BellaBeat-Google Data Analytics Capstone Project (With R)

Olumide Olaoye

11/22/2021

Google Data Analytics Capstone Project (With R)

Case Study: How can a Wellness Technology Company play it smart?

Introduction

Welcome to the Bellabeat data analysis case study!

In this case study, you will perform many real-world tasks of a junior data analyst. You will imagine you are working for Bellabeat, a high-tech manufacturer of health-focused products for women, and meet different characters and team members. In order to answer the key business questions, you will follow the steps of the data analysis process: ask, prepare, process, analyze, share, and act. Along the way, the Case Study Roadmap tables — including guiding questions and key tasks — will help you stay on the right path.

By the end of this lesson, you will have a portfolio-ready case study. Download the packet and reference the details of this case study anytime. Then, when you begin your job hunt, your case study will be a tangible way to demonstrate your knowledge and skills to potential employers.

Scenario

You are a junior data analyst working on the marketing analyst team at Bellabeat, a high-tech manufacturer of health-focused products for women. Bellabeat is a successful small company, but they have the potential to become a larger player in the global smart device market. Urška Sršen, cofounder and Chief Creative Officer of Bellabeat, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company. You have been asked to focus on one of Bellabeat's products and analyze smart device data to gain insight into how consumers are using their smart devices. The insights you discover will then help guide marketing strategy for the company. You will present your analysis to the Bellabeat executive team along with your high-level recommendations for Bellabeat's marketing strategy.

Characters and products

- Characters
  - Urška Sršen: Bellabeat's cofounder and Chief Creative Officer

○ Sando Mur: Mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team

○ Bellabeat marketing analytics team: A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy. You joined this team six months ago and have been busy learning about Bellabeat''s mission and business goals — as well as how you, as a junior data analyst, can help Bellabeat achieve them.

● Products

○ 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.

About the company

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. By 2016, Bellabeat had opened offices around the world and launched multiple products. Bellabeat products became available through a growing number of online retailers in addition to their own e-commerce channel on their website. The company has invested in traditional advertising media, such as radio, out-of-home billboards, print, and television, but focuses on digital marketing extensively. Bellabeat invests year-round in Google Search, maintaining active Facebook and Instagram pages, and consistently engages consumers on Twitter. Additionally, Bellabeat runs video ads on Youtube and display ads on the Google Display Network to support campaigns around key marketing dates. Sršen knows that an analysis of Bellabeat's available

consumer data would reveal more opportunities for growth. She has asked the marketing analytics team to focus on a Bellabeat product and analyze smart device usage data in order to gain insight into how people are already using their smart devices. Then, using this information, she would like high-level recommendations for how these trends can inform Bellabeat marketing strategy

STEP 1: ASK

Business task:

Analyze consumers use of an existing competitor to identify potential opportunities for growth and recommendations for the Bellabeat marketing strategy

Questions for the analysis:

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

Key Stakeholders:

· Urška Sršen — Bellabeat's cofounder and Chief Creative Officer

· Sando Mur — Mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team

· Bellabeat marketing analytics team — A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy.

STEP 2: PREPARE

The data for this analysis will come from FitBit Fitness Tracker Data on Kaggle. These 18 datasets were generated by respondents to a distributed survey via Amazon Mechanical Turk between 03.12.2016–05.12.2016. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. Individual reports can be parsed by export session ID (column A) or timestamp (column B). Variation between output represents use of different types of Fitbit trackers and individual tracking behaviors / preferences.

Limitations for this data exist due to the sample size and absence of key characteristics of the participants, such as gender, age, location, lifestyle.

For this analysis the datasets for daily activity, daily calories, daily intensities, daily steps, heartrate by seconds, minute METs, daily sleep, and weight log information, will be used.

Because of the largeness of the datasets being used, R Studio was used to prepare, process and complete this analysis of which the many packages and data visualization features available therein can be used to explore the data.

Setting up my environment by installing the packages

```
# install.packages("tidyverse")
# install.packages("here")
# install.packages("janitor")
# install.packages("skimr")
# install.packages("dplyr")
```

Loading the packages

```
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_conflict
s() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(here)

## here() starts at C:/Users/user/OneDrive/Documents

library(janitor)

##
## Attaching package: 'janitor'

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

library(skimr)
library(dplyr)
```

Importing the datasets after cleaning and preparing in Excel

```
daily_activity <- read_csv("C:\\Users\\user\\Downloads\\Fitabase_Data\\dailyA
ctivity_merged.csv")

## Rows: 940 Columns: 15

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

```
##
## 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 m
essage.

daily_calories <- read_csv("C:\\Users\\user\\Downloads\\Fitabase_Data\\dailyC
alories_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 m
essage.

daily_intensities <- read_csv("C:\\Users\\user\\Downloads\\Fitabase_Data\\dai
lyIntensities_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 m
essage.

daily_steps <- read_csv("C:\\Users\\user\\Downloads\\Fitabase_Data\\dailySteps
s_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 m
essage.

minute_METs <- read_csv("C:\\Users\\user\\Downloads\\Fitabase_Data\\minuteMET
sNarrow_merged.csv")

