Directory

```
getwd()   # Displays the working directory
```

```
## [1] "C:/Users/Ola/Documents"
```

Importing The Datasets

There are 18 csv files in the dataset. Each of them displays data related to the device's various functions: calories, activity level, daily steps, and so on.

To simplify the analysis, we will concentrate on daily data in this study.

Daily Activity

```
daily_activity <- read_csv("Fitabase Data 4.12.16-5.12.16/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.
```

```
View(daily_activity)
```

Daily Calories

```
daily_calories <- read_csv("Fitabase Data 4.12.16-5.12.16/dailyCalories_merged.csv")
```

```
## Rows: 940 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityDay
## 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.
```

Daily Intensities

```
daily_intensities <- read_csv("Fitabase Data 4.12.16-5.12.16/dailyIntensities_merged.csv")
```

```
## Rows: 940 Columns: 10
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityDay
## dbl (9): Id, SedentaryMinutes, LightlyActiveMinutes, FairlyActiveMinutes, Ve...
##
## 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.
```

Daily Steps

```
daily_steps <- read_csv("Fitabase Data 4.12.16-5.12.16/dailySteps_merged.csv")
```

```
## Rows: 940 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityDay
## 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.
```

Daily Sleep

```
daily_sleep <- read_csv("Fitabase Data 4.12.16-5.12.16/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

```
weight_info <- read_csv("Fitabase Data 4.12.16-5.12.16/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.
```

Preview Datasets

Let have a look at the various datasets and have a clear understanding of how they look, similarities and cohesion between the various datasets.

Daily Activity

```
View(daily_activity)
```

Daily Calories

```
View(daily_calories)
```

Daily Intensities

```
View(daily_intensities)
```

Daily Steps

```
View(daily_steps)
```

Daily Sleep

```
View(daily_sleep)
```

Weight

```
View(weight_info)
```

Cleaning and Formatting Our Dataset

After examining the various data sets, it is possible to conclude that table 1 (Daily activity) already contains information from table 2 (Daily calories), table 3 (Daily steps), and table 4 (Daily intensities). Another observation is that each dataset has the same number of observations. As a result, those dataframes will be removed.

```
rm(daily_calories, daily_intensities, daily_steps) #(removing tables)
```

Transforming the data to be homogeneous

Before merging the datasets, let's clean the date columns to make them homogeneous and transform them to right data type.

```
# Cleaning the variables
daily_activity <- daily_activity %>%
  rename(Date = ActivityDate) %>%
  mutate(Date = as.Date(Date, format = "%m/%d/%y"))

daily_sleep <- daily_sleep %>%
  rename(Date = SleepDay) %>%
  mutate(Date = as.Date(Date, format = "%m/%d/%y"))

weight_info <- weight_info %>%
  select(-LogId) %>%
  mutate(Date = as.Date(Date, format = "%m/%d/%y")) %>%
  mutate(IsManualReport = as.factor(IsManualReport))
```

Merging the Datasets

```
final_data <- merge(merge(daily_activity, daily_sleep, by=c('Id','Date'), all = TRUE), weight_info, by = c('Id',
'Date'), all = TRUE)
```

Viewing the Merged dataframe (final_data)

```
View(final_data)
```

Removing extra/irrelevant variables

```
final_data <- final_data %>%
  select(-c(TrackerDistance, LoggedActivitiesDistance, TotalSleepRecords, WeightPounds, Fat, BMI, IsManualRepor
t))
```

Reviewing the Merged dataframe (final_data) again after removing unwanted variables

```
View(final_data)
```

Checking the variables & data types

```
str(final_data)
```

```
## 'data.frame':    943 obs. of  16 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ Date                    : Date, format: "2020-04-12" "2020-04-13" ...
##  $ TotalSteps              : num  13162 10735 10460 9762 12669 ...
##  $ TotalDistance           : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ 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       : num  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : num  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : num  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : num  728 776 1218 726 773 ...
##  $ Calories                : num  1985 1797 1776 1745 1863 ...
##  $ TotalMinutesAsleep      : num  327 384 NA 412 340 700 NA 304 360 325 ...
##  $ TotalTimeInBed          : num  346 407 NA 442 367 712 NA 320 377 364 ...
##  $ WeightKg                : num  NA NA NA NA NA NA NA NA NA NA ...
```

We can see that majority of the variables are numerical.

