Google Capstone Project II: How Can Bellabeat Play It Smart?

Adedoyin Tiamiyu

2022-06-04

Introduction

I recently completed the Google Data Analytics Certification offered on Coursera. I learned about the six phases of the data process (ask, prepare, process, analyze, share, act) and how to use the technical tools (Spreadsheet, SQL, Tableau, and R). To complete the course the students were required to create their capstone project to highlight the technical skills they learned and showcase their understanding of the six steps of the data analysis process.

In this report, I will show my approach to answering some of the business questions and solving the problem of the capstone project.

Overview

Bellabeat is a high-tech company that manufactures health-focused products for women, with the potential for growth in the global-smart economy, since 2013 and was founded by Urška Sršen and Sando Mur. Inspiring and empowering women with knowledge about their health and habits, Bellabeat has grown rapidly and quickly positioned itself as a tech-driven wellness company for females.

1. ASK

In this step, we define the problem and state the objectives of our case study.

Business Task

Analyze and discover trends in how the customers use the smart device to gain insight and help guide the Bellabeat marketing strategy for global growth.

Key Business questions

How do users use the smart device and how can the insight be used to help influence Bellabeat marketing strategy?

The analysis of the data provided will help identify opportunities for improvement in how users interact with the device and help make recommendations according to their needs.

Key Stakeholders

- Urška Sršen: Bellabeat's co-founder and Chief Creative Officer
- Sando Mur: Mathematician and Bellabeat's co-founder; a key member of the Bellabeat executive team
- Bellabeat marketing analytics team: The marketing analyst team
- The customers (External Stakeholders)

2. PREPARE

Data Source Information

The data used is publicly available on Kaggle; Fitbit Fitness Tracker Data, the dataset was generated by respondents to a distributed survey via Amazon Mechanical Turk between 12 March 2016 to 12 May 2016 and stored in 18 CSV files. Thirty FitBit users consented to the submission of personal tracker data and the data collected includes physical activity recorded in minutes, heart rate, sleep monitoring, daily activity, and steps. Data Credibility and Integrity

I will use the "ROCCC" system to determine the credibility and integrity of the datasets.

- Reliability: This data is not reliable because it has only 30 respondents which means that the data might not represent a major part of the fitness population and makes the data biased.

- Originality: This is not an original dataset as it was originally collected from a third party (Amazon Mechanical Murk).

- Comprehensiveness: This data is not comprehensive. There is no information about the participants, such as gender, age, health state, etc. This could mean that data was not randomized.

- Current: The data was collected five years ago which makes it outdated and may not represent the current trends in smart device usage.

- Cited: As stated before, Amazon Mechanical Murk created the dataset, but we have no information on whether this is a credible source.

The dataset is not ROCCC and it is not recommended for producing reliable insights for making business decisions.

Limitations of Data

- The data has a small sample size with only and does not represent the entire fitness population.

- Bellabeat is a company that is focused on women's products but there are no genders revealed in this dataset. Others key pieces of information missing include the users' ages and the region where the data was collected.

3. PROCESS

I will be using R for data cleaning, analysis, and visualization.

Loading and Installing Packages

I will be using these packages for the analysis

- tidyverse
- lubridate
- here
- skimr
- janitor
- ggpubr
- dplyr
- ggplot2

```
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   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

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

library(lubridate)

##
## Attaching package: 'lubridate'

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

library(here)

## here() starts at C:/Users/adedo/OneDrive/Documents/Bella_beat
```

```
library(skimr)
library(janitor)

##
## Attaching package: 'janitor'

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

library(ggpubr)
library(dplyr)
library(ggplot2)
```

Importing Dataset

```
daily_activity <- read.csv("dailyActivity_merged.csv")
daily_calories <- read.csv("dailyCalories_merged.csv")
daily_intensities <- read.csv("dailyIntensities_merged.csv")
daily_steps <- read.csv("dailySteps_merged.csv")
daily_heartrate <- read.csv("heartrate_seconds_merged.csv")
daily_sleep <- read.csv("sleepDay_merged.csv")
daily_weight <- read.csv("weightlogInfo_merged.csv")
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Data Cleaning, Manipulation, and Exploring

I will explore the data, and look for irregularities such as outliers and null values. Transform the data so it is ready for analysis and document the cleaning process.

Viewing Datasets

```
str(daily_activity)

## 'data.frame':    940 obs. of  15 variables:
##  $ Id                       : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09
...
##  $ ActivityDate             : chr  "4/12/2016" "4/13/2016" "4/14/2016"
"4/15/2016" ...
##  $ TotalSteps               : int  13162 10735 10460 9762 12669 9705 13019
15506 10544 9819 ...
##  $ TotalDistance            : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance          : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance       : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance      : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes        : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes      : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes     : int  328 217 181 209 221 164 233 264 205 211
```

```
...
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818
838 ...
##  $ Calories                : int  1985 1797 1776 1745 1863 1728 1921 2035
1786 1775 ...

str(daily_calories)

## 'data.frame':    940 obs. of  3 variables:
##  $ Id         : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDay: chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ Calories   : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...