## Rows: 1048575 Columns: 3

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

##
## 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 m
essage.

heart_rate_sec <- read_csv("C:\\Users\\user\\Downloads\\Fitabase_Data\\heartr
ate_seconds_merged.csv")

## Rows: 1048575 Columns: 3

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

##
## 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 m
essage.

sleep_day <- read_csv("C:\\Users\\user\\Downloads\\Fitabase_Data\\sleepDay_me
rged.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 m
essage.

weight_log <- read_csv("C:\\Users\\user\\Downloads\\Fitabase_Data\\weightLogI
nfo_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 m
essage.
```

Viewing the dataframes

daily_activity

```
head(daily_activity)

## # A tibble: 6 x 15
##        Id ActivityDate TotalSteps TotalDistance TrackerDistance LoggedActi
vitie~
##     <dbl> <chr>             <dbl>         <dbl>           <dbl>
<dbl>
## 1  1.50e9 4/12/2016         13162          8.5             8.5
0
## 2  1.50e9 4/13/2016         10735          6.97            6.97
0
## 3  1.50e9 4/14/2016         10460          6.74            6.74
0
## 4  1.50e9 4/15/2016          9762          6.28            6.28
0
## 5  1.50e9 4/16/2016         12669          8.16            8.16
0
## 6  1.50e9 4/17/2016          9705          6.48            6.48
0
## # ... with 9 more variables: VeryActiveDistance <dbl>,
## #   ModeratelyActiveDistance <dbl>, LightActiveDistance <dbl>,
## #   SedentaryActiveDistance <dbl>, VeryActiveMinutes <dbl>,
## #   FairlyActiveMinutes <dbl>, LightlyActiveMinutes <dbl>,
## #   SedentaryMinutes <dbl>, Calories <dbl>

colnames(daily_activity)

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

```
## [13] "LightlyActiveMinutes"     "SedentaryMinutes"
## [15] "Calories"
```

```
glimpse(daily_activity)
```

```
## Rows: 940
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 15039
6036~
## $ ActivityDate             <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4
/15/~
## $ TotalSteps               <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 1
3019~
## $ TotalDistance            <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59,
9.8~
## $ TrackerDistance          <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59,
9.8~
## $ LoggedActivitiesDistance <dbl> 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,
3.5~
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64,
1.3~
## $ LightActiveDistance      <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71,
5.0~
## $ SedentaryActiveDistance  <dbl> 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, 6
6, 4~
## $ FairlyActiveMinutes      <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27
, 21~
## $ LightlyActiveMinutes     <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 2
05, ~
## $ SedentaryMinutes         <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775,
818~
## $ Calories                 <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921,
203~
```

daily_calories

```
head(daily_calories)
```

```
## # A tibble: 6 x 3
##           Id ActivityDay Calories
##        <dbl> <chr>          <dbl>
## 1 1503960366 4/12/2016       1985
## 2 1503960366 4/13/2016       1797
## 3 1503960366 4/14/2016       1776
## 4 1503960366 4/15/2016       1745
## 5 1503960366 4/16/2016       1863
## 6 1503960366 4/17/2016       1728
```

```
colnames(daily_calories)
```

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

```
glimpse(daily_calories)
```

```
## Rows: 940
## Columns: 3
## $ Id          <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150396
0366~
## $ ActivityDay <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/2016", "4
/16/~
## $ Calories    <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 2035, 1786,
1775~
```

daily_intensities

```
head(daily_intensities)
```

```
## # A tibble: 6 x 10
##           Id ActivityDay SedentaryMinutes LightlyActiveMinutes FairlyActiv
eMinu~
##        <dbl> <chr>                 <dbl>                <dbl>
<dbl>
## 1 1503960366 4/12/2016               728                  328
13
## 2 1503960366 4/13/2016               776                  217
19
## 3 1503960366 4/14/2016              1218                  181
11
## 4 1503960366 4/15/2016               726                  209
34
## 5 1503960366 4/16/2016               773                  221
10
## 6 1503960366 4/17/2016               539                  164
20
## # ... with 5 more variables: VeryActiveMinutes <dbl>,
## #   SedentaryActiveDistance <dbl>, LightActiveDistance <dbl>,
## #   ModeratelyActiveDistance <dbl>, VeryActiveDistance <dbl>
```

```
colnames(daily_intensities)
```

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

```
glimpse(daily_intensities)
```

```
## Rows: 940
## Columns: 10
```

```
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 15039
6036~
## $ ActivityDay              <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4
/15/~
## $ SedentaryMinutes         <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775,
818~
## $ LightlyActiveMinutes     <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 2
05, ~
## $ FairlyActiveMinutes      <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27
, 21~
## $ VeryActiveMinutes        <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 6
6, 4~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, ~
## $ LightActiveDistance      <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71,
5.0~
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64,
1.3~
## $ VeryActiveDistance       <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25,
3.5~
```

daily_steps

```
head(daily_steps)
```

```
## # A tibble: 6 x 3
##           Id ActivityDay StepTotal
##        <dbl> <chr>           <dbl>
## 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
```

```
colnames(daily_steps)
```

