BELLA BEAT CASE STUDY

Wonder Unaji

INTRODUCTION

Bellabeat is a high-tech company that manufactures health-focused smart products.They offer different smart devices that collect data on activity, sleep, stress, and reproductive health to empower women with knowledge about their own health and habits.

The main focus of this case is to analyze smart devices fitness data and determine how it could help unlock new growth opportunities for Bellabeat. We will focus on one of Bellabeat's 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.

2. ASK

2.1 Business Task

Identify trends in how consumers use non-Bellabeat smart devices to apply insights into Bellabeat's marketing strategy.

Stakeholders

- Urška Sršen - Bellabeat cofounder and Chief Creative Officer
- Sando Mur - Bellabeat cofounder and key member of Bellabeat executive team
- Bellabeat Marketing Analytics team

3. PREPARE

3.1 Dataset used:

The data source used for our case study is FitBit Fitness Tracker Data. This dataset is stored in Kaggle and was made available through Mobius.

3.2 ANSWERING THE QUESTION OF BIAS AND CREDIBILITY (ROCCC)

- RELIABILITY

Data collected from only 30 Fitbit Users is too small a sample set hence answering the question of RELIABLITY. We could encounter problems of sampling bias

- ORIGINAL

Verifying the metadata of our dataset we can confirm it is open-source. By releasing all worldwide rights to the work under copyright law, including any connected and adjacent rights, to the degree permitted by law, the owner has devoted the work to the public domain. Without obtaining consent, you are free to distribute, perform, copy, alter, and even use the work for profit.

- COMPREHENSIVE

Available to us are 18 CSV documents. Each document represents different quantitative data tracked by Fitbit. The data is considered long since each row is one time point per subject, so each subject will have data in multiple rows.Every user has a unique ID and different rows since data is tracked by day and time.

- CURRENT

These datasets were generated by respondents to a distributed survey via Amazon Mechanical Turk between 03.12.2016-05.12.2016 and this analysis is being carried out in 2023 approximately 7 years after so the result might not be relevant at the time.

- CITED

As regards CITED all we know about the dataset is it was provided by respondents to a distributed survey via Amazon Mechanical Turk. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. Variation between output represents use of different types of Fitbit trackers and individual tracking behaviors / preferences.

4. PROCESS

I decided to use R and Rstudio for the process,cleaning and analysis of this case study because of the amount of data and it's inbuilt reproducible visualisation features to explore the data.

4.1 Installing packages and Loading Libraries

```r
# Setting up the environment (installing packages)
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
install.packages("readr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
install.packages("dplyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
install.packages("tidyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
install.packages("here")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
install.packages("skimr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
install.packages("janitor")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
install.packages("ggpubr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
install.packages("ggrepel")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
#loading the packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(readr)
library(dplyr)
library(tidyr)
library(here)
```

```
## here() starts at /cloud/project
```

```r
library(skimr)
library(janitor)
```

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

```r
library(lubridate)
```

4.2 Importing the Dataset

Of the 18 csv files we already have I decided to work with 8 to streamline my analysis.

```r
daily_activity<-read.csv("dailyActivity_merged.csv")
daily_calories<-read.csv("dailyCalories_merged.csv")
daily_intensities<-read.csv("dailyIntensities_merged.csv")
daily_steps<-read.csv("dailySteps_merged.csv")
daily_sleep<-read.csv("sleepDay_merged.csv")
weight_log<-read.csv("weightLogInfo_merged.csv")
minute_METs<-read.csv("minuteMETsNarrow_merged.csv")
hourly_steps<-read.csv("hourlySteps_merged.csv")
```

You could use the view(dataset) function to view any of the dataframes.

4.3 Formatting Date and Time Column

I decided to do this early before viewing the datasets to ensure consistency in the different columns using the POSI.xct function.

If you open the datasets in googlesheets or excel you'd notice the inconsistencies in the date columns in the datasets.

```
daily_activity$ActivityDate = as.POSIXct(daily_activity$ActivityDate, format = "%m/%d/%Y",tz = Sys.time
daily_calories$ActivityDay = as.POSIXct(daily_calories$ActivityDay, format = "%m/%d/%Y",tz = Sys.timezo
daily_intensities$ActivityDay = as.POSIXct(daily_calories$ActivityDay, format = "%m/%d/%Y",tz = Sys.tim
daily_steps$ActivityDay = as.POSIXct(daily_calories$ActivityDay, format = "%m/%d/%Y",tz = Sys.timezone(
daily_sleep$SleepDay = as.POSIXct(daily_sleep$SleepDay, format = "%m/%d/%Y",tz = Sys.timezone())
weight_log$Date = as.POSIXct(weight_log$Date, format = "%m/%d/%Y",tz = Sys.timezone())
minute_METs$ActivityMinute = as.POSIXct(minute_METs$ActivityMinute,format = "%m/%d/%Y",tz = Sys.timezon
hourly_steps$ActivityHour = as.POSIXct(hourly_steps$ActivityHour, format = "%m/%d/%Y %I:%M:%S %p", tz =
```

Quick Verification if it has been formatted

```
class(daily_activity$ActivityDate)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(daily_calories$ActivityDay)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(daily_intensities$ActivityDay)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(daily_steps$ActivityDay)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(daily_sleep$SleepDay)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(weight_log$Date)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(minute_METs$ActivityMinute)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(hourly_steps$ActivityHour)
```

```
## [1] "POSIXct" "POSIXt"
```

4.4 Preview of the Dataframes

The head() function is being used to ensure the dataframes are imported correctly.

The str() function to show the structure of the dataframes and any interesting features of the datframe.

The colnames() function to retrieve the column names of the dataframes.

```
head(daily_activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366   2016-04-12      13162          8.50            8.50
## 2 1503960366   2016-04-13      10735          6.97            6.97
## 3 1503960366   2016-04-14      10460          6.74            6.74
## 4 1503960366   2016-04-15       9762          6.28            6.28
## 5 1503960366   2016-04-16      12669          8.16            8.16
## 6 1503960366   2016-04-17       9705          6.48            6.48
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               1.88                     0.55
## 2                        0               1.57                     0.69
## 3                        0               2.44                     0.40
## 4                        0               2.14                     1.26
## 5                        0               2.71                     0.41
## 6                        0               3.19                     0.78
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                3.91                       0                30
## 4                2.83                       0                29
## 5                5.04                       0                36
## 6                2.51                       0                38
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  13                  328              728     1985
## 2                  19                  217              776     1797
## 3                  11                  181             1218     1776
## 4                  34                  209              726     1745
## 5                  10                  221              773     1863
## 6                  20                  164              539     1728
```

```r
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            : POSIXct, format: "2016-04-12" "2016-04-13" ...
##  $ 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 ...
```

```r
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"
```

```r
head(daily_calories)
```

```
##           Id ActivityDay Calories
## 1 1503960366  2016-04-12     1985
## 2 1503960366  2016-04-13     1797
## 3 1503960366  2016-04-14     1776
## 4 1503960366  2016-04-15     1745
## 5 1503960366  2016-04-16     1863
## 6 1503960366  2016-04-17     1728
```

```r
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: POSIXct, format: "2016-04-12" "2016-04-13" ...
##  $ Calories   : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```r
colnames(daily_calories)
```