str(daily_heartrate)

## 'data.frame':    2483658 obs. of  3 variables:
##  $ Id   : num  2.02e+09 2.02e+09 2.02e+09 2.02e+09 2.02e+09 ...
##  $ Time : chr  "4/12/2016 7:21:00 AM" "4/12/2016 7:21:05 AM" "4/12/2016
7:21:10 AM" "4/12/2016 7:21:20 AM" ...
##  $ Value: int  97 102 105 103 101 95 91 93 94 93 ...

str(daily_intensities)

## 'data.frame':    940 obs. of  10 variables:
##  $ Id                       : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09
...
##  $ ActivityDay              : chr  "4/12/2016" "4/13/2016" "4/14/2016"
"4/15/2016" ...
##  $ SedentaryMinutes         : int  728 776 1218 726 773 539 1149 775 818
838 ...
##  $ LightlyActiveMinutes     : int  328 217 181 209 221 164 233 264 205 211
...
##  $ FairlyActiveMinutes      : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ VeryActiveMinutes        : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LightActiveDistance      : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ VeryActiveDistance       : num  1.88 1.57 2.44 2.14 2.71 ...

str(daily_sleep)

## 'data.frame':    413 obs. of  5 variables:
##  $ Id                 : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay           : chr  "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00
AM" "4/15/2016 12:00:00 AM" "4/16/2016 12:00:00 AM" ...
##  $ TotalSleepRecords : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : int  346 407 442 367 712 320 377 364 384 449 ...

str(daily_weight)
```

```
## 'data.frame':    67 obs. of  8 variables:
##  $ Id            : num  1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...
##  $ Date          : chr  "5/2/2016 11:59:59 PM" "5/3/2016 11:59:59 PM"
"4/13/2016 1:08:52 AM" "4/21/2016 11:59:59 PM" ...
##  $ WeightKg      : num  52.6 52.6 133.5 56.7 57.3 ...
##  $ WeightPounds  : num  116 116 294 125 126 ...
##  $ Fat           : int  22 NA NA NA NA 25 NA NA NA NA ...
##  $ BMI           : num  22.6 22.6 47.5 21.5 21.7 ...
##  $ IsManualReport: chr  "True" "True" "False" "True" ...
##  $ LogId         : num  1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
```

Cleaning Dataset

checking for numbers of the unique ID in each dataframes

```
n_distinct(daily_activity$Id)

## [1] 33

n_distinct(daily_calories$Id)

## [1] 33

n_distinct(daily_intensities$Id)

## [1] 33

n_distinct(daily_steps$Id)

## [1] 33

n_distinct(daily_heartrate$Id)

## [1] 14

n_distinct(daily_sleep$Id)

## [1] 24

n_distinct(daily_weight$Id)

## [1] 8
```

Checking for duplicates

```
sum(duplicated(daily_activity))

## [1] 0

sum(duplicated(daily_calories))

## [1] 0

sum(duplicated(daily_heartrate))
```

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

```
sum(duplicated(daily_intensities))
```

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

```
sum(duplicated(daily_sleep))
```

```
## [1] 3
```

```
sum(duplicated(daily_weight))
```

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

Checking for Na

```
sum(is.na(daily_activity))
```

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

```
sum(is.na(daily_calories))
```

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

```
sum(is.na(daily_intensities))
```

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

```
sum(is.na(daily_heartrate))
```

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

```
sum(is.na(daily_sleep))
```

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

```
sum(is.na(daily_weight))
```

```
## [1] 65
```

Observations from exploring the datasets

- daily_calories, daily_intensities, and daily_steps are merged into daily_activity.
- daily_activity and daily_sleep have 33 and 24 unique IDs instead of 30.
- daily_sleep has 3 duplicates.
- daily_weight has too many missing values (65).
- The date data type is wrongly classified as a character.
- The variable name doesn't fit the naming convention.
- Some rows have zeros in total steps, which could mean that no distance was covered.
- Total distance and Tracker Distance on the daily_activity dataframe have the same data, and tracker distance will be removed later in the analysis.

- Logged activity distance on daily_activity dataframe has too many zeros which could mean that the users didn't log in their info. The column will be dropped later.

4. ANALYSE

For this analysis, I will be focusing on the daily_activity and daily_sleep dataframe as they are the datasets that will help discover trends and give useful insights.

Data Manipulation (daily_activity)

Viewing the Dataset

```
str(daily_activity)

## 'data.frame':    940 obs. of  15 variables:
##  $ Id                       : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09
...
##  $ ActivityDate             : chr  "4/12/2016" "4/13/2016" "4/14/2016"
"4/15/2016" ...
##  $ TotalSteps               : int  13162 10735 10460 9762 12669 9705 13019
15506 10544 9819 ...
##  $ TotalDistance            : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance          : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance       : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance      : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes        : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes      : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes     : int  328 217 181 209 221 164 233 264 205 211
...
##  $ SedentaryMinutes         : int  728 776 1218 726 773 539 1149 775 818
838 ...
##  $ Calories                 : int  1985 1797 1776 1745 1863 1728 1921 2035
1786 1775 ...