```
summary(final_data)
```

```
##        Id                 Date              TotalSteps    TotalDistance
##  Min.   :1.504e+09   Min.   :2020-04-12   Min.   :    0   Min.   : 0.000
##  1st Qu.:2.320e+09   1st Qu.:2020-04-19   1st Qu.: 3795   1st Qu.: 2.620
##  Median :4.445e+09   Median :2020-04-26   Median : 7439   Median : 5.260
##  Mean   :4.858e+09   Mean   :2020-04-26   Mean   : 7652   Mean   : 5.503
##  3rd Qu.:6.962e+09   3rd Qu.:2020-05-04   3rd Qu.:10734   3rd Qu.: 7.720
##  Max.   :8.878e+09   Max.   :2020-05-12   Max.   :36019   Max.   :28.030
##
##  VeryActiveDistance ModeratelyActiveDistance LightActiveDistance
##  Min.   : 0.000     Min.   :0.0000           Min.   : 0.000
##  1st Qu.: 0.000     1st Qu.:0.0000           1st Qu.: 1.950
##  Median : 0.220     Median :0.2400           Median : 3.380
##  Mean   : 1.504     Mean   :0.5709           Mean   : 3.349
##  3rd Qu.: 2.065     3rd Qu.:0.8050           3rd Qu.: 4.790
##  Max.   :21.920     Max.   :6.4800           Max.   :10.710
##
##  SedentaryActiveDistance VeryActiveMinutes FairlyActiveMinutes
##  Min.   :0.000000        Min.   :  0.00    Min.   :  0.00
##  1st Qu.:0.000000        1st Qu.:  0.00    1st Qu.:  0.00
##  Median :0.000000        Median :  4.00    Median :  7.00
##  Mean   :0.001601        Mean   : 21.24    Mean   : 13.63
##  3rd Qu.:0.000000        3rd Qu.: 32.00    3rd Qu.: 19.00
##  Max.   :0.110000        Max.   :210.00    Max.   :143.00
##
##  LightlyActiveMinutes SedentaryMinutes    Calories    TotalMinutesAsleep
##  Min.   :  0          Min.   :   0.0   Min.   :   0   Min.   : 58.0
##  1st Qu.:127          1st Qu.: 729.0   1st Qu.:1830   1st Qu.:361.0
##  Median :199          Median :1057.0   Median :2140   Median :433.0
##  Mean   :193          Mean   : 990.4   Mean   :2308   Mean   :419.5
##  3rd Qu.:264          3rd Qu.:1229.0   3rd Qu.:2796   3rd Qu.:490.0
##  Max.   :518          Max.   :1440.0   Max.   :4900   Max.   :796.0
##                                                       NA's   :530
##  TotalTimeInBed     WeightKg
##  Min.   : 61.0   Min.   : 52.60
##  1st Qu.:403.0   1st Qu.: 61.40
##  Median :463.0   Median : 62.50
##  Mean   :458.6   Mean   : 72.04
##  3rd Qu.:526.0   3rd Qu.: 85.05
##  Max.   :961.0   Max.   :133.50
##  NA's   :530     NA's   :876
```

5. The Analyze and Share Phase

In this phase we will be plotting various graphs to analyze our dataset for possible findings.

Users Daily Activity

Now with data merged, we can check for Users daily activities in a simple box plot

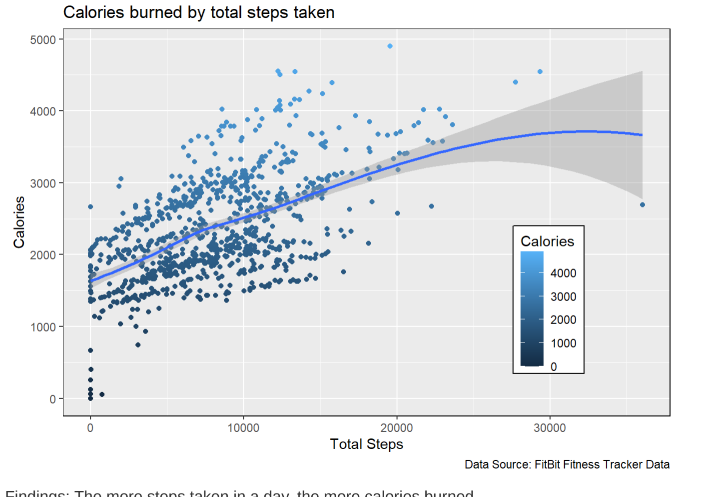
```
final_data %>%
  mutate(weekdays = weekdays(Date)) %>%
  select(weekdays, TotalSteps) %>%
  mutate(weekdays = factor(weekdays, levels = c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturda
y', 'Sunday'))) %>%
  drop_na() %>%
  ggplot(aes(weekdays, TotalSteps, fill = weekdays)) +
  geom_boxplot() +
  scale_fill_brewer(palette="Set2") +
  theme(legend.position="none") +
  labs(title = "Users' activity by day",x = "Day of the week",y = "Steps",
    caption = 'Data Source: FitBit Fitness Tracker Data')
```

Next, Check for Calories burned by Steps Taken

Check for calories calories burned by steps (i.e Calories vs Total Steps)