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

```r
head(daily_intensities)
```

```
##           Id ActivityDay SedentaryMinutes LightlyActiveMinutes
## 1 1503960366  2016-04-12              728                  328
## 2 1503960366  2016-04-13              776                  217
## 3 1503960366  2016-04-14             1218                  181
## 4 1503960366  2016-04-15              726                  209
## 5 1503960366  2016-04-16              773                  221
## 6 1503960366  2016-04-17              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
```

```r
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             : POSIXct, format: "2016-04-12" "2016-04-13" ...
##  $ 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 ...
```

```
colnames(daily_intensities)
```

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

```
head(daily_steps)
```

```
##           Id ActivityDay StepTotal
## 1 1503960366  2016-04-12     13162
## 2 1503960366  2016-04-13     10735
## 3 1503960366  2016-04-14     10460
## 4 1503960366  2016-04-15      9762
## 5 1503960366  2016-04-16     12669
## 6 1503960366  2016-04-17      9705
```

```
str(daily_steps)
```

```
## '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: POSIXct, format: "2016-04-12" "2016-04-13" ...
##  $ StepTotal  : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
```

```
colnames(daily_steps)
```

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

```
head(daily_sleep)
```

```
##           Id   SleepDay TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
## 1 1503960366 2016-04-12                 1                327            346
## 2 1503960366 2016-04-13                 2                384            407
## 3 1503960366 2016-04-15                 1                412            442
## 4 1503960366 2016-04-16                 2                340            367
## 5 1503960366 2016-04-17                 1                700            712
## 6 1503960366 2016-04-19                 1                304            320
```

```
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          : POSIXct, format: "2016-04-12" "2016-04-13" ...
##  $ 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 ...
```

```
colnames(daily_sleep)
```

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

```
head(weight_log)
```

```
##           Id       Date WeightKg WeightPounds Fat   BMI IsManualReport
## 1 1503960366 2016-05-02     52.6     115.9631  22 22.65           True
## 2 1503960366 2016-05-03     52.6     115.9631  NA 22.65           True
## 3 1927972279 2016-04-13    133.5     294.3171  NA 47.54          False
## 4 2873212765 2016-04-21     56.7     125.0021  NA 21.45           True
## 5 2873212765 2016-05-12     57.3     126.3249  NA 21.69           True
## 6 4319703577 2016-04-17     72.4     159.6147  25 27.45           True
##          LogId
## 1 1.462234e+12
## 2 1.462320e+12
## 3 1.460510e+12
## 4 1.461283e+12
## 5 1.463098e+12
## 6 1.460938e+12
```

```
str(weight_log)
```

```
## '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          : POSIXct, format: "2016-05-02" "2016-05-03" ...
##  $ 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 ...
```

```
colnames(weight_log)
```

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

```
head(minute_METs)
```

```
##           Id ActivityMinute METs
## 1 1503960366     2016-04-12   10
## 2 1503960366     2016-04-12   10
## 3 1503960366     2016-04-12   10
## 4 1503960366     2016-04-12   10
## 5 1503960366     2016-04-12   10
## 6 1503960366     2016-04-12   12
```

```
str(minute_METs)
```

```
## 'data.frame':    1325580 obs. of  3 variables:
##  $ Id            : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityMinute: POSIXct, format: "2016-04-12" "2016-04-12" ...
##  $ METs          : int  10 10 10 10 10 12 12 12 12 12 ...
```

```
colnames(minute_METs)
```

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

```
head(hourly_steps)
```

```
##           Id        ActivityHour StepTotal
## 1 1503960366 2016-04-12 00:00:00       373
## 2 1503960366 2016-04-12 01:00:00       160
## 3 1503960366 2016-04-12 02:00:00       151
## 4 1503960366 2016-04-12 03:00:00         0
## 5 1503960366 2016-04-12 04:00:00         0
## 6 1503960366 2016-04-12 05:00:00         0
```

```
str(hourly_steps)
```

```
## 'data.frame':    22099 obs. of  3 variables:
##  $ Id          : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour: POSIXct, format: "2016-04-12 00:00:00" "2016-04-12 01:00:00" ...
##  $ StepTotal   : int  373 160 151 0 0 0 0 0 250 1864 ...
```

```
colnames(hourly_steps)
```

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

4.5 Verifying dataframes

You'd notice that in every data frame we selected **(8)** there's a particular column common to all and that's the 'Id' which could be very useful if/when we want to join/merge our dataframes together into one dataframe just like in SQL.

You'd also notice when previewing the dataframes that dataframes such as **daily_calories**, **daily_intensities** and **daily_steps** have columns also in the **daily_activity** merged dataframe implying they've been merged.

Since they all have a common **Id** we can verify using an SQL package(sqldf) to check whether we could use **daily_activity** dataframe in place of the other three(3) dataframes to avoid making decisions based on assumptions.

To do this the number of observations must match the number of observations for each ID number.

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

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
library(sqldf)
```

```
## Loading required package: gsubfn

## Loading required package: proto

## Warning in fun(libname, pkgname): couldn't connect to display ":0"

## Loading required package: RSQLite
```

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

```
head(daily_activity_2)
```

```
##           Id ActivityDate Calories
## 1 1503960366   2016-04-12     1985
## 2 1503960366   2016-04-13     1797
## 3 1503960366   2016-04-14     1776
## 4 1503960366   2016-04-15     1745
## 5 1503960366   2016-04-16     1863
## 6 1503960366   2016-04-17     1728
```

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

```
head(sql_check)
```

```
##           Id ActivityDate Calories
## 1 1503960366   2016-04-12     1985
## 2 1503960366   2016-04-13     1797
## 3 1503960366   2016-04-14     1776
## 4 1503960366   2016-04-15     1745
## 5 1503960366   2016-04-16     1863
## 6 1503960366   2016-04-17     1728
```

```
nrow(daily_activity_2)
```

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

```
nrow(sql_check)
```

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

```
daily_activity_3 <- daily_activity %>%
    select(Id, ActivityDate, SedentaryMinutes, LightlyActiveMinutes, FairlyActiveMinutes, VeryActiveMinu
```

```
head(daily_activity_3)
```

```
##           Id ActivityDate SedentaryMinutes LightlyActiveMinutes
## 1 1503960366   2016-04-12              728                  328
## 2 1503960366   2016-04-13              776                  217
## 3 1503960366   2016-04-14             1218                  181
## 4 1503960366   2016-04-15              726                  209
## 5 1503960366   2016-04-16              773                  221
## 6 1503960366   2016-04-17              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
```

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

```
head(sql_check_2)
```

```
##           Id ActivityDate SedentaryMinutes LightlyActiveMinutes
## 1 1503960366   2016-04-12              728                  328
## 2 1503960366   2016-04-13              776                  217
## 3 1503960366   2016-04-14             1218                  181
## 4 1503960366   2016-04-15              726                  209
## 5 1503960366   2016-04-16              773                  221
## 6 1503960366   2016-04-17              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(daily_activity_3)
```

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

```
nrow(sql_check_2)
```

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

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

```
head(daily_activity_4)
```

```
##           Id ActivityDate TotalSteps
## 1 1503960366   2016-04-12      13162
## 2 1503960366   2016-04-13      10735
## 3 1503960366   2016-04-14      10460
## 4 1503960366   2016-04-15       9762
## 5 1503960366   2016-04-16      12669
## 6 1503960366   2016-04-17       9705
```

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