n_distinct(daily_activity$Id)

## [1] 33

sum(duplicated(daily_activity))

## [1] 0

sum(is.na(daily_activity))

## [1] 0
```

- Dataset has 940 observations and 15 variable.

- There are no missing values.
- There are 33 unique IDs.
- ActiviyDate is wrongly as character and the date format is wrong

Converting date format and data type

```
daily_activity$ActivityDate <- format(as.Date(daily_activity$ActivityDate,
                                              format = "%m/%d/%Y"), "%Y-%m-
%d")
```

Dropping some variables

```
daily_activity <- daily_activity %>%
  select(-TrackerDistance, -LoggedActivitiesDistance) %>%
  subset()
```

To get the total active minute I will sum up the active minutes aside from the sedentary minutes.

```
daily_activity$total_active_minutes <-
rowSums(cbind(daily_activity$VeryActiveMinutes,

daily_activity$FairlyActiveMinutes,

daily_activity$LightlyActiveMinutes))
```

Renaming the column names to fit the naming convention

```
daily_activity <- daily_activity %>%
  rename(Date = ActivityDate, total_steps = TotalSteps, total_distance =
TotalDistance,
         very_active_distance = VeryActiveDistance,
         moderately_active_distance = ModeratelyActiveDistance,
light_active_distance =
           LightActiveDistance,
         sedentary_active_distance = SedentaryActiveDistance,
very_active_minutes =
           VeryActiveMinutes,
         fairly_active_minutes = FairlyActiveMinutes, sedentary_minutes =
SedentaryMinutes,
         light_active_minutes = LightlyActiveMinutes, calories = Calories)
```

I found some cells with “0” values, so I will omit these to prevent misleading results.

```
daily_activity <- daily_activity %>%
  filter(total_steps !=0, total_distance !=0, total_distance !=0)
```

Changing total active minutes to hours and sedentary minutes to hours

```
daily_activity <- daily_activity %>%
  mutate(total_active_hours = total_active_minutes / 60,
         sedentary_hours = sedentary_minutes / 60 )
```

Adding the days of the week, for the visualization

```
daily_activity <- daily_activity %>%
  mutate(days = strftime(daily_activity$Date, "%A"))
```

Data Manipulation (daily_sleep)

Viewing dataset

```
str(daily_sleep)
```

```
## 'data.frame':    413 obs. of  5 variables:
##  $ Id                : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : chr  "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00
AM" "4/15/2016 12:00:00 AM" "4/16/2016 12:00:00 AM" ...
##  $ TotalSleepRecords : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : int  346 407 442 367 712 320 377 364 384 449 ...
```

```
n_distinct(daily_sleep$Id)
```

```
## [1] 24
```

```
sum(duplicated(daily_sleep))
```

```
## [1] 3
```

```
sum(is.na(daily_sleep))
```

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

- Dataset has 413 observations and 5 variable.
- There are no missing values.
- There are 24 unique IDs.
- Sleepday is wrongly as character and the date format is wrong.
- They are 3 duplicates

The date and time are in the same column in the daily_sleep dataframe and I will be splitting them with the separate () function and converting the date format and data type.

```
daily_sleep <- daily_sleep %>%
  separate(SleepDay, c("Date", "time"), " ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 413 rows [1, 2,
3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
daily_sleep$Date <- format(as.Date(daily_sleep$Date,
                                   format = "%m/%d/%Y"), "%Y-%m-%d")
```

I will also discard the time column because the timestamp is all midnight which could mean it is a fixed time for the logs.

Dropping the time column

```
daily_sleep <- daily_sleep %>%
  select(-time) %>%
  subset()
```

Removing duplicates

```
daily_sleep <- daily_sleep %>%
  distinct()

sum(duplicated(daily_sleep))

## [1] 0

nrow(daily_sleep)

## [1] 410
```

Renaming the column names to fit the naming convention

```
daily_sleep <- daily_sleep %>%
  rename(total_mins_asleep = TotalMinutesAsleep,
         total_sleep_rec = TotalSleepRecords,
         total_mins_in_bed = TotalTimeInBed)
```

changing minutes to hours in the daily _sleep dataset

```
daily_sleep <- daily_sleep %>%
  mutate(total_hrs_asleep = total_mins_asleep / 60,
         total_hrs_in_bed = total_mins_in_bed / 60)
```

Checking and filtering out for '0'

```
daily_sleep <- daily_sleep %>%
  filter(total_sleep_rec !=0, total_hrs_in_bed !=0, total_hrs_asleep !=0)
```

Statistical Ananlysis

I will be merge the daily_activity and daily_sleep for my analysis.