```
final_data %>%
  group_by(TotalSteps, Calories) %>%
  ggplot(aes(x = TotalSteps, y = Calories, color = Calories)) +
  geom_point() +
  geom_smooth() +
  theme(legend.position = c(.8, .3),
        legend.spacing.y = unit(1, "mm"),
        panel.border = element_rect(colour = "black", fill=NA),
        legend.background = element_blank(),
        legend.box.background = element_rect(colour = "black")) +
  labs(title = 'Calories burned by total steps taken',y = 'Calories',
       x = 'Total Steps',caption = 'Data Source: FitBit Fitness Tracker Data')
```

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



Findings: The more steps taken in a day, the more calories burned

These two variables have a clear positive correlation: the more steps taken in a day, the more calories burned. To verify this assumption, we can use the Pearson Correlation Coefficient to examine the correlation between these two variables.

Simply put, the Pearson Correlation Coefficient is a measure of two variables' linear correlation. Click here for more information.

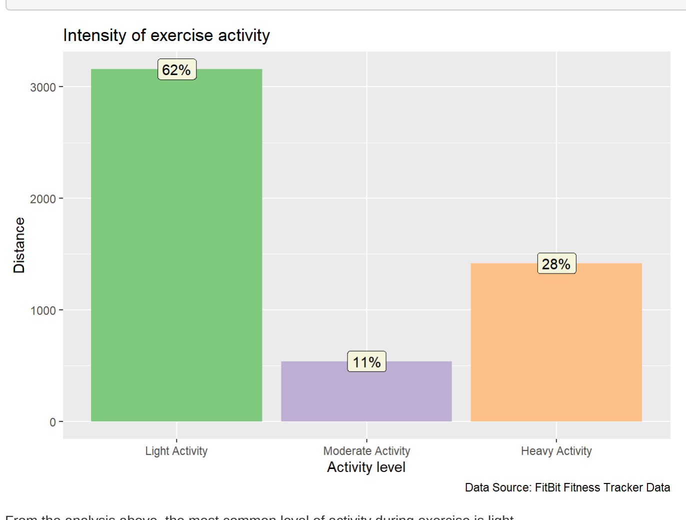
```
cor.test(final_data$TotalSteps, final_data$Calories, method = 'pearson', conf.level = 0.95)
```