```
head(sql_check_3)
```

```
##           Id ActivityDate TotalSteps
## 1 1503960366   2016-04-12      13162
## 2 1503960366   2016-04-13      10735
## 3 1503960366   2016-04-14      10460
## 4 1503960366   2016-04-15       9762
## 5 1503960366   2016-04-16      12669
## 6 1503960366   2016-04-17       9705
```

```
nrow(daily_activity_4)
```

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

```
nrow(sql_check_3)
```

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

The outputs of the temporary data frames' head() function match the outputs of the original data frames' head() function.

The outputs of the SQL data frames' head() function match the outputs of the temporary data frames' head() function. The number of observations in each SQL data frame is 940.

In conclusion, **daily_activity** contains the data for the **daily_calories**, **daily_intensities**, and **daily_steps** data frames. Hence, these three data frames will be excluded from the study.

4.5.1 Verifying number of participants

```r
n_unique(daily_activity$Id)
```

```
## [1] 33
```

```r
n_unique(weight_log$Id)
```

```
## [1] 8
```

```r
n_unique(daily_sleep$Id)
```

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

```r
n_unique(hourly_steps$Id)
```

```
## [1] 33
```

The weight log data frame contain a very low number of participants based on the n_unique() outputs. Thus, reliable recommendations and conclusions cannot be made solely from these data frame.Hence we'd drop this dataset.

###. 4.5.2 Checking for Duplicates

We'd now check for duplicates

```r
sum(duplicated(daily_activity))
```

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

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

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

```r
sum(duplicated(hourly_steps))
```

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

###. 4.5.3 Removing Duplicates

Although we already know that the daily_sleep dataframe contains duplicates, we'd run for all of them

```r
#daily_activity
daily_activity<-daily_activity %>%
  distinct() %>%
  drop_na()

#daily_sleep
daily_sleep<-daily_sleep %>%
  distinct() %>%
  drop_na()

#hourly_steps
hourly_steps<-hourly_steps %>%
  distinct() %>%
  drop_na()
```

We can verify to ensure the duplicates have been removed

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

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

4.6 Merging Datasets

To merge daily_activity and daily_sleep together, we have to ensure the primary keys to be used such as ID and Date are uniform. Hence a bit of renaming.

```
## renaming ActivityDate in daily_activity
daily_activity <- daily_activity %>%
  rename(date = ActivityDate) %>%
  mutate(date)
```

```
## renaming SleepDay in daily_sleep
daily_sleep <- daily_sleep %>%
  rename(date = SleepDay) %>%
  mutate(date)
```

Now merging the datasets

```
daily_activity_sleep<- merge(daily_activity, daily_sleep, by=c ("Id","date"))
glimpse(daily_activity_sleep)
```

```
## Rows: 410
## Columns: 18
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ date                     <dttm> 2016-04-12, 2016-04-13, 2016-04-15, 2016-04-~
## $ TotalSteps               <int> 13162, 10735, 9762, 12669, 9705, 15506, 10544~
## $ TotalDistance            <dbl> 8.50, 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.3~
## $ TrackerDistance          <dbl> 8.50, 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.3~
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance       <dbl> 1.88, 1.57, 2.14, 2.71, 3.19, 3.53, 1.96, 1.3~
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 1.26, 0.41, 0.78, 1.32, 0.48, 0.3~
## $ LightActiveDistance      <dbl> 6.06, 4.71, 2.83, 5.04, 2.51, 5.03, 4.24, 4.6~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes        <int> 25, 21, 29, 36, 38, 50, 28, 19, 41, 39, 73, 3~
## $ FairlyActiveMinutes      <int> 13, 19, 34, 10, 20, 31, 12, 8, 21, 5, 14, 23,~
## $ LightlyActiveMinutes     <int> 328, 217, 209, 221, 164, 264, 205, 211, 262, ~
## $ SedentaryMinutes         <int> 728, 776, 726, 773, 539, 775, 818, 838, 732, ~
## $ Calories                 <int> 1985, 1797, 1745, 1863, 1728, 2035, 1786, 177~
## $ TotalSleepRecords        <int> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ TotalMinutesAsleep       <int> 327, 384, 412, 340, 700, 304, 360, 325, 361, ~
## $ TotalTimeInBed           <int> 346, 407, 442, 367, 712, 320, 377, 364, 384, ~
```

5. ANALYZE AND SHARE PHASE

We analyze the trends of FitBit users in order to help us make informed BellaBeat marketing strategies.

We want to ascertain the type of users with the data we have because we don't have any demographic information from our sample. We can categorize people based on their daily number of steps using this information provided by CDC.

- Sedentary is less than 5,000 steps per day
- Low active is 5,000 to 7,499 steps per day

- Somewhat active is 7,500 to 9,999 steps per day
- Active is more than 10,000 steps per day
- Highly active is more than 12,500

We calculate the daily average steps by the user

```
daily_average <- daily_activity_sleep %>%
  group_by(Id) %>%
  summarise(mean_daily_steps = mean(TotalSteps), mean_daily_calories = mean(Calories), mean_daily_sleep

head(daily_average)
```

```
## # A tibble: 6 x 4
##           Id mean_daily_steps mean_daily_calories mean_daily_sleep
##        <dbl>            <dbl>               <dbl>            <dbl>
## 1 1503960366           12406.               1872.             360.
## 2 1644430081            7968.               2978.             294
## 3 1844505072            3477                1676.             652
## 4 1927972279            1490                2316.             417
## 5 2026352035            5619.               1541.             506.
## 6 2320127002            5079                1804               61
```

We use the daily_average steps to classify our users

```
user_type <- daily_average %>%
  mutate(user_type = case_when(
    mean_daily_steps < 5000 ~ "sedentary",
    mean_daily_steps >= 5000 & mean_daily_steps < 7499 ~ "lightly active",
    mean_daily_steps >= 7500 & mean_daily_steps < 9999 ~ "fairly active",
    mean_daily_steps >= 10000 ~ "very active"
  ))

head(user_type)
```

```
## # A tibble: 6 x 5
##           Id mean_daily_steps mean_daily_calories mean_daily_sleep user_type
##        <dbl>            <dbl>               <dbl>            <dbl> <chr>
## 1 1503960366           12406.               1872.             360. very active
## 2 1644430081            7968.               2978.             294  fairly active
## 3 1844505072            3477                1676.             652  sedentary
## 4 1927972279            1490                2316.             417  sedentary
## 5 2026352035            5619.               1541.             506. lightly acti~
## 6 2320127002            5079                1804               61  lightly acti~
```

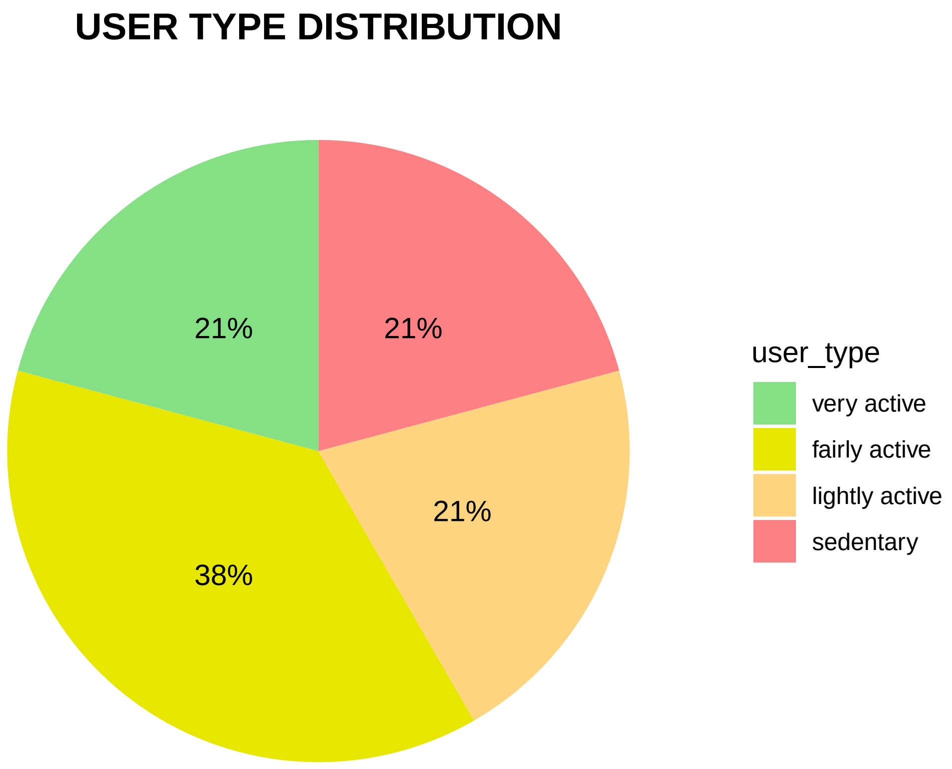
We will create a data frame with the percentage of each user type now that we have a new column with the user type to better visualise them on a graph.