```
activity_sleep <- merge(daily_activity, daily_sleep, by = c("Id", "Date"))
```

Statistical Summary for activity_sleep dataframe

```
activity_sleep%>%
  select(total_steps, total_distance, very_active_distance,
moderately_active_distance,
  light_active_distance,sedentary_active_distance, very_active_minutes,
fairly_active_minutes, light_active_minutes,
sedentary_minutes,sedentary_hours, total_active_minutes, total_active_hours,
total_sleep_rec, total_mins_asleep, total_hrs_asleep, total_mins_in_bed,
```

```
total_hrs_in_bed, calories) %>%
  summary()

##   total_steps    total_distance   very_active_distance
##  Min.   :   17   Min.   : 0.010   Min.   : 0.000
##  1st Qu.: 5189   1st Qu.: 3.592   1st Qu.: 0.000
##  Median : 8913   Median : 6.270   Median : 0.570
##  Mean   : 8515   Mean   : 6.012   Mean   : 1.446
##  3rd Qu.:11370   3rd Qu.: 8.005   3rd Qu.: 2.360
##  Max.   :22770   Max.   :17.540   Max.   :12.540
##  moderately_active_distance light_active_distance
sedentary_active_distance
##  Min.   :0.0000             Min.   :0.010         Min.   :0.0000000
##  1st Qu.:0.0000             1st Qu.:2.540         1st Qu.:0.0000000
##  Median :0.4200             Median :3.665         Median :0.0000000
##  Mean   :0.7439             Mean   :3.791         Mean   :0.0009268
##  3rd Qu.:1.0375             3rd Qu.:4.918         3rd Qu.:0.0000000
##  Max.   :6.4800             Max.   :9.480         Max.   :0.1100000
##  very_active_minutes fairly_active_minutes light_active_minutes
##  Min.   :  0.00      Min.   :  0.00        Min.   :  2.0
##  1st Qu.:  0.00      1st Qu.:  0.00        1st Qu.:158.0
##  Median :  9.00      Median : 11.00        Median :208.0
##  Mean   : 25.05      Mean   : 17.92        Mean   :216.5
##  3rd Qu.: 38.00      3rd Qu.: 26.75        3rd Qu.:263.0
##  Max.   :210.00      Max.   :143.00        Max.   :518.0
##  sedentary_minutes sedentary_hours total_active_minutes total_active_hours
##  Min.   :   0.0    Min.   : 0.00   Min.   :  2.0        Min.   :0.03333
##  1st Qu.: 631.2    1st Qu.:10.52   1st Qu.:206.5        1st Qu.:3.44167
##  Median : 717.0    Median :11.95   Median :263.5        Median :4.39167
##  Mean   : 712.1    Mean   :11.87   Mean   :259.5        Mean   :4.32520
##  3rd Qu.: 782.8    3rd Qu.:13.05   3rd Qu.:315.5        3rd Qu.:5.25833
##  Max.   :1265.0    Max.   :21.08   Max.   :540.0        Max.   :9.00000
##  total_sleep_rec total_mins_asleep total_hrs_asleep  total_mins_in_bed
##  Min.   :1.00    Min.   : 58.0     Min.   : 0.9667   Min.   : 61.0
##  1st Qu.:1.00    1st Qu.:361.0     1st Qu.: 6.0167   1st Qu.:403.8
##  Median :1.00    Median :432.5     Median : 7.2083   Median :463.0
##  Mean   :1.12    Mean   :419.2     Mean   : 6.9862   Mean   :458.5
##  3rd Qu.:1.00    3rd Qu.:490.0     3rd Qu.: 8.1667   3rd Qu.:526.0
##  Max.   :3.00    Max.   :796.0     Max.   :13.2667   Max.   :961.0
##  total_hrs_in_bed    calories
##  Min.   : 1.017   Min.   : 257
##  1st Qu.: 6.729   1st Qu.:1841
##  Median : 7.717   Median :2207
##  Mean   : 7.641   Mean   :2389
##  3rd Qu.: 8.767   3rd Qu.:2920
##  Max.   :16.017   Max.   :4900
```

Statistical Summary Interpretation

- Average steps logged by the users is 8,329 which means they were somewhat active and according to a 2011 study found that healthy adults can take anywhere between

approximately 4,000 and 18,000 steps/day and that 10,000 steps/day is a reasonable target for healthy adults. **Source:** https://www.healthline.com/health/how-many-steps-a-day

- The average total distance covered is 5.986km daily.
- Much of the distance covered was light active (3.643 km).
- Most of the active time covered was light active by 210.3 mins or 3.5hrs.
- The average time users spent being sedentary is 712.1 mins or 11.87 hrs which is more than the average total active time (259.5 mins or 4.3 hrs), which means the users were largely inactive or they spent less time exercising.
- The users had 6.9 hours of sleep (419.2 mins), and the average total hrs in bed is 7.6 hrs (458.5 mins), the data shows that users spend an average of 39.3 minutes in bed before drifting off to sleep. "Normal sleep for adults means that you fall asleep within 10 to 20 minutes and get about 7–8 hours a night" which means users have an adequate amount of sleep. **Source:** https://www.healthline.com/health/healthy-sleep/how-long-does-it-take-to-fall-asleep
- Average amount of calories burned is 2389 calories with a minimum of 257 calories and a maximum of 4900 calories.

I will look for a correlation between columns using the Pearson coefficient correlation ranging from +1 to -1.