```
##
##  Pearson's product-moment correlation
##
## data:  final_data$TotalSteps and final_data$Calories
## t = 22.588, df = 941, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5499261 0.6328341
## sample estimates:
##       cor
## 0.5929493
```

With a confidence level of 95%, the correlation between the variables is almost 0.6. This means that there is a strong relationship between the variables.

Next, Check for Intensity of Excercise Activity

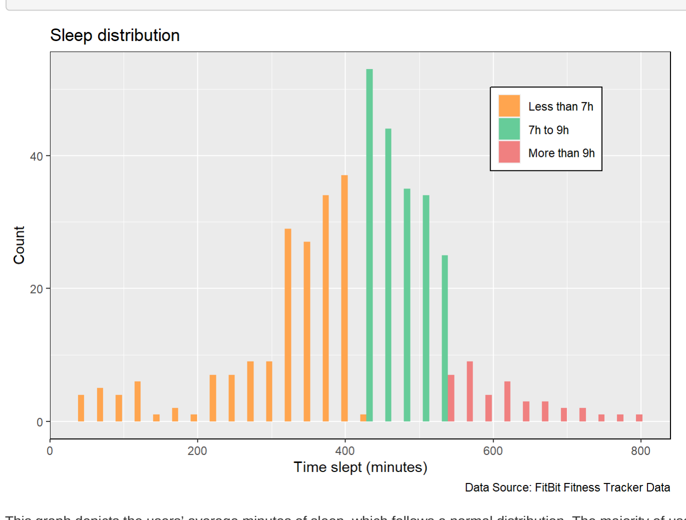
```
final_data %>%
  select(VeryActiveDistance,
         ModeratelyActiveDistance,
         LightActiveDistance) %>%
  summarise(across(everything(), list(sum))) %>%
  gather(activities, value) %>%
  mutate(ratio = value / sum(value),
         label = percent(ratio %>% round(4))) %>%
  mutate(activities = factor(activities,labels = c('Light Activity','Moderate Activity', 'Heavy Activity'))) %>%
  ggplot(aes(x = (activities),y = value,label = label,fill = activities)) +
  geom_bar(stat='identity') +
  geom_label(aes(label = label),fill = "beige", colour = "black",vjust = 0.5) +
  scale_fill_brewer(palette="Accent") +
  theme(legend.position="none") +
  labs(title = "Intensity of exercise activity",x = "Activity level",
    y = "Distance", caption = 'Data Source: FitBit Fitness Tracker Data')
```



From the analysis above, the most common level of activity during exercise is light.

Next, Sleep Distribution

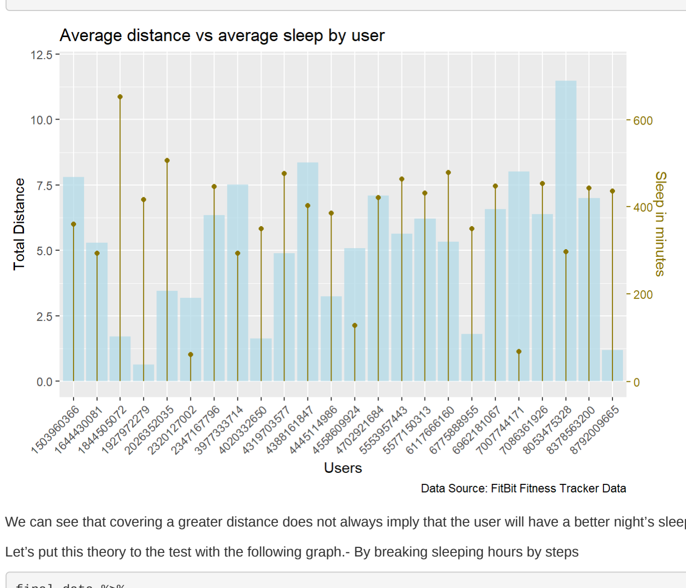
```
final_data %>%
  select(TotalMinutesAsleep) %>%
  drop_na() %>%
  mutate(sleep_quality = ifelse(TotalMinutesAsleep <= 420, 'Less than 7h',
                                ifelse(TotalMinutesAsleep <= 540, '7h to 9h',
                                       'More than 9h'))) %>%
  mutate(sleep_quality = factor(sleep_quality,
                                levels = c('Less than 7h','7h to 9h',
                                           'More than 9h'))) %>%
  ggplot(aes(x = TotalMinutesAsleep, fill = sleep_quality)) +
  geom_histogram(position = 'dodge', bins = 30) +
  scale_fill_manual(values=c("tan1", "#66CC99", "lightcoral")) +
  theme(legend.position = c(.80, .80),legend.title = element_blank(),legend.spacing.y = unit(0, "mm"),
        panel.border = element_rect(colour = "black", fill=NA),
        legend.background = element_blank(),legend.box.background = element_rect(colour = "black")) +
    labs(title = "Sleep distribution",x = "Time slept (minutes)",y = "Count",
    caption = 'Data Source: FitBit Fitness Tracker Data')
```



This graph depicts the users' average minutes of sleep, which follows a normal distribution. The majority of users sleep for 320 to 530 minutes.

Sleep Vs Distance Covered