```
user_type_percent <- user_type %>%
  group_by(user_type) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(user_type) %>%
  summarise(total_percent = total / totals) %>%
  mutate(labels = scales::percent(total_percent))

user_type_percent$user_type <- factor(user_type_percent$user_type , levels = c("very active", "fairly ac
head(user_type_percent)
```

```
## # A tibble: 4 x 3
##   user_type      total_percent labels
##   <fct>                  <dbl> <chr>
## 1 fairly active          0.375 38%
## 2 lightly active         0.208 21%
## 3 sedentary              0.208 21%
## 4 very active            0.208 21%
```

We can infer from this table that the all the user type wear their smart devices based on user activity.

```
user_type_percent %>%
  ggplot(aes(x="",y=total_percent, fill=user_type)) +
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0)+
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5, size=14, face = "bold")) +
  scale_fill_manual(values = c("#85e085","#e6e600", "#ffd480", "#fd8083")) +
  geom_text(aes(label = labels),
            position = position_stack(vjust = 0.5))+
  labs(title="USER TYPE DISTRIBUTION")
```



5.1 Steps and minutes asleep per weekday

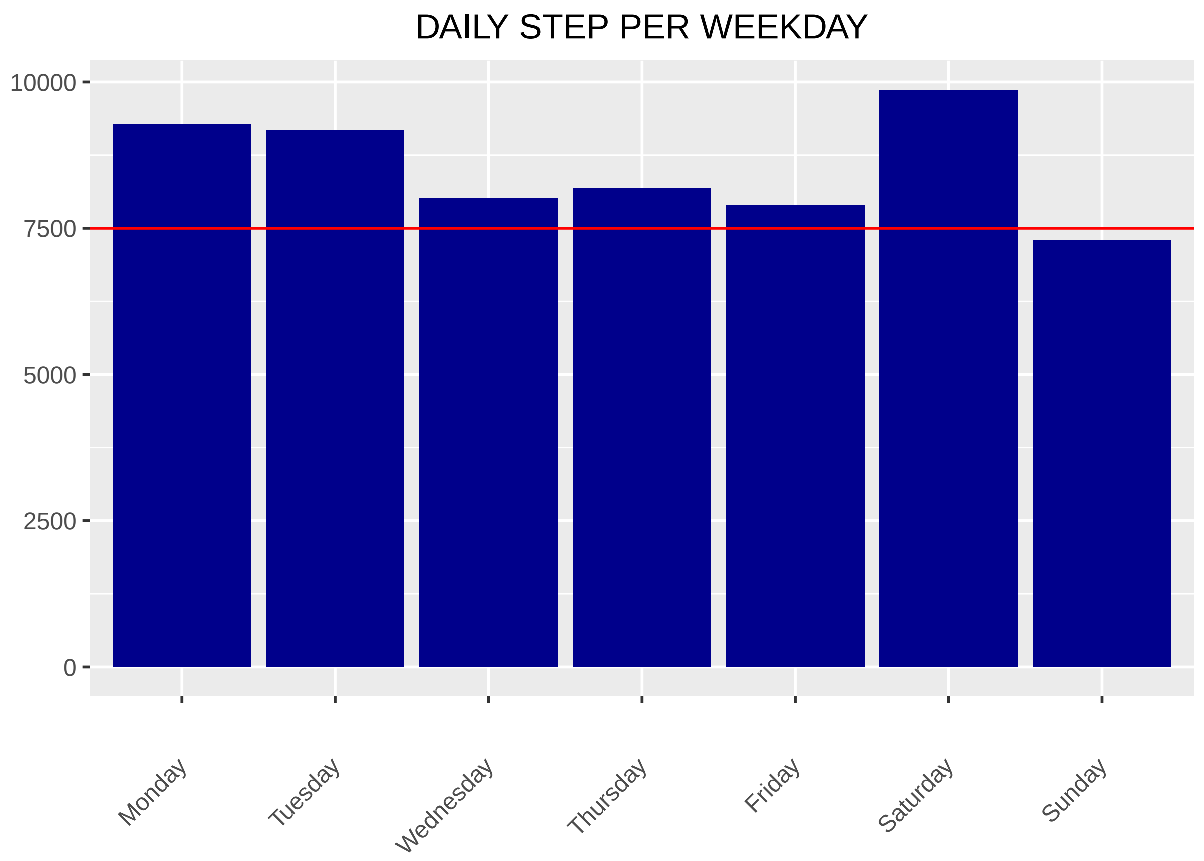
We want to know which days of the week users are the most active and which days of the week users sleep the most. We will also check to see if the users are taking the required number of steps and getting enough sleep.

The weekdays are calculated below depending on our column date. We are also estimating the average number of steps taken and minutes slept per weekday.

```
weekday_steps_sleep <- daily_activity_sleep %>%
  mutate(weekday = weekdays(date))

weekday_steps_sleep$weekday <-ordered(weekday_steps_sleep$weekday, levels=c("Monday", "Tuesday", "Wedne
"Friday", "Saturday", "Sunday"))



weekday_steps_sleep <-weekday_steps_sleep%>%
  group_by(weekday) %>%
  summarize (daily_steps = mean(TotalSteps), daily_sleep = mean(TotalMinutesAsleep))

head(weekday_steps_sleep)
```

```
## # A tibble: 6 x 3
##   weekday   daily_steps daily_sleep
##   <ord>           <dbl>       <dbl>
## 1 Monday          9273.        420.
## 2 Tuesday         9183.        405.
## 3 Wednesday       8023.        435.
## 4 Thursday        8184.        401.
## 5 Friday          7901.        405.
## 6 Saturday        9871.        419.
```

