Bellabeat Case Study with R

Janice Lee

1/12/2022

**About Bellabeat**

Bellabeat is a high-tech manufacturer of health-focused products for women. It’s a successful company that has the potential to become a large player in the global smart device market. They collect data on activity, sleep, stress, and reproductive health to allow their users to learn about their personal health habits for effective change.

**Scenario**

You have been asked to focus on one of Bellabeat’s five products and analyze Fitbit data to gain insight into how non-Bellabeat consumers are using their smart devices. The insights you discover will then be presented to stakeholders such as Bellabeat’s co-founders, Urska Srsen and Sando Mur, as well as the marketing analytics team, to help guide marketing strategy.

**Steps & Notes for Preparation:**

Dataset information: Thirty-three eligible Fitbit users consented to the submission of personal tracker data. The data consists of information about daily activity, sleep, and weight that can be used to explore users’ health habits. There are issues with bias and credibility since we don’t know where or how this survey was conducted. We are also unaware of who participated in the survey, if it was even randomized, and how the questions were asked. This means we cannot check for any bias and therefore must question it’s validity. The size of the sample group is merely 33 people, this is very small and the data is not original as it is from a survey that was distributed via Amazon Mechanical Turk. The data is not all-inclusive, the sample size is small and we cannot conclude that this will be a good sample group for our target market, especially since our brand is catered towards women. The tables however, give us good examples of how people are using their devices. This data is not current since the survey was conducted nearly six years ago(collected between 03.12.2016 and 05.12.2016) and the tech industry is always improving. Tools: I chose to use R since the datasets used are small enough to work with and R is able to clean, transform, analyze and visualize data all in one place.

**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 marketing strategy?

##Load the csv files

#downloading packages to use  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.5 ✓ purrr 0.3.4  
## ✓ tibble 3.1.6 ✓ dplyr 1.0.7  
## ✓ tidyr 1.1.4 ✓ stringr 1.4.0  
## ✓ readr 2.1.1 ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(tidyr)  
library(dplyr)  
library(ggplot2)  
library(lubridate)

##   
## Attaching package: 'lubridate'

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

library(stringr)  
library(janitor)

##   
## Attaching package: 'janitor'

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

library(skimr)  
library(here)

## here() starts at /Users/janicelee

library(RColorBrewer)  
  
#uploading all daily csv files  
daily\_activity <- read\_csv("Desktop/dailyActivity\_merged.csv")

## Rows: 940 Columns: 15

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

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

daily\_calories <- read\_csv("Desktop/dailyCalories\_merged.csv")

## Rows: 940 Columns: 3

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): ActivityDay  
## dbl (2): Id, Calories

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

daily\_intensities <- read\_csv("Desktop/dailyIntensities\_merged.csv")

## Rows: 940 Columns: 10

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

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

daily\_steps <- read\_csv("Desktop/dailySteps\_merged.csv")

## Rows: 940 Columns: 3

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): ActivityDay  
## dbl (2): Id, StepTotal

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

sleep\_day <- read\_csv("Desktop/sleepDay\_merged.csv")

## Rows: 413 Columns: 5

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

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

weight\_log <- read\_csv("Desktop/weightLogInfo\_merged.csv")

## Rows: 67 Columns: 8

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

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

#view headers for all files  
head(daily\_activity)

## # A tibble: 6 × 15  
## Id ActivityDate TotalSteps TotalDistance TrackerDistance LoggedActivitie…  
## <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>

head(daily\_calories)

## # A tibble: 6 × 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

head(daily\_intensities)

## # A tibble: 6 × 10  
## Id ActivityDay SedentaryMinutes LightlyActiveMinutes FairlyActiveMinu…  
## <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>

head(daily\_steps)

## # A tibble: 6 × 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

head(sleep\_day)

## # A tibble: 6 × 5  
## Id SleepDay TotalSleepRecor… TotalMinutesAsle… TotalTimeInBed  
## <dbl> <chr> <dbl> <dbl> <dbl>  
## 1 1503960366 4/12/2016 12:00:… 1 327 346  
## 2 1503960366 4/13/2016 12:00:… 2 384 407  
## 3 1503960366 4/15/2016 12:00:… 1 412 442  
## 4 1503960366 4/16/2016 12:00:… 2 340 367  
## 5 1503960366 4/17/2016 12:00:… 1 700 712  
## 6 1503960366 4/19/2016 12:00:… 1 304 320

head(weight\_log)

## # A tibble: 6 × 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 4/13/201… 134. 294. NA 47.5 FALSE 1.46e12  
## 4 2873212765 4/21/201… 56.7 125. NA 21.5 TRUE 1.46e12  
## 5 2873212765 5/12/201… 57.3 126. NA 21.7 TRUE 1.46e12  
## 6 4319703577 4/17/201… 72.4 160. 25 27.5 TRUE 1.46e12

##Data Cleaning

skim(sleep\_day)