```
#a. total_steps and total_distance
cor.test(activity_sleep$total_steps, activity_sleep$total_distance, method =
"pearson" )

##
##  Pearson's product-moment correlation
##
## data:  activity_sleep$total_steps and activity_sleep$total_distance
## t = 104.29, df = 408, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9778840 0.9849518
## sample estimates:
##       cor
## 0.9817539

# b. total_steps and calories
cor.test(activity_sleep$total_steps, activity_sleep$calories, method =
"pearson")

##
##  Pearson's product-moment correlation
##
## data:  activity_sleep$total_steps and activity_sleep$calories
```

```
## t = 8.9816, df = 408, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3221289 0.4840991
## sample estimates:
##       cor
## 0.4063007
```

```
#c. total_active_hours and total_distance
cor.test(activity_sleep$total_distance, activity_sleep$total_active_hours,
method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  activity_sleep$total_distance and activity_sleep$total_active_hours
## t = 21.101, df = 408, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6725917 0.7656646
## sample estimates:
##       cor
## 0.7223839
```

```
#d. total_active_hours and calories
cor.test(activity_sleep$total_active_hours, activity_sleep$calories, method =
"pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  activity_sleep$total_active_hours and activity_sleep$calories
## t = 8.5546, df = 408, p-value = 2.417e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3046418 0.4691127
## sample estimates:
##       cor
## 0.3899832
```

```
#e. sedentary_distance and calories
cor.test(activity_sleep$sedentary_active_distance, activity_sleep$calories,
method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  activity_sleep$sedentary_active_distance and
activity_sleep$calories
## t = 0.57508, df = 408, p-value = 0.5656
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.06857722  0.12496198
## sample estimates:
##       cor
## 0.0284591

#f. sedentary_hours and calories
cor.test(activity_sleep$sedentary_hours, activity_sleep$calories, method =
"pearson")

##
##  Pearson's product-moment correlation
##
## data:  activity_sleep$sedentary_hours and activity_sleep$calories
## t = 2.0025, df = 408, p-value = 0.04589
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.001825848 0.193652746
## sample estimates:
##       cor
## 0.09865571

#g. total_distance and sedentary_hours
cor.test(activity_sleep$total_steps, activity_sleep$sedentary_hours, method =
"pearson")

##
##  Pearson's product-moment correlation
##
## data:  activity_sleep$total_steps and activity_sleep$sedentary_hours
## t = -2.6491, df = 408, p-value = 0.008384
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.22406152 -0.03361196
## sample estimates:
##       cor
## -0.130036

#h. total_hrs_asleep and total_hrs_in_bed
cor.test(activity_sleep$total_hrs_asleep, activity_sleep$total_hrs_in_bed,
method = 'pearson')

##
##  Pearson's product-moment correlation
##
## data:  activity_sleep$total_hrs_asleep and activity_sleep$total_hrs_in_bed
## t = 51.28, df = 408, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9161262 0.9423551
## sample estimates:
```

```
##       cor
## 0.9304224

#i. total_hrs_asleep and calories
cor.test(activity_sleep$total_hrs_asleep, activity_sleep$calories, method =
'pearson')