```
ggplot(weekday_steps_sleep)+
  geom_col(aes(weekday, daily_steps), fill= "darkblue")+
  geom_hline(yintercept = 7500, colour ="red")+
  labs(title = "DAILY STEP PER WEEKDAY", x= " ", y = " ")+
  theme(plot.title = element_text(hjust = 0.5),axis.text.x = element_text(angle = 45,vjust = 0.5, hjust
```

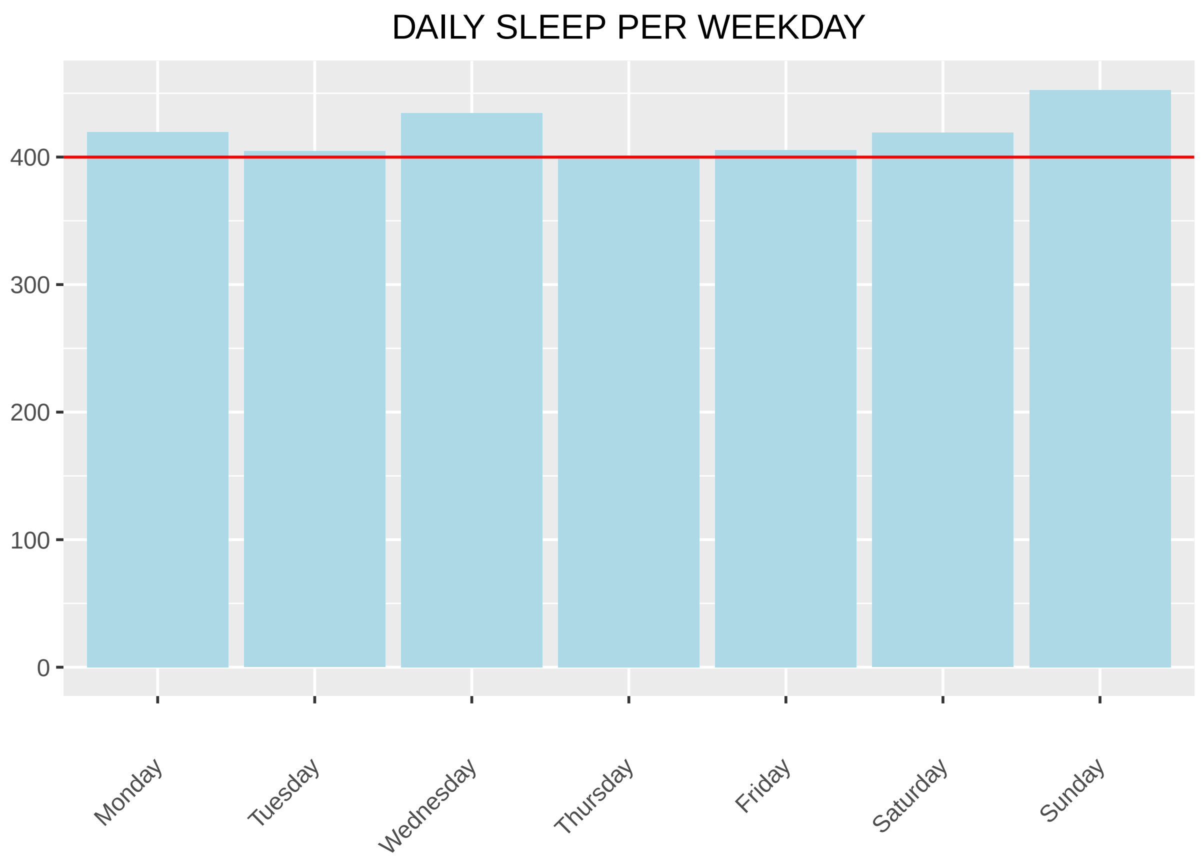


From this graph we can infer that Fitbit Users walk an average of 7500 steps daily asides sundays which according to CDC's reccommendation falls under "somewhat active".

Researchers found that, in a study published on March 24, 2020, in the Journal of the American Medical Association, "taking 8,000 steps per day was associated with a 51% lower risk for all-cause mortality (or death from all causes), compared with taking 4,000 steps per day, a number considered to be low for adults." Compared to 4,000 steps a day, walking 12,000 steps was linked to a 65% decreased risk.

For most adults, the CDC advises taking 8,000 steps a day, with a goal of 10,000 steps. **Although we had 7500 it's averagely okay but could be much better.**

```
ggplot(weekday_steps_sleep)+
  geom_col(aes(weekday, daily_sleep), fill= "lightblue")+
  geom_hline(yintercept = 400, colour ="red")+
  labs(title = "DAILY SLEEP PER WEEKDAY", x= " ", y = " ")+
  theme(plot.title = element_text(hjust = 0.5),axis.text.x = element_text(angle = 45,vjust = 0.5, hjust
```



From this graph we can infer that Fitbit users sleep and average of 400 minutes daily with sunday having the highest time spent sleeping owing to possibly less activity(weekend).

However 400 minutes which is equivalent to 6.67 hours daily is below the reccommended sleep by CDC for adults which is 8 hours.

5.2 Most/least active days

We want to ascertain the the time users were more/least active during the day using the hourly steps dataset

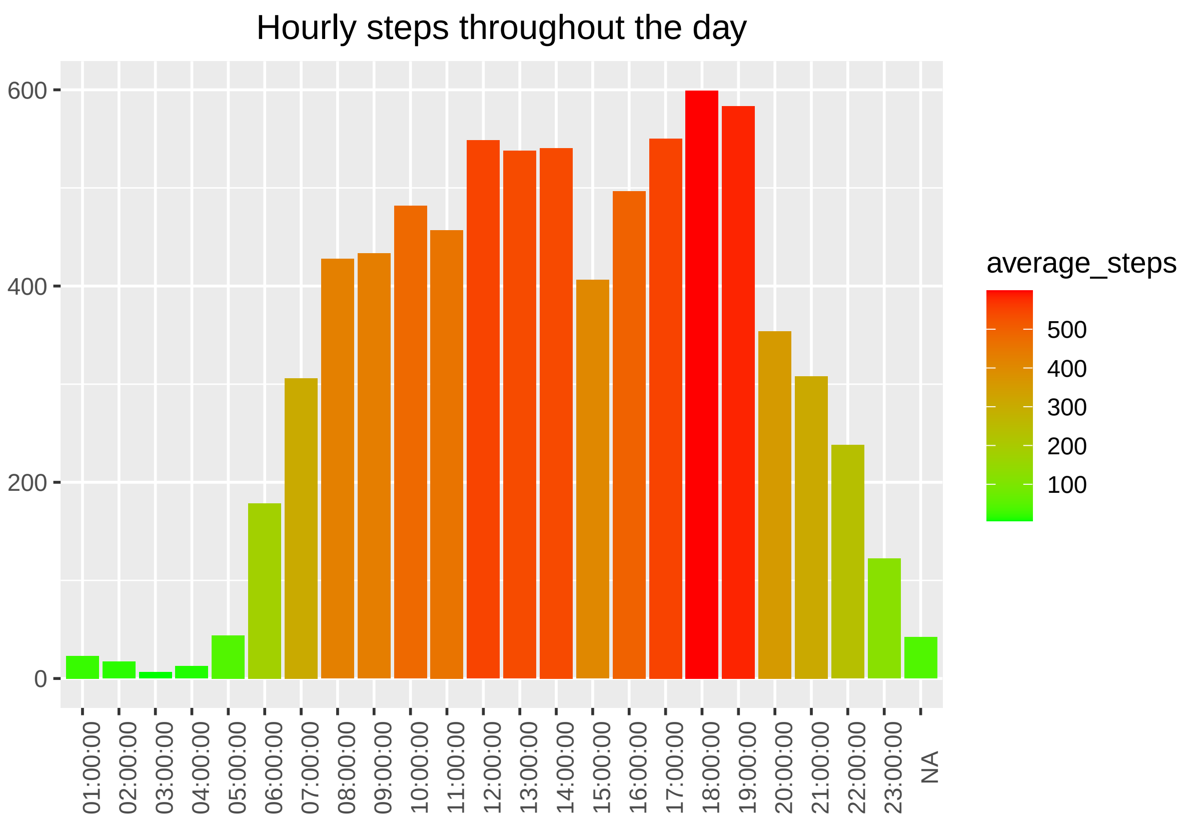
```
hourly_steps <- hourly_steps %>%
  rename(date_time = ActivityHour) %>%
  separate(date_time, into = c("date", "time"), sep = " ") %>%
  mutate(date = ymd(date))
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 934 rows [1, 25, 49, 73,
## 97, 121, 145, 169, 193, 217, 241, 265, 289, 313, 337, 361, 385, 409, 433, 457,
## ...].
```