Data summary

|  |  |
| --- | --- |
| Name | sleep\_day |
| Number of rows | 413 |
| Number of columns | 5 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 1 |
| numeric | 4 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| SleepDay | 0 | 1 | 20 | 21 | 0 | 31 | 0 |

**Variable type: numeric**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| Id | 0 | 1 | 5.000979e+09 | 2.06036e+09 | 1503960366 | 3977333714 | 4702921684 | 6962181067 | 8792009665 | ▆▆▇▅▃ |
| TotalSleepRecords | 0 | 1 | 1.120000e+00 | 3.50000e-01 | 1 | 1 | 1 | 1 | 3 | ▇▁▁▁▁ |
| TotalMinutesAsleep | 0 | 1 | 4.194700e+02 | 1.18340e+02 | 58 | 361 | 433 | 490 | 796 | ▁▂▇▃▁ |
| TotalTimeInBed | 0 | 1 | 4.586400e+02 | 1.27100e+02 | 61 | 403 | 463 | 526 | 961 | ▁▃▇▁▁ |

View(sleep\_day)  
sleep\_day\_1 <-  
sleep\_day %>%   
 separate(SleepDay, c("SleepDate", "SleepTime","AM/PM"), sep="([ ])") %>%  
 unite(SleepTimes, 3:4, sep=" ", remove=FALSE)%>%  
 mutate(SleepDate= as.Date(SleepDate, "%m/%d/%Y"))%>%  
 mutate(SleepTime= parse\_time(SleepTimes, "%H:%M:%S %p"))%>%  
 distinct()  
new\_sleep\_day<- subset(sleep\_day\_1, select = -c(3,5))  
View(new\_sleep\_day)  
head(new\_sleep\_day)

## # A tibble: 6 × 6  
## Id SleepDate SleepTime TotalSleepRecor… TotalMinutesAsl… TotalTimeInBed  
## <dbl> <date> <time> <dbl> <dbl> <dbl>  
## 1 1.50e9 2016-04-12 00'00" 1 327 346  
## 2 1.50e9 2016-04-13 00'00" 2 384 407  
## 3 1.50e9 2016-04-15 00'00" 1 412 442  
## 4 1.50e9 2016-04-16 00'00" 2 340 367  
## 5 1.50e9 2016-04-17 00'00" 1 700 712  
## 6 1.50e9 2016-04-19 00'00" 1 304 320

skim(weight\_log)

Data summary

|  |  |
| --- | --- |
| Name | weight\_log |
| Number of rows | 67 |
| Number of columns | 8 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 1 |
| logical | 1 |
| numeric | 6 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| Date | 0 | 1 | 19 | 21 | 0 | 56 | 0 |

**Variable type: logical**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | mean | count |
| IsManualReport | 0 | 1 | 0.61 | TRU: 41, FAL: 26 |

**Variable type: numeric**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| Id | 0 | 1.00 | 7.009282e+09 | 1.950322e+09 | 1.503960e+09 | 6.962181e+09 | 6.962181e+09 | 8.877689e+09 | 8.877689e+09 | ▁▁▂▇▆ |
| WeightKg | 0 | 1.00 | 7.204000e+01 | 1.392000e+01 | 5.260000e+01 | 6.140000e+01 | 6.250000e+01 | 8.505000e+01 | 1.335000e+02 | ▇▃▃▁▁ |
| WeightPounds | 0 | 1.00 | 1.588100e+02 | 3.070000e+01 | 1.159600e+02 | 1.353600e+02 | 1.377900e+02 | 1.875000e+02 | 2.943200e+02 | ▇▃▃▁▁ |
| Fat | 65 | 0.03 | 2.350000e+01 | 2.120000e+00 | 2.200000e+01 | 2.275000e+01 | 2.350000e+01 | 2.425000e+01 | 2.500000e+01 | ▇▁▁▁▇ |
| BMI | 0 | 1.00 | 2.519000e+01 | 3.070000e+00 | 2.145000e+01 | 2.396000e+01 | 2.439000e+01 | 2.556000e+01 | 4.754000e+01 | ▇▁▁▁▁ |
| LogId | 0 | 1.00 | 1.461772e+12 | 7.829948e+08 | 1.460444e+12 | 1.461079e+12 | 1.461802e+12 | 1.462375e+12 | 1.463098e+12 | ▇▇▆▇▇ |

View(weight\_log)  
weight\_log\_1<-  
weight\_log %>% separate(Date, c("LogDate", "LogTime","AM/PM"), sep="([ ])")%>%  
 unite(LogTimes, 3:4, sep=" ", remove=FALSE)%>%  
 mutate(LogDate= as.Date(LogDate, "%m/%d/%Y"))%>%  
 mutate(LogTime= parse\_time(LogTimes, "%H:%M:%S %p"))%>%  
 distinct()  
weight\_log\_1= weight\_log\_1[,!sapply(weight\_log\_1, function(x) mean(is.na(x)))>0.5]  
new\_weight\_log<- subset(weight\_log\_1, select = -c(3,5))  
View(new\_weight\_log)  
head(new\_weight\_log)