##
##  Pearson's product-moment correlation
##
## data:  activity_sleep$total_hrs_asleep and activity_sleep$calories
## t = -0.64061, df = 408, p-value = 0.5221
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.12815287  0.06534893
## sample estimates:
##         cor
## -0.03169899
```

a. Total steps and total correlation is 0.9817539. This means that they have a very strong positive correlation showing that they increase in the same direction. This implies that total distance is derived from steps taken.

b. Total Steps and calories correlation is 0.4063007. This means that have a positive moderate relationship.

c. Total distance and total active correlation is 0.7223839. This means that they have a strong positive correlation showing that they increase in almost the same direction.

d. Total active hours and calories correlation is 0.3899832. This means that they have a positive but weak relationship.

e. Sedentary distance and calories correlation is 0.0284591. This means that they do not correlate.

f. Sedentary hours and calories correlation is 0.0284591. This means that they do not correlate.

g. Total active distance and sedentary hours correlation is -0.130036. This means that they have a negative correlation. The variables do not move in the same direction: one increases as the other decreases. Most likely as more distance is covered increases, sedentary hours decrease.

h. Total hours asleep and total hours in bed correlation is 0.9304224. This means that they have a very strong positive correlation showing that they increase in the same direction. This implies that spending some time in bed helps the users fall asleep.

i. Total hours asleep and calories correlation is -0.03169899. This means that the variables do not correlate.

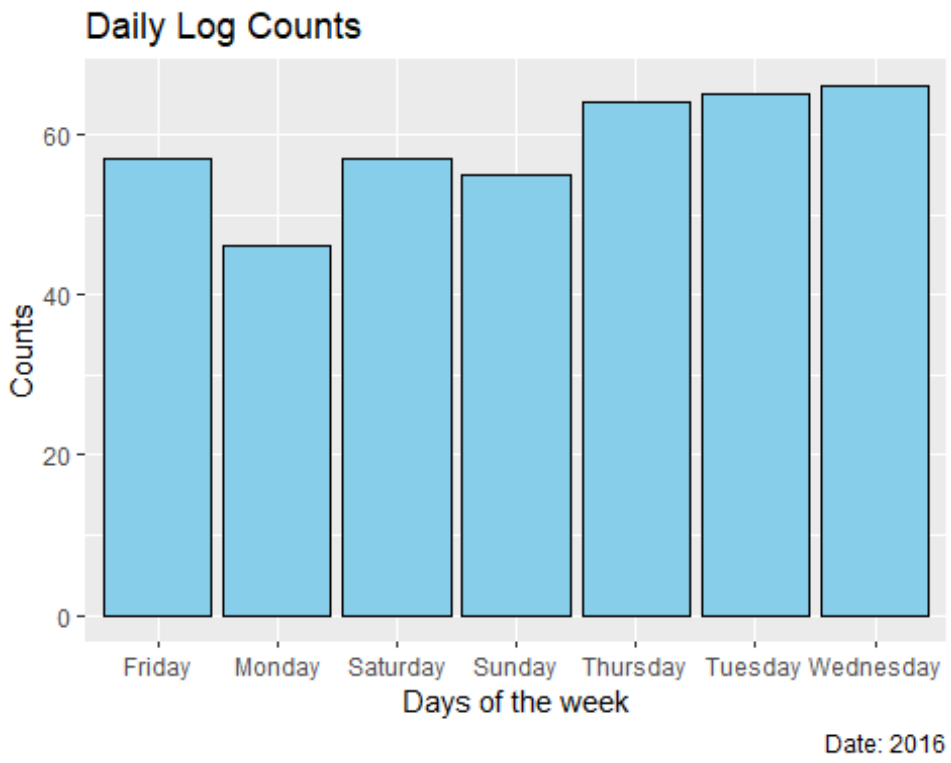
5. SHARE

Data visualization

In this step, I will create visualizations to communicate the findings from my analysis

Daily log Frequency

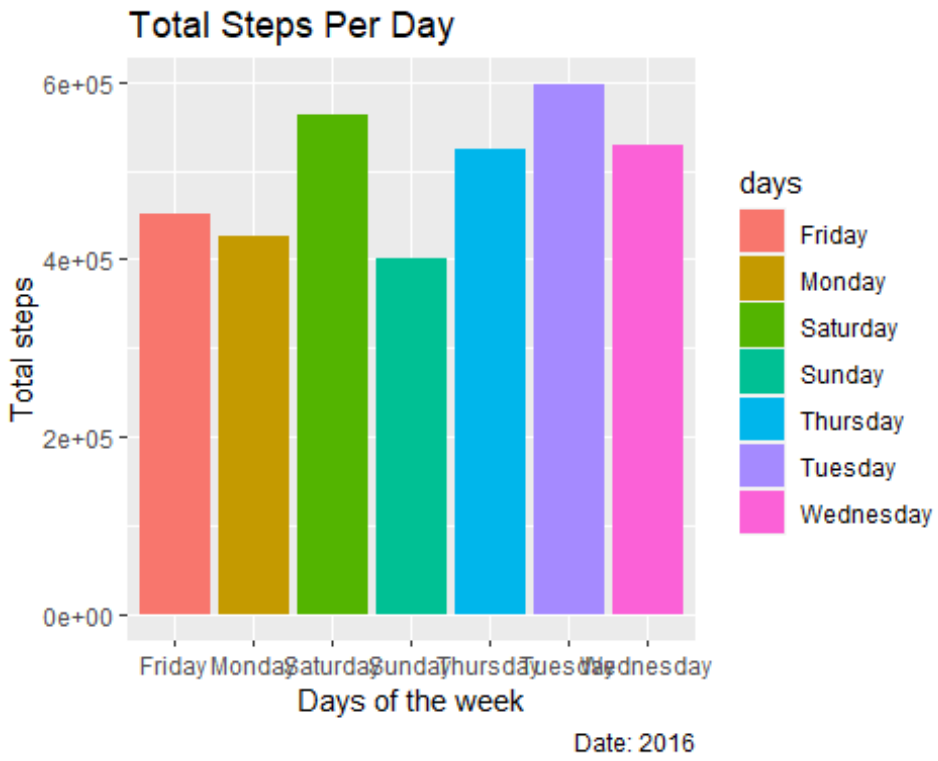
```
ggplot(activity_sleep, aes(x=days))+
  geom_histogram(stat = "Count", color="black", fill="skyblue")+
  labs(title = "Daily Log Counts", x="Days of the week", y="Counts",caption =
"Date: 2016")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



The highest log days were on Tuesdays, Wednesdays, and Thursdays but decreased during the weekend; from Friday to Sunday, and Monday.