```
head(hourly_steps)
```

```
##           Id       date     time StepTotal
## 1 1503960366 2016-04-12     <NA>       373
## 2 1503960366 2016-04-12 01:00:00       160
## 3 1503960366 2016-04-12 02:00:00       151
## 4 1503960366 2016-04-12 03:00:00         0
## 5 1503960366 2016-04-12 04:00:00         0
## 6 1503960366 2016-04-12 05:00:00         0
```

```r
hourly_steps %>%
  group_by(time) %>%
  summarize(average_steps = mean(StepTotal)) %>%
  ggplot() +
  geom_col(mapping = aes(x=time, y = average_steps, fill = average_steps)) +
  labs(title = "Hourly steps throughout the day", x="", y="") +
  theme(plot.title = element_text(hjust = 0.5))+
  scale_fill_gradient(low = "green", high = "red")+
  theme(axis.text.x = element_text(angle = 90))
```



5.3 Correlations among different Variables

We would now check for correlations and possibly causations between :

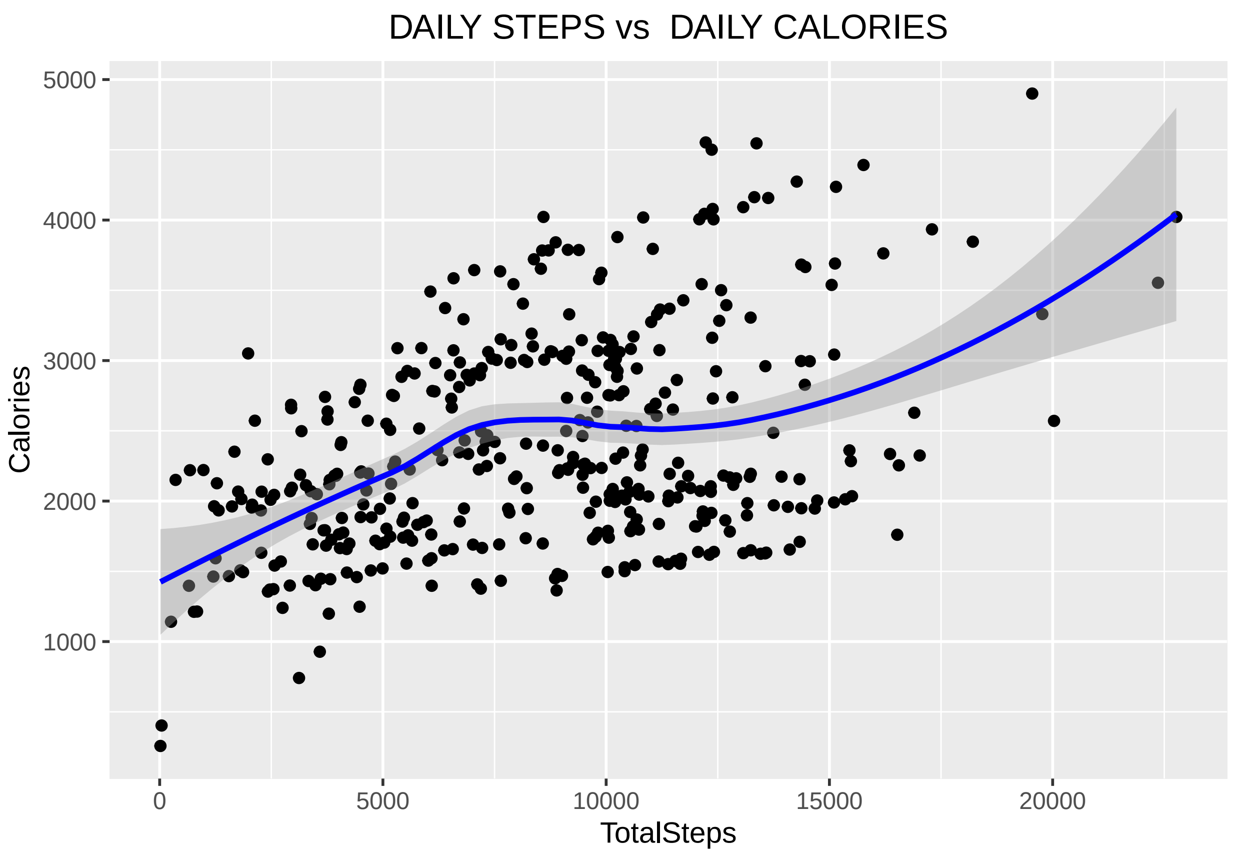