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

```
glimpse(daily_steps)
```

```
## Rows: 940
## Columns: 3
## $ Id          <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150396
0366~
## $ ActivityDay <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/2016", "4
/16/~
## $ StepTotal   <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 13019, 15506,
1054~
```

minute_METs

```
head(minute_METs)
```

```
## # A tibble: 6 x 3
##           Id ActivityMinute  METs
##        <dbl> <time>         <dbl>
## 1 1503960366 00'00"            10
## 2 1503960366 01'00"            10
## 3 1503960366 02'00"            10
## 4 1503960366 03'00"            10
## 5 1503960366 04'00"            10
## 6 1503960366 05'00"            12
```

```
colnames(minute_METs)
```

```
## [1] "Id"             "ActivityMinute" "METs"
```

```
glimpse(minute_METs)
```

```
## Rows: 1,048,575
## Columns: 3
## $ Id             <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150
3960~
## $ ActivityMinute <time> 00:00:00, 00:01:00, 00:02:00, 00:03:00, 00:04:00,
00:0~
## $ METs           <dbl> 10, 10, 10, 10, 10, 12, 12, 12, 12, 12, 12, 12, 10,
10,~
```

heart_rate_sec

```
head(heart_rate_sec)
```

```
## # A tibble: 6 x 3
##           Id Time   Value
##        <dbl> <time> <dbl>
## 1 2022484408 07:21     97
## 2 2022484408 07:21    102
## 3 2022484408 07:21    105
## 4 2022484408 07:21    103
## 5 2022484408 07:21    101
## 6 2022484408 07:22     95
```

```
colnames(heart_rate_sec)
```

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

```
glimpse(heart_rate_sec)
```

```
## Rows: 1,048,575
## Columns: 3
## $ Id    <dbl> 2022484408, 2022484408, 2022484408, 2022484408, 2022484408,
2022~
## $ Time  <time> 07:21:00, 07:21:00, 07:21:00, 07:21:00, 07:21:00, 07:22:00,
```

```
07:~
## $ Value <dbl> 97, 102, 105, 103, 101, 95, 91, 93, 94, 93, 92, 89, 83, 61,
60, ~
```

sleep_day

```
head(sleep_day)
```

```
## # A tibble: 6 x 5
##           Id SleepDay  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##        <dbl> <chr>                 <dbl>              <dbl>          <dbl>
## 1 1503960366 4/12/2016                 1                327            346
## 2 1503960366 4/13/2016                 2                384            407
## 3 1503960366 4/15/2016                 1                412            442
## 4 1503960366 4/16/2016                 2                340            367
## 5 1503960366 4/17/2016                 1                700            712
## 6 1503960366 4/19/2016                 1                304            320
```

```
colnames(sleep_day)
```

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

```
glimpse(sleep_day)
```

```
## Rows: 413
## Columns: 5
## $ Id                 <dbl> 1503960366, 1503960366, 1503960366, 1503960366,
150~
## $ SleepDay           <chr> "4/12/2016", "4/13/2016", "4/15/2016", "4/16/20
16",~
## $ TotalSleepRecords  <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, ~
## $ TotalMinutesAsleep <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, 43
0, 2~
## $ TotalTimeInBed     <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, 44
9, 3~
```

weight_log

```
head(weight_log)
```

```
## # A tibble: 6 x 8
##           Id Date      WeightKg WeightPounds   Fat   BMI IsManualReport
LogId
##        <dbl> <chr>        <dbl>        <dbl> <dbl> <dbl> <lgl>
<dbl>
## 1 1503960366 5/2/2016      52.6         116.    22  22.6 TRUE           1
.46e12
## 2 1503960366 5/3/2016      52.6         116.    NA  22.6 TRUE           1
.46e12
## 3 1927972279 4/13/2016    134.          294.    NA  47.5 FALSE          1
```

```
.46e12
## 4 2873212765 4/21/2016     56.7         125     NA  21.4 TRUE            1
.46e12
## 5 2873212765 5/12/2016     57.3         126.    NA  21.7 TRUE            1
.46e12
## 6 4319703577 4/17/2016     72.4         160.    25  27.4 TRUE            1
.46e12
```

```
colnames(weight_log)
```

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

```
glimpse(weight_log)
```

```
## Rows: 67
## Columns: 8
## $ Id             <dbl> 1503960366, 1503960366, 1927972279, 2873212765, 287
3212~
## $ Date           <chr> "5/2/2016", "5/3/2016", "4/13/2016", "4/21/2016", "
5/12~
## $ WeightKg       <dbl> 52.6, 52.6, 133.5, 56.7, 57.3, 72.4, 72.3, 69.7, 70
.3, ~
## $ WeightPounds   <dbl> 115.96, 115.96, 294.32, 125.00, 126.32, 159.61, 159
.39,~
## $ Fat            <dbl> 22, NA, NA, NA, NA, 25, NA, NA, NA, NA, NA, NA, NA,
NA,~
## $ BMI            <dbl> 22.65, 22.65, 47.54, 21.45, 21.69, 27.45, 27.38, 27
.25,~
## $ IsManualReport <lgl> TRUE, TRUE, FALSE, TRUE, TRUE, TRUE, TRUE, TRUE, TR
UE, ~
## $ LogId          <dbl> 1.46e+12, 1.46e+12, 1.46e+12, 1.46e+12, 1.46e+12, 1
.46e~
```

Removing data frames:

The sqldf package is loaded to utilize SQL syntax to determine if the values of daily_calories, daily_intensities, and daily_steps are contained in daily_activity.