Total steps per day

```
ggplot(data = activity_sleep, aes(x=days, y = total_steps, fill = days)) +
  geom_bar(stat = 'identity') +
  labs(title = "Total Steps Per Day", x= "Days of the week", y= "Total
steps",
       caption = "Date: 2016")
```

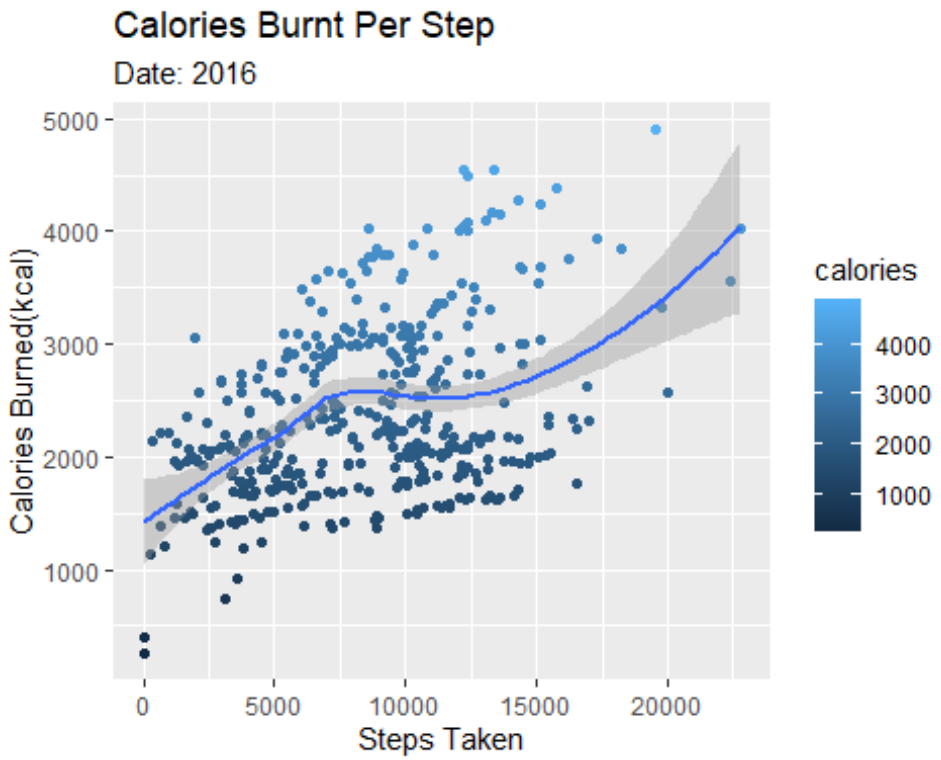


From the graph bar above, we can see that Tuesday, Wednesday, and Thursday are the most active days for users. And the least active day for the user is Sundays.

Calories burnt per step

```
ggplot(data = activity_sleep, mapping = aes(x=total_steps, y=calories, color
= calories )) +
  geom_point() +
  geom_smooth(method = "loess") +
  labs(title = "Calories Burnt Per Step", x = "Steps Taken",
       y = "Calories Burned(kcal)", subtitle="Date: 2016")
```

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



From the graph above,

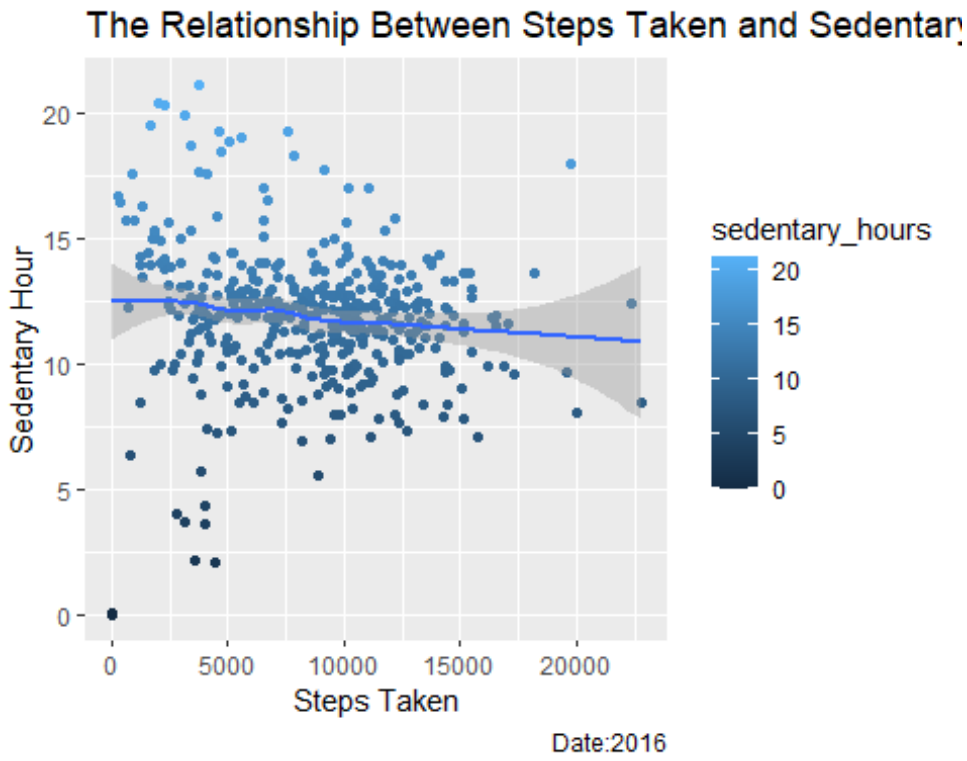
- It shows that they have a positive correlation.
- We can also see that the larger amount of steps taken, the more calories are burned.