daily_steps and calories and

daily_steps and daily_sleep

sendentary minutes and Minutes asleep

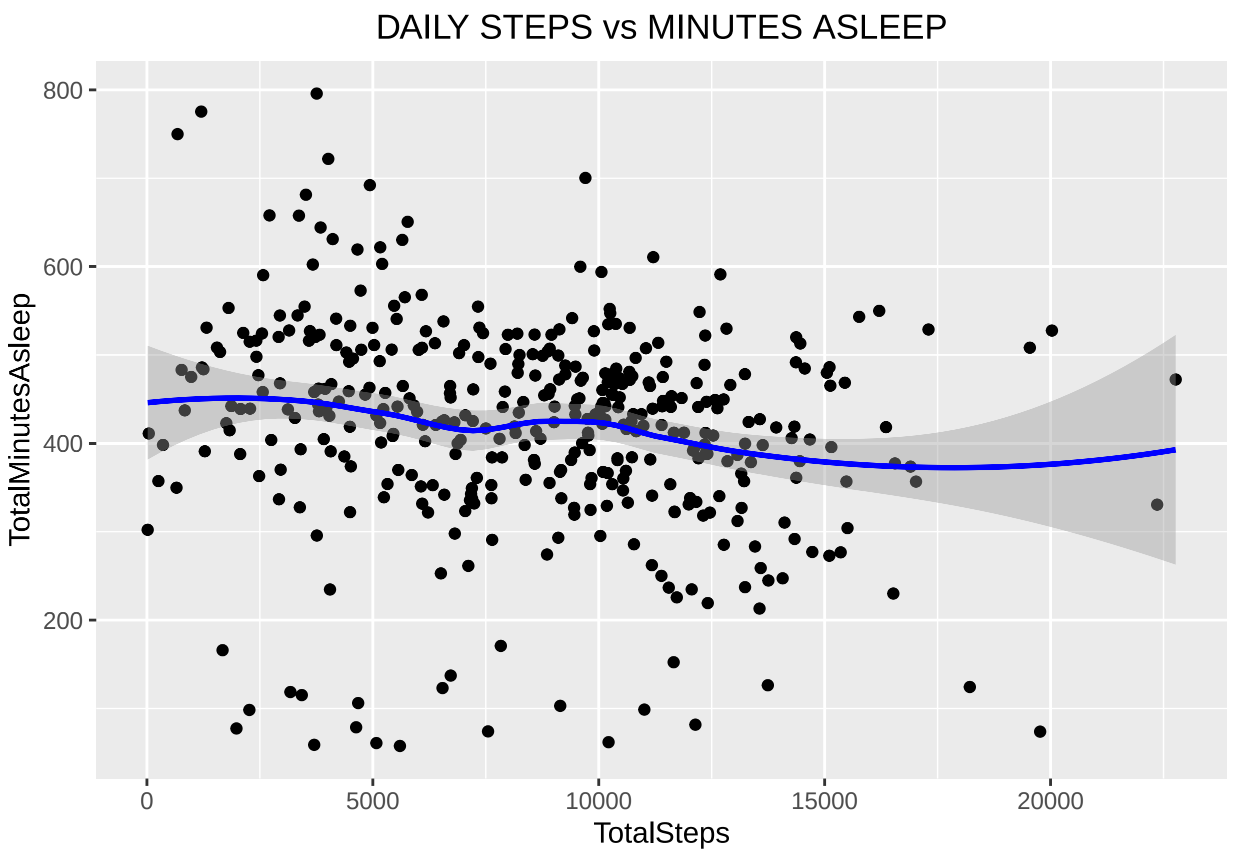
```r
# daily_steps and Calories
ggplot(daily_activity_sleep, aes(x = TotalSteps, y =Calories))+
  geom_jitter()+
  geom_smooth(colour = "blue")+
  labs(title = "DAILY STEPS vs  DAILY CALORIES")+
  theme(plot.title = element_text(hjust = 0.5),
        plot.background = element_rect())
```

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



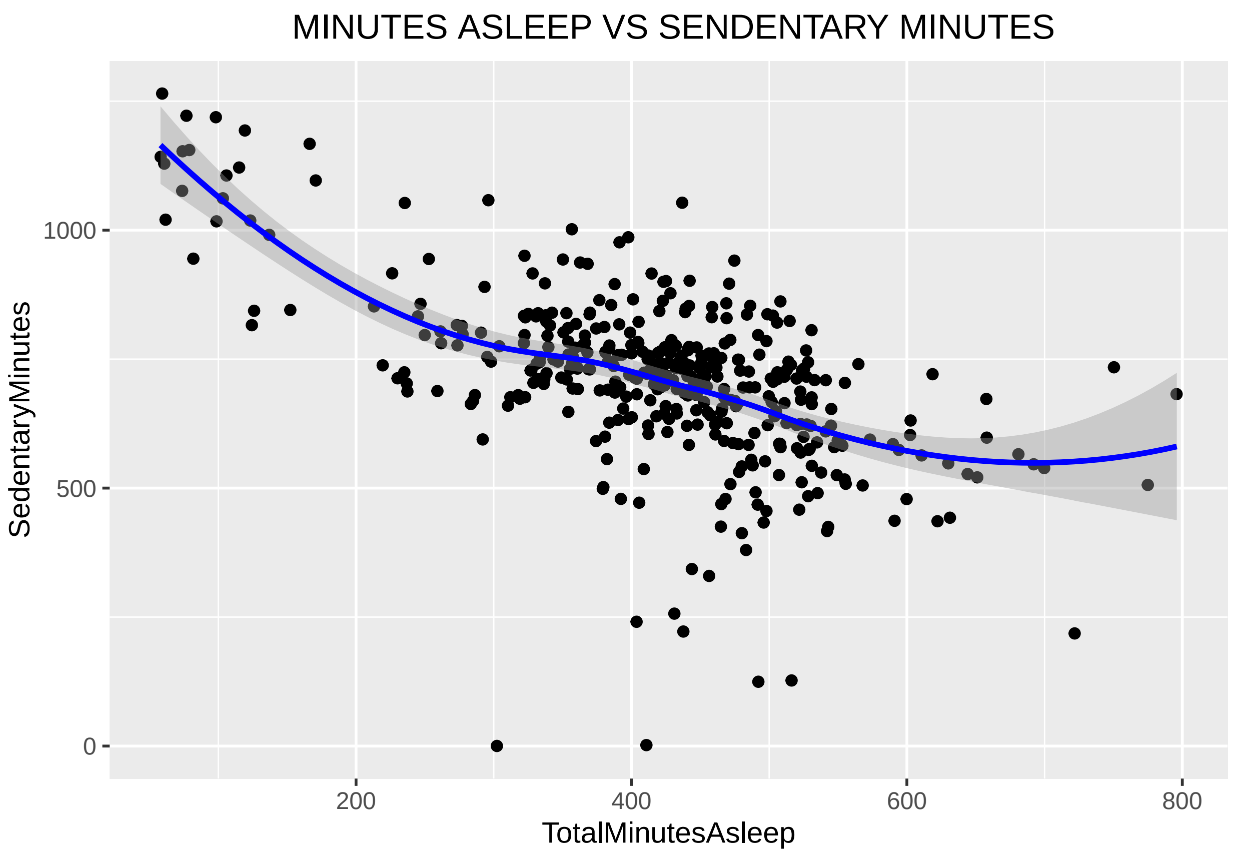
```r
# daily steps and minutes asleep
ggplot(daily_activity_sleep, aes(x = TotalSteps, y =TotalMinutesAsleep))+
  geom_jitter()+
  geom_smooth(colour = "blue")+
  labs(title = "DAILY STEPS vs MINUTES ASLEEP")+
  theme(plot.title = element_text(hjust = 0.5),
        plot.background = element_rect())
```

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