```
final_data %>%
    select(Id, TotalDistance, TotalMinutesAsleep) %>%
    group_by(Id) %>%
    summarise_all(list(~mean(., na.rm=TRUE))) %>%
    drop_na() %>%
    mutate(Id = factor(Id)) %>%
    ggplot() +
    geom_bar(aes(x = Id, y = TotalDistance), stat = "identity", fill = 'lightblue', alpha = 0.7) +
    geom_point(aes(x = Id, y = TotalMinutesAsleep/60), color = 'gold4') +
    geom_segment(aes(x = Id, xend = Id, y = 0, yend = TotalMinutesAsleep/60), color = 'gold4' ,group = 1) +
    scale_y_continuous(limits=c(0, 12), name = "Total Distance",
                       sec.axis = sec_axis(~.*60, name = "Sleep in minutes")) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    theme(axis.title.y.right = element_text(color = "gold4"),axis.ticks.y.right = element_line(color = "gold4"),
          axis.text.y.right = element_text(color = "gold4")) +
    labs(
      title = "Average distance vs average sleep by user",x = "Users",
      caption = 'Data Source: FitBit Fitness Tracker Data')
```

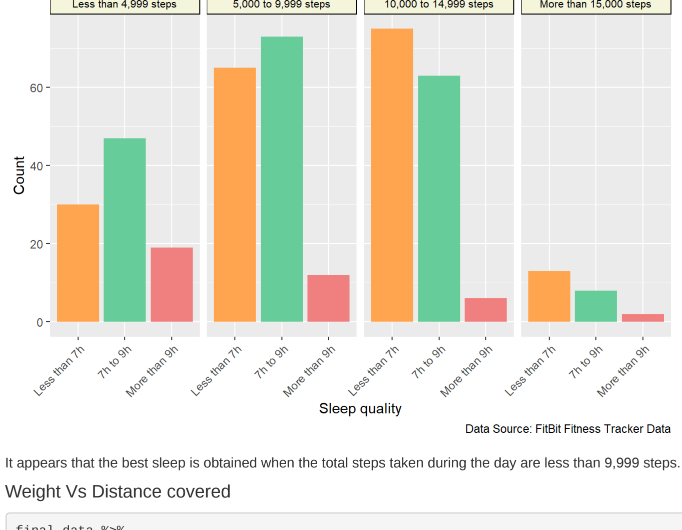


We can see that covering a greater distance does not always imply that the user will have a better night's sleep (on average).

Let's put this theory to the test with the following graph.- By breaking sleeping hours by steps