Noted a few outliers:

- Zero steps with zero to minimal calories burned.
- 1 observation of < 20,000 steps with < 5,000 calories burned.
- The outliers could be due to natural variation of data, change in user usage, or errors in data collection (miscalculations, data contamination, or human error).

The relationship between steps taken and sedentary hours

```
ggplot(data = activity_sleep, mapping = aes(x=total_steps, y=sedentary_hours,
                                            color = sedentary_hours))+
  geom_point()+
  geom_smooth(method = 'loess')+
  labs(title = "The Relationship Between Steps Taken and Sedentary Time",
caption = "Date:2016", x = "Steps Taken", y = "Sedentary Hour")

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

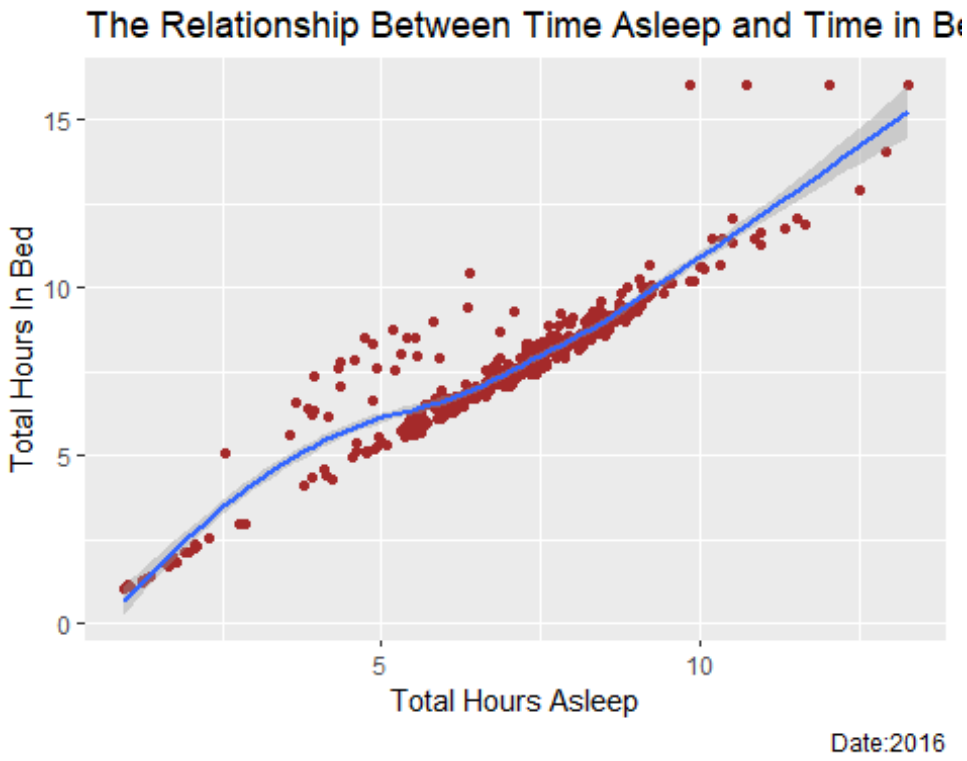


The graph above shows a negative correlation between total steps and sedentary hours, the higher the sedentary hours the lower the steps taken.

The relationship between total hours asleep and total hours in bed.

```
ggplot(data = activity_sleep)+
  geom_point(mapping = aes(x=total_hrs_asleep, y= total_hrs_in_bed), color=
'brown')+
  geom_smooth(mapping = aes(x=total_hrs_asleep, y= total_hrs_in_bed))+
  labs(title = "The Relationship Between Time Asleep and Time in Bed",
caption = "Date:2016",
       x = "Total Hours Asleep", y = "Total Hours In Bed")
```

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



From the graph above, there is a positive correlation between the time asleep and the amount of time spent in bed, This implies that spending some time in bed helps the users fall asleep.

6. ACT

Key Findings

- The users logged their activities more during mid-week days (Tuesday, Wednesday, and Thursday), more than on the weekends.
- Most of the users spend the time being inactive.
- Majority of users use the app to track sedentary activities more than they track tracking their active period.
- Most of the users did not take enough steps to be considered active daily.
- Most of all the users make use of their FitBit gadgets regularly while sleeping.

Recommendation

- Since most users are only active on certain days and times. They can implement features that could help the users set a reminder for a specific time they would like to exercise daily.

- Bellabeat could implement a push notification that can be used to remind the users to exercise on the weekends.
- They can add a dairy feature where the users can document their daily activities.
- They can create a menstrual tracking algorithm to help predict the next menstrual-cycle period and send a reminder to the users.
- Upload articles to educate the user on different health habits and lifestyles, like daily calorie intake calculations, or food suggestions.