```
# sendentary minutes and minutes asleep
ggplot(daily_activity_sleep, aes (x=TotalMinutesAsleep, y =SedentaryMinutes))+
  geom_jitter()+
  geom_smooth(colour = "blue")+
  labs( title = " MINUTES ASLEEP VS SENDENTARY MINUTES")+
  theme(plot.title = element_text(hjust = 0.5),
        plot.background = element_rect())
```

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



From the first plot we can infer a positive relationship between Daily steps and Daily Calories meaning the more steps the users took in the day the more calories they burnt.

From the second plot we can infer no relationship exists between Daily steps and Minutes asleep .

From the third plot we can infer a negative relationship between Sedentary minutes and Sleep time. Howevever, **if Bellabeat users want to improve their sleep, Bellabeat app can recommend reducing sedentary time.**

5.4 Use of Smart Devices

We will calculate the number of users who use their smart device on a daily basis, categorizing our sample into three groups based on the date interval of 31 days:

- High use - users who use their device between 21 and 31 days.
- Moderate use - users who use their device between 10 and 20 days.
- Low use - users who use their device between 1 and 10 days.

Creating a new dataframe to house this information.

```r
daily_use <- daily_activity_sleep %>%
  group_by(Id) %>%
  summarize(days_used =sum(n())) %>%
  mutate(usage = case_when(
    days_used >= 1 & days_used <= 10 ~ "Low use",
    days_used >= 11 & days_used <= 20 ~ "Moderate use",
    days_used >= 21 & days_used <= 31 ~ "High use",
      )
    )
head(daily_use)
```

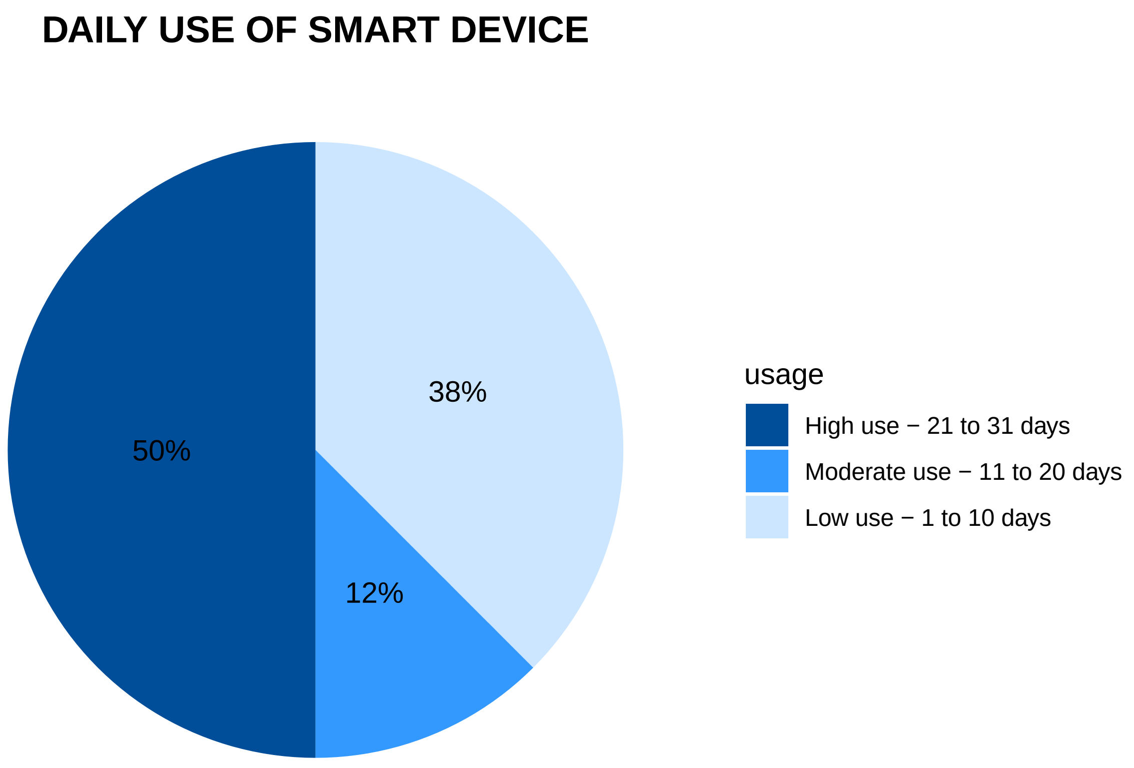
```
## # A tibble: 6 x 3
##           Id days_used usage
##        <dbl>     <int> <chr>
## 1 1503960366        25 High use
## 2 1644430081         4 Low use
## 3 1844505072         3 Low use
## 4 1927972279         5 Low use
## 5 2026352035        28 High use
## 6 2320127002         1 Low use
```

To better visualize the results on the graph, we are going to create a percentage data frame. Our utilization levels are also being ordered.

```
daily_use_percent <- daily_use %>%
  group_by(usage) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(usage) %>%
  summarise(total_percent = total / totals) %>%
  mutate(labels = scales::percent(total_percent))

daily_use_percent$usage <- factor(daily_use_percent$usage, levels = c("High use", "Moderate use", "Low u



head(daily_use_percent)
```

```
## # A tibble: 3 x 3
##   usage        total_percent labels
##   <fct>                <dbl> <chr>
## 1 High use             0.5   50%
## 2 Low use              0.375 38%
## 3 Moderate use         0.125 12%
```

```
daily_use_percent %>%
  ggplot(aes(x="",y=total_percent, fill=usage)) +
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0)+
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5, size=14, face = "bold")) +
  geom_text(aes(label = labels),
            position = position_stack(vjust = 0.5))+
  scale_fill_manual(values = c("#004d99","#3399ff","#cce6ff"),
                    labels = c("High use - 21 to 31 days",
                               "Moderate use - 11 to 20 days",
                               "Low use - 1 to 10 days"))+

  labs(title="DAILY USE OF SMART DEVICE")
```



From this visualization we can infer that

- 50% of the users use their smart devices highly(21-31 days)
- 12% of the users use their smart devices moderately(11-20 days)
- 38% of the users use their smart devices rarely/lowly(1-10 days)

6. ACT (CONCLCUSION)

In summary, Bellabeat has been able to provide women with information about their own health and behaviors by gathering data on activity, sleep, stress, and reproductive health. Bellabeat was established in 2013 and has since expanded significantly, positioning itself as a tech-driven health firm catering to women.

However having done an in-depth analysis accompanied by visualisations of some particular datasets. Here are a few recommendations to improve BellaBeat's app and in turn inform BellaBeat marketing strategy.

Daily notification on steps and posts on BellaBeat app

With the exception of Sundays, users fall into 4 categories, and we found that on average, they walk over 7,500 steps every day. We can motivate users to complete the CDC's suggested 8,000 steps by setting alarms to notify them when they haven't completed the required number of steps and by posting messages on our app outlining the advantages of accomplishing that goal. According to the CDC, the death risk decreases with increasing step count. Additionally, we observed a favorable relationship between calories and steps.

Socialisation Features

With data obtained from hourly and daily metrics. Users can be engaged further by receiving prompts on how they are doing at the point of the day or week when compared to others in their age/location/gender group.

- For Activity Comparison with Age Group:

"Good job! You've taken as many steps as those in your age group!"

"You're keeping up with your age group's activity levels. Keep it going!"

- For Activity Comparison with Past Performance:

"Keep it up! You're almost as active as you were this time last week!"

"You're making great progress. Stay consistent!"

- For Activity Comparison with Weekly Percentile:

"You're not alone in this! You're in the 50th percentile of most active minutes this week!"

"You're among the most active users this week. Keep pushing!"

- For Recovery:

"Rest up! A good recovery is just as important as a good run."

"Your body needs rest too. Take a break and recover."

Sleep Notification and techniques

Our findings indicate that users get less than 8 hours of sleep each day. Users might schedule a specific time to go to bed and get a notification a few minutes beforehand to get ready for bed. Provide customers with more sleep-related materials, such as breathing exercises, calming music podcasts, and sleep strategies.

Reward system

We may develop a sort of game on our app for a little period of time because we know that not everyone is driven by notifications. The goal of the game would be to advance through several stages according to how many steps you take each day. In order to advance to the next level, you must continue your current level of activity for a while—perhaps a month. You would receive a set number of stars for completing each level, which you could then exchange for merchandise or discounts on other Bellabeat products.

Special Thanks

The following authors, who have finished their own capstone work, deserve all credit. I was able to get direction and clarity on how to approach my first attempt at R programming thanks to their writings. I strongly advise all students attempting this capstone to review their work:

1. https://www.kaggle.com/code/macarenalacasa/capstone-case-study-bellabeat/notebook
2. https://www.kaggle.com/code/zulkhaireesulaiman/bellabeat-capstone-project-in-r#-6.-Discussions