However, the number of columns must be the same between the data frames, so a temporary data frame with the important columns is created first.

```
# install.packages("sqldf")
```

```
library(sqldf)
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Loading required package: RSQLite
```

```r
daily_activity_2 <- daily_activity %>%
  select(Id, ActivityDate, Calories)

head(daily_activity_2)
```

```
## # A tibble: 6 x 3
##           Id ActivityDate Calories
##        <dbl> <chr>           <dbl>
## 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
```

```r
sql_check <- sqldf('SELECT * FROM daily_activity_2 INTERSECT SELECT *
                   FROM daily_calories')

head(sql_check)
```

```
##           Id ActivityDate 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
```

```r
nrow(daily_activity_2)
```

```
## [1] 940
```

```r
nrow(sql_check)
```

```
## [1] 940
```

```r
daily_activity_3 <- daily_activity %>%
  select(Id, ActivityDate, SedentaryMinutes, LightlyActiveMinutes,
         FairlyActiveMinutes, VeryActiveMinutes, SedentaryActiveDistance,
         LightActiveDistance, ModeratelyActiveDistance, VeryActiveDistance)

head(daily_activity_3)
```

```
## # A tibble: 6 x 10
##           Id ActivityDate SedentaryMinutes LightlyActiveMinutes FairlyActi
veMin~
##        <dbl> <chr>                  <dbl>                <dbl>
<dbl>
## 1 1503960366 4/12/2016                728                  328
13
## 2 1503960366 4/13/2016                776                  217
19
```

```
## 3 1503960366 4/14/2016                1218                   181
11
## 4 1503960366 4/15/2016                 726                   209
34
## 5 1503960366 4/16/2016                 773                   221
10
## 6 1503960366 4/17/2016                 539                   164
20
## # ... with 5 more variables: VeryActiveMinutes <dbl>,
## #   SedentaryActiveDistance <dbl>, LightActiveDistance <dbl>,
## #   ModeratelyActiveDistance <dbl>, VeryActiveDistance <dbl>
```

```
nrow(daily_activity_3)
```

```
## [1] 940
```

```
sql_check_2 <- sqldf('SELECT * FROM daily_activity_3 INTERSECT SELECT *
                     FROM daily_intensities')
```

```
head(sql_check_2)
```

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

```
nrow(sql_check_2)
```

```
## [1] 940
```

```
daily_activity_4 <- daily_activity %>%
  select(Id, ActivityDate, TotalSteps)
```

```
head(daily_activity_4)
```

```
## # A tibble: 6 x 3
##           Id ActivityDate TotalSteps
##        <dbl> <chr>             <dbl>
## 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
```

```
sql_check_3 <- sqldf('SELECT * FROM daily_activity_4 INTERSECT SELECT *
                     FROM daily_steps')
```

```
head(sql_check_3)
```

```
##           Id ActivityDate TotalSteps
## 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
```

```
nrow(daily_activity_4)
```

```
## [1] 940
```

```
nrow(sql_check_3)
```

```
## [1] 940
```

The outputs of the head() function of the temporary data frames created, match the outputs of the head() function for the original data frames.

The outputs of the head() function of the SQL data frames match the outputs of the head() function for the temporary data frames.

The number of observations for each SQL data frame are equal to 940.

Conclusively, the data for the daily_calories, daily_intensities, and daily_steps data frames are contained in daily_activity. These three data frames will be removed from the analysis for simplicity.

STEP 4: Analyze

Summarizing the data:

The n_distinct() and nrow() functions are used to determine the number of unique values and the number of rows in a data frame, respectively.

```
n_distinct(daily_activity$Id)
```

```
## [1] 33

n_distinct(minute_METs$Id)

## [1] 27

n_distinct(heart_rate_sec$Id)

## [1] 7

n_distinct(sleep_day$Id)

## [1] 24

n_distinct(weight_log$Id)

## [1] 8

nrow(daily_activity)

## [1] 940

nrow(minute_METs)

## [1] 1048575

nrow(heart_rate_sec)

## [1] 1048575

nrow(sleep_day)

## [1] 413

nrow(weight_log)

## [1] 67
```

The heart rate and weight log data frames contain a very low number of participants based on the n_distinct() outputs. Thus, reliable recommendations and conclusions cannot be made solely from these data frames.