```
final_data %>%
    select(TotalMinutesAsleep, TotalSteps) %>%
    mutate(sleep_quality = ifelse(TotalMinutesAsleep <= 420, 'Less than 7h',
                                  ifelse(TotalMinutesAsleep <= 540, '7h to 9h',
                                         'More than 9h'))) %>%
    mutate(active_level = ifelse(TotalSteps >= 15000,'More than 15,000 steps',
                                 ifelse(TotalSteps >= 10000,'10,000 to 14,999 steps',
                                        ifelse(TotalSteps >= 5000, '5,000 to 9,999 steps',
                                               'Less than 4,999 steps')))) %>%
    select(-c(TotalMinutesAsleep, TotalSteps)) %>%
    drop_na() %>%
    group_by(sleep_quality, active_level) %>%
    summarise(counts = n()) %>%
    mutate(active_level = factor(active_level,
                                 levels = c('Less than 4,999 steps',
                                            '5,000 to 9,999 steps',
                                            '10,000 to 14,999 steps',
                                            'More than 15,000 steps'))) %>%
    mutate(sleep_quality = factor(sleep_quality,
                                  levels = c('Less than 7h','7h to 9h',
                                             'More than 9h'))) %>%
    ggplot(aes(x = sleep_quality,
               y = counts,
               fill = sleep_quality)) +
    geom_bar(stat = "identity") +
    scale_fill_manual(values=c("tan1", "#66CC99", "lightcoral")) +
    facet_wrap(~active_level, nrow = 1) +
    theme(legend.position = "none") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    theme(strip.text = element_text(colour = 'black', size = 8)) +
    theme(strip.background = element_rect(fill = "beige", color = 'black'))+
    labs(
      title = "Sleep quality by steps",
      x = "Sleep quality",
      y = "Count",
      caption = 'Data Source: FitBit Fitness Tracker Data')
```

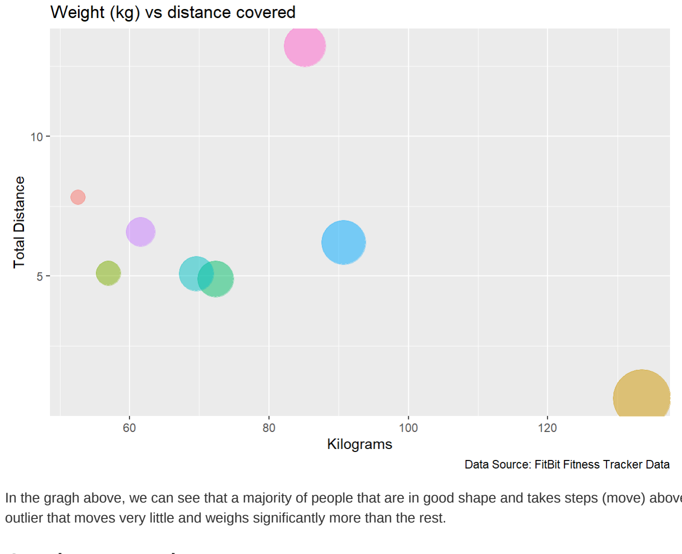
```
## `summarise()` has grouped output by 'sleep_quality'. You can override using the
## `.groups` argument.
```



It appears that the best sleep is obtained when the total steps taken during the day are less than 9,999 steps.

Weight Vs Distance covered

```
final_data %>%
    select(Id, WeightKg, TotalDistance) %>%
    group_by(Id) %>%
    summarise_all(list(~mean(., na.rm=TRUE))) %>%
    drop_na() %>%
    mutate(Id = factor(Id)) %>%
    ggplot(aes(WeightKg, TotalDistance, fill = Id)) +
    geom_point(aes(color = Id, size = WeightKg), alpha = 0.5) +
    scale_size(range = c(5, 20)) +
    theme(legend.position = "none") +
    labs(
      title = "Weight (kg) vs distance covered",
      x = "Kilograms",
      y = "Total Distance",
      caption = 'Data Source: FitBit Fitness Tracker Data')
```



In the gragh above, we can see that a majority of people that are in good shape and takes steps (move) above 5 miles. However, there is one outlier that moves very little and weighs significantly more than the rest.

6. The Act Phase

Finally, in this phase I get to share with the stakeholders my suggestions and conclusions based on the finds in our analysis.

Findings & Conclusions:

- Steps taken on a daily basis burn calories. Bellabeat could recommend a minimum number of steps for users to take (per day) based on their objectives to encourage them to achieve their goals.
- Bellabeat could send a notification (in the form of a pop up or calendar update) at a specific time for the user to remain consistent throughout the week in order to create a daily habit of exercising for its users.
- Furthermore, the data shows that light to moderate exercise is the best type of exercise for improving sleep (less than 10,000 steps). Bellabeat may recommend this level of exercise for people who want to live a healthy lifestyle but do not participate in high-level sports.
- Bellabeat may also think about gamification for some users who aren't motivated by notifications. The game can be designed to reward players based on the number of steps they take each day. To advance to the next level, you must maintain your activity level for a period of time (perhaps a month). For each level, you will receive a certain number of stars that can be redeemed for merchandise or discounts on other Bellabeat products.

Thank you very much!

Special thanks to Miguel Fzzz for his contribution to the open source community in assisting people to learn and be influenced by the approach by referring to case studies (as used in this analysis).