The summary() function is used to pull key statistics about the data frames.

daily_activity:

```
daily_activity %>%
  select(TotalSteps, TotalDistance, SedentaryMinutes, LightlyActiveMinutes,
         FairlyActiveMinutes, VeryActiveMinutes, Calories) %>%
  summary()

##    TotalSteps    TotalDistance    SedentaryMinutes LightlyActiveMinutes
##  Min.   :    0   Min.   : 0.000   Min.   :   0.0   Min.   :  0.0
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 729.8   1st Qu.:127.0
##  Median : 7406   Median : 5.245   Median :1057.5   Median :199.0
```

```
##  Mean   : 7638   Mean   : 5.490   Mean   : 991.2   Mean   :192.8
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:1229.5   3rd Qu.:264.0
##  Max.   :36019   Max.   :28.030   Max.   :1440.0   Max.   :518.0
##  FairlyActiveMinutes VeryActiveMinutes    Calories
##  Min.   :  0.00      Min.   :  0.00    Min.   :   0
##  1st Qu.:  0.00      1st Qu.:  0.00    1st Qu.:1828
##  Median :  6.00      Median :  4.00    Median :2134
##  Mean   : 13.56      Mean   : 21.16    Mean   :2304
##  3rd Qu.: 19.00      3rd Qu.: 32.00    3rd Qu.:2793
##  Max.   :143.00      Max.   :210.00    Max.   :4900
```

This summary shows the average user is taking 7638 steps per day, missing the recommended 10,000 steps for health by the Centre for Disease Control (CDC). On average, users are getting 21.16 minutes of very active or vigorous activity a day, this equates to 148.12 minutes a week. The CDC recommends 75 minutes of vigorous activity a week, so the typical Fitbit user is doing well in this area and achieving additional health benefits. In contrast, participants are averaging 991.2 minutes, or 16.52 hours of sedentary time a day! This is a significant amount of time and can lead to other health issues because the body functions best upright. Scientists have determined that 40 minutes of moderate to vigorous activity a day will balance out the effects of sitting up to 10 hours a day.

Furthermore, this summary shows the average user is burning 2304 calories per day. Studies show the average person in the population burns 1800 calories a day, but burning 3500 is needed to lose a pound of weight.

The Fitbit users in this case are burning more than normal, and are on track to lose a few pounds a week if they so choose.

heart_rate_sec:

```
heart_rate_sec %>%
  select(Value) %>%
  summary()

##      Value
##  Min.   : 38.00
##  1st Qu.: 64.00
##  Median : 75.00
##  Mean   : 77.02
##  3rd Qu.: 87.00
##  Max.   :203.00
```

Despite the low number of users in the heart rate data frame, the average heartrate of 77 beats per minute (bpm) fits within the “normal” range.

The ranges between 50 to 80 bpm for men, and 53 to 82 bpm for women are considered Normal.

However, finding suggests that it is more important for individuals to determine what is a normal and healthy heartrate for them, and not compare to population levels. This is

because resting heart rates between different people can vary by as much as 70 bpm. Changes in resting heartrate over days can be a sign of infection, menstrual cycle effects, or other acute triggers.

Thus, making heartrate a vital health characteristic to monitor.

minute_METs:

```
minute_METs %>%
  select(METs) %>%
  summary()
```

```
##       METs
##  Min.   :  0.00
##  1st Qu.: 10.00
##  Median : 10.00
##  Mean   : 14.47
##  3rd Qu.: 11.00
##  Max.   :157.00
```

The summary of minute METs shows the average user has a MET of 14.47.

A MET is the division of your working metabolic rate and resting metabolic rate. One MET is the energy your body consumes when at rest. This means an activity with a MET of four, would require a person to exert four times the energy they do when they are sitting. Therefore, a user averaging 14.47 MET throughout the day is considerably high, which leads to the assumption that the Fitbit is not calculating this data point correctly. Due to this, the minute MET data frame will not be used further in this analysis.

sleep_day:

```
sleep_day %>%
  select(TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed) %>%
  summary()
```

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

The summary of the sleep data frame shows that the average user sleeps once per day for 419.5 minutes, or roughly 7 hours. This falls within the CDC's recommendations for adults in order to get the proper amount of rest.

The average participant is spending 458.6 minutes in bed, or 7.64 hours.

This means the typical user is spending 38.6 minutes awake in bed.

According to Health Central, people should not spend more than 1 hour in bed awake. This is to prevent a mental link being formed between being awake and being in bed, which can lead to insomnia.

weight_log:

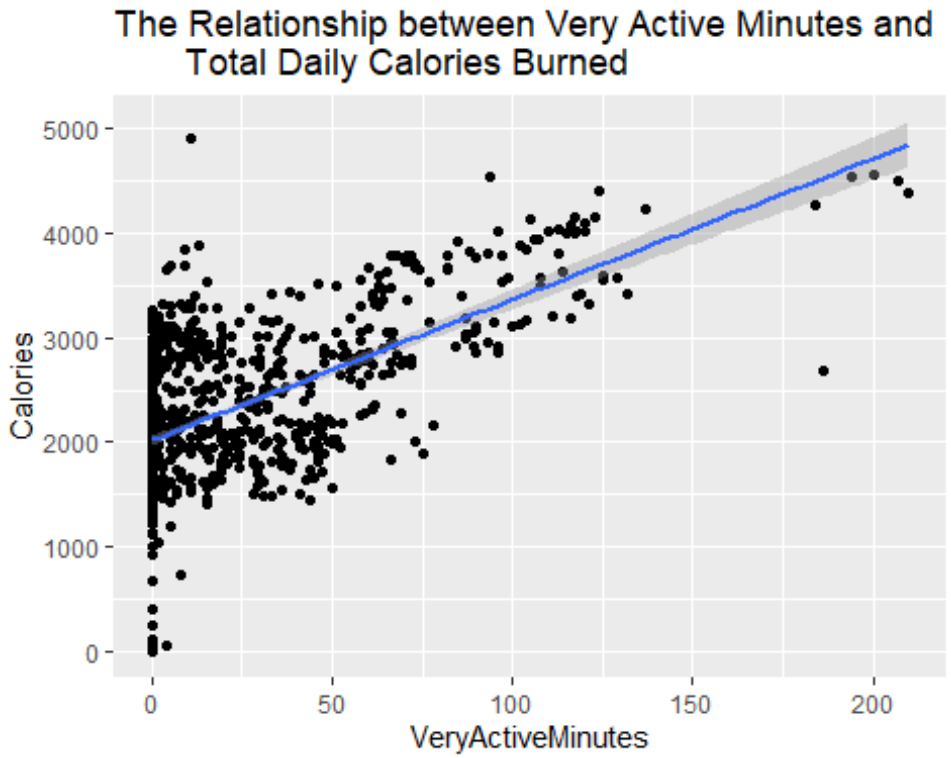
```
weight_log %>%
  select(WeightPounds, BMI) %>%
  summary()

##   WeightPounds        BMI
##  Min.   :116.0   Min.   :21.45
##  1st Qu.:135.4   1st Qu.:23.96
##  Median :137.8   Median :24.39
##  Mean   :158.8   Mean   :25.19
##  3rd Qu.:187.5   3rd Qu.:25.56
##  Max.   :294.3   Max.   :47.54
```

While this data frame has a low number of participants, the average BMI is 25.19. This is considered an overweight BMI. However, BMI can be a screening tool and does not diagnose the body fatness or health of an individual.

STEP 5: Share

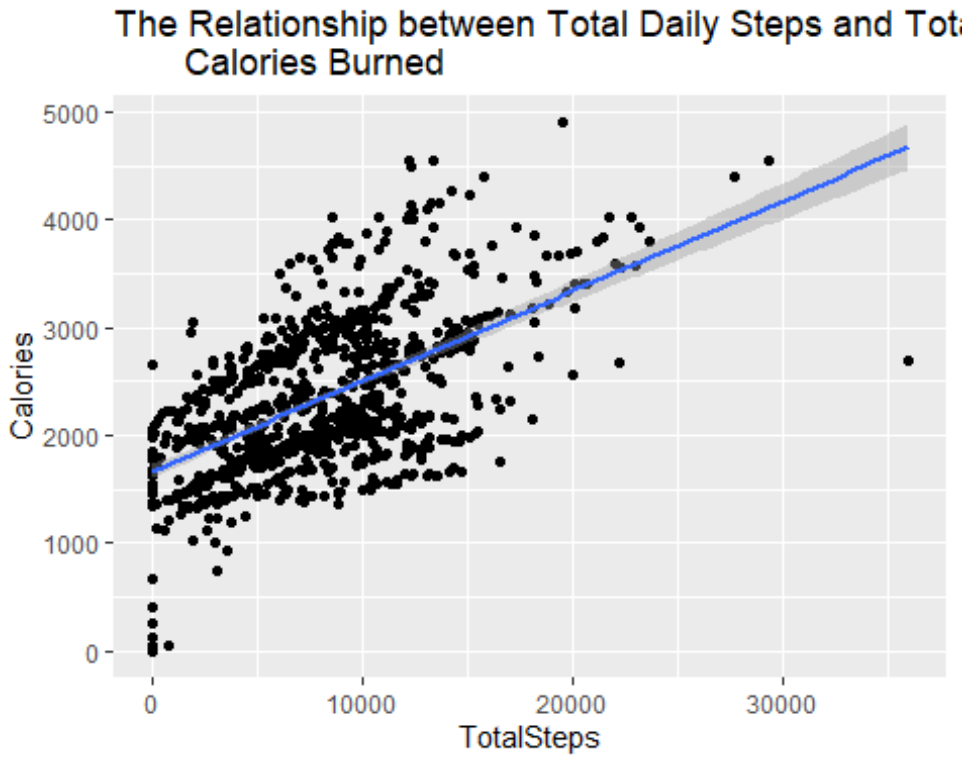
The ggplot() function of R Studio was used to create data visualizations that show patterns and trends found in the data frames.

```
ggplot(data=daily_activity, aes(x=VeryActiveMinutes, y=Calories))+
  geom_point()+
  stat_smooth(method=lm)+
  labs(title="The Relationship between Very Active Minutes and
       Total Daily Calories Burned")

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



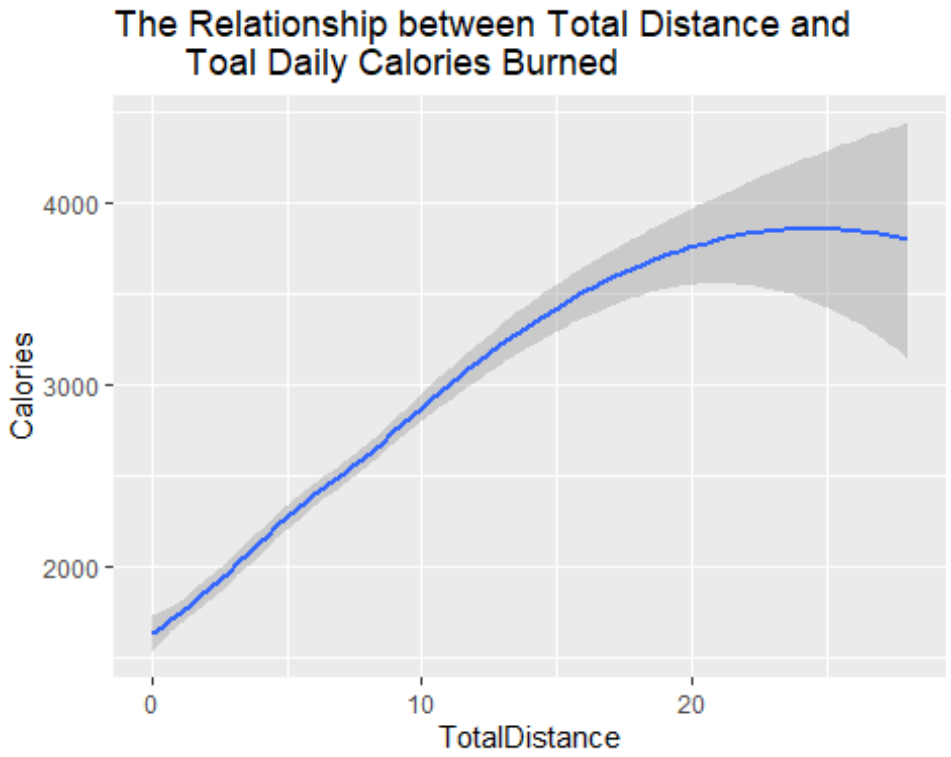
The 1st plot above displays a positive relationship between very active minutes and total daily calories burned. This means that the more vigorous physical activity the participant did, the more calories they burned.

```
ggplot(data=daily_activity, aes(x=TotalSteps, y=Calories))+
  geom_point()+
  stat_smooth(method=lm)+
  labs(title="The Relationship between Total Daily Steps and Total Daily
       Calories Burned")

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



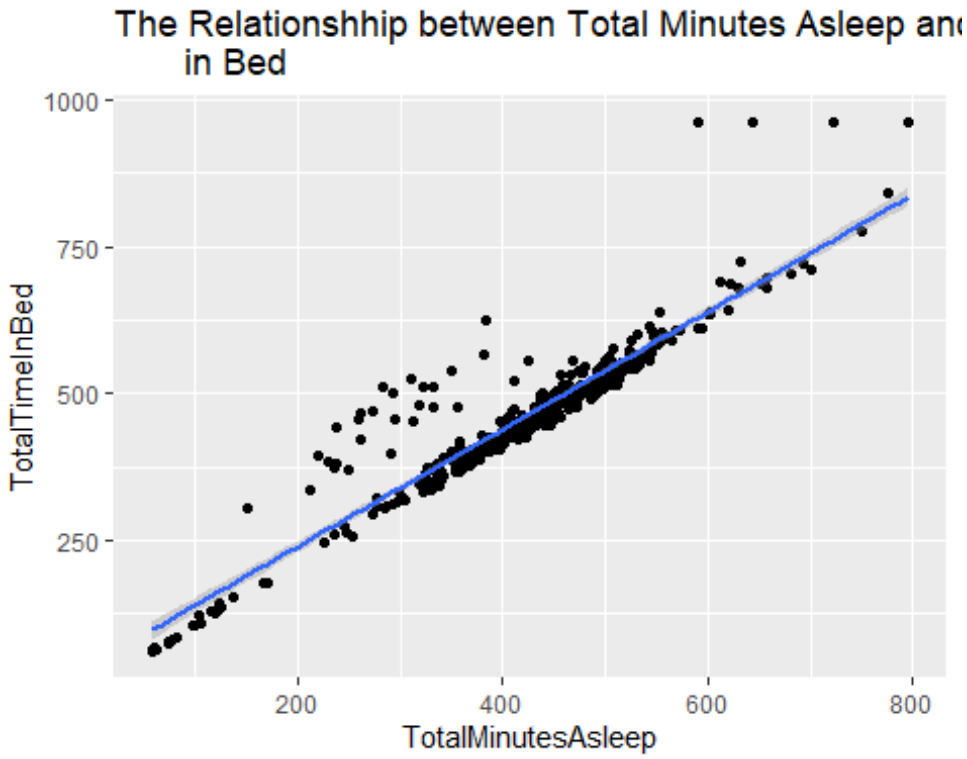
The 2nd plot above shows a positive relationship between total daily steps taken and total calories burned. This means the more steps the Fitbit users took, the more calories they burned.

```
ggplot(data=daily_activity, aes(x=TotalDistance, y=Calories))+
  geom_smooth()+
  labs(title="The Relationship between Total Distance and
       Toal Daily Calories Burned")

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



The 3rd plot depicts a positive trend between total distance and total daily calories burned. As the participants moved a greater distance, the number of calories they burned also increased.

```
ggplot(data=sleep_day, aes(x=TotalMinutesAsleep, y=TotalTimeInBed))+
  geom_point()+
  stat_smooth(method=lm)+
  labs(title="The Relationshhip between Total Minutes Asleep and Total Time
       in Bed")

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



The 4th plot shows a positive relationship between total minutes asleep and total time in bed. For the most part, the time participants spent asleep and the time they spent in bed was very similar.

STEP 6: Act

Bellabeat has been successful since it was founded by empowering women through providing data on their activity, sleep, stress, hydration levels, and reproductive health. Based on analyzing how Fitbit consumers use and respond to features, recommendations can be made to promote further growth for Bellabeat.

The Bellabeat app should be completely enhanced and revamped. Rather than simply providing data on user's health, the app should further encourage users to meet fitness goals and become a social media platform.

The CDC recommends working out with a friend in order to feel more motivated, be more adventurous in trying workouts, and to become consistent.

The CDC even recommends the use of a social media workout app to connect with friends and reach your goals. The Bellabeat app could become that social media workout app that women turn to, by creating an online community of supportive women ready to prioritize their health.

Recommendations for Bellabeat app:

1. Enable social networking so users can post their favorite workouts, wellness tips, healthy meals, etc.

2. Create weekly fitness and wellness challenges to encourage use.
3. Have health and fitness companies pay for advertising.
4. Enable users to add friends and view each other's activity.
5. Recommend users get 75 minutes of vigorous activity a week and enable alert notifications to entourage users to meet this.
6. Recommend users to get 10,000 steps a day and enable alert notifications to encourage users to meet goal.
7. Recommend users to get at least 7 hours of sleep a night and enable alert notifications to encourage users to meet this.
8. Encourage users to enter in weight and height to track BMI.
9. If users are interested in losing weight, enable notifications to keep users on track to burn necessary calories to meet goal.
10. Enable alert notifications if user's resting heart rate varies significantly from their normal.
11. Enable notifications to encourage activity if a user has spent an hour in bed awake.
12. Enable notifications to encourage activity if a user has been sedentary for an extended period of time.

Recommendations for Bellabeat membership:

1. Offer discounts for Bellabeat smart device products with membership.
2. Partner with health & fitness companies and offer discounts for members.
3. Offer 30-day free trial subscription.
4. Offer reduced subscription fee when a member refers a friend.

Recommendations for Bellabeat products:

1. Offer a bundle deal for the Spring and Leaf together.
2. Heavily market Spring as Fitbit does not track hydration levels.

Works Cited

"The Dangers of Sitting: Why Sitting Is the New Smoking." The Dangers of Sitting: Why Sitting Is the New Smoking — Better Health Channel, 22 Aug. 2020, www.betterhealth.vic.gov.au/health/healthyliving/the-dangers-of-sitting

"3 Reasons to Work out with a Friend." Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, 23 Apr. 2021, www.cdc.gov/diabetes/library/spotlights/workout-buddy.html

"About Adult Bmi." Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, 17 Sept. 2020, www.cdc.gov/healthyweight/assessing/bmi/adult_bmi/index.html

Gornall, Lucy. "How to Lose Weight: How Many Calories Should i Eat to Lose Weight?" GoodtoKnow, 12 Aug. 2020, www.goodto.com/wellbeing/diets-exercise/what-is-calorie-how-many-lose-weigt-425557

"CDC — How Much Sleep Do I Need? — Sleep and Sleep Disorders." Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, 2 Mar. 2017, www.cdc.gov/sleep/about_sleep/how_much_sleep.html

Reed, Martin. "Spend Less Time In Bed If You Want More Sleep." Healthcentral.com, 7 May 2017, www.healthcentral.com/article/spend-less-time-in-bed-if-you-want-more-sleep

Roland, James. "What Are Mets, and How Are They Calculated?" Healthline, Healthline Media, 21 Oct. 2019, www.healthline.com/health/what-are-mets#calculation

Grey, Heather. "Heart Rates Can Vary by 70 Bpm: What That Means for Your Health." Healthline, Healthline Media, 9 Feb. 2020, www.healthline.com/health-news/what-your-heart-rate-says-about-your-health

"How Much Physical Activity Do Adults Need?" Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, 7 Oct. 2020, www.cdc.gov/physicalactivity/basics/adults/index.htm

Nield, David. "Scientists Figured out How Much Exercise You Need to 'Offset' a Day of Sitting." ScienceAlert, 26 Nov. 2020, www.sciencealert.com/getting-a-sweat-on-for-30-40-minutes-could-offset-a-day-of-sitting-down