## # A tibble: 6 × 8  
## Id LogDate LogTime WeightKg WeightPounds BMI IsManualReport LogId  
## <dbl> <date> <time> <dbl> <dbl> <dbl> <lgl> <dbl>  
## 1 1.50e9 2016-05-02 23:59:59 52.6 116. 22.6 TRUE 1.46e12  
## 2 1.50e9 2016-05-03 23:59:59 52.6 116. 22.6 TRUE 1.46e12  
## 3 1.93e9 2016-04-13 01:08:52 134. 294. 47.5 FALSE 1.46e12  
## 4 2.87e9 2016-04-21 23:59:59 56.7 125. 21.5 TRUE 1.46e12  
## 5 2.87e9 2016-05-12 23:59:59 57.3 126. 21.7 TRUE 1.46e12  
## 6 4.32e9 2016-04-17 23:59:59 72.4 160. 27.5 TRUE 1.46e12

colnames(daily\_activity)[2] <-"ActivityDay"  
colnames(daily\_steps)[3] <- "TotalSteps"

After checking their headers, I can see that dates for sleep\_day and weight\_log are in character format not date format so I fix this. I also check for duplicates and missing information. I found 3 duplicate rows in the sleep\_day dataset and deleted them. I also found that 65 of 67 rows are empty in the Fat column under the weight\_log dataset so decided to get rid of the column completely by using a formula that gets rid of columns where more than 50% of entries are empty/NA. I also clean up the names in daily\_activity and daily\_steps to make columns match up when merging sets of data together.

##Data Prep

#let's see how many unique participants are there for each activity  
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(new\_sleep\_day$Id)

## [1] 24

n\_distinct(new\_weight\_log$Id)

## [1] 8

#I found that most people use their Fitbit device to track daily activities(33), intensities(33), calories(33), steps(33), and sleep(24). We can see that our activities data is just a merge of our intensities, calories, and steps data. We should merge these together to make sure no data was lost in on dataset into the other. With this in mind we only need to focus on analyzing the daily activities and sleep categories.  
  
#merging everything together  
m0 <- merge(daily\_activity,daily\_intensities,  
 by=c("Id","ActivityDay","SedentaryMinutes", "LightlyActiveMinutes",  
 "FairlyActiveMinutes","VeryActiveMinutes", "SedentaryActiveDistance",   
 "LightActiveDistance", "ModeratelyActiveDistance", "VeryActiveDistance"))  
daily\_data <- merge(m0, daily\_steps,  
 by= c("Id","ActivityDay", "TotalSteps"))  
View(daily\_data)  
  
#I check our new dataframe for duplicate data and find that there's no duplicate data since we still have 940 rows. I can also see that ActivityDay is not in date format so I fix that.  
  
daily\_data%>% distinct() %>% View  
daily\_data<-  
 daily\_data %>% mutate(ActivityDate= as.Date(ActivityDay, "%m/%d/%Y"))  
new\_daily\_data<- subset(daily\_data, select = -c(2))  
View(new\_daily\_data)  
str(new\_daily\_data)

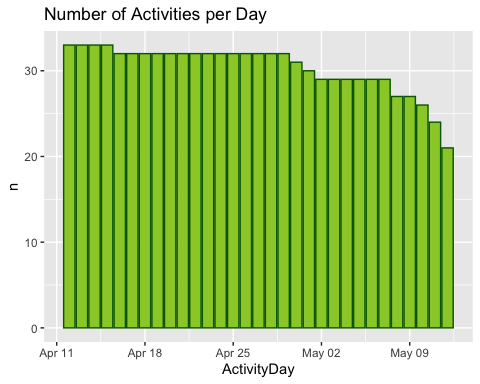
## 'data.frame': 940 obs. of 15 variables:  
## $ Id : num 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...  
## $ TotalSteps : num 13162 10735 10460 9762 12669 ...  
## $ SedentaryMinutes : num 728 776 1218 726 773 ...  
## $ LightlyActiveMinutes : num 328 217 181 209 221 164 233 264 205 211 ...  
## $ FairlyActiveMinutes : num 13 19 11 34 10 20 16 31 12 8 ...  
## $ VeryActiveMinutes : num 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 ...  
## $ 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 ...  
## $ Calories : num 1985 1797 1776 1745 1863 ...  
## $ ActivityDate : Date, format: "2016-04-12" "2016-04-13" ...

new\_daily\_data<-rename(new\_daily\_data, ActivityDay = "ActivityDate")  
  
new\_daily\_data %>%  
 summarise(Activity\_Participants= n\_distinct(new\_daily\_data$Id))

## Activity\_Participants  
## 1 33

##Analysis Now our data is all clean and formatted the way we want it. Lets analyze our data.

#plot number of activities per day  
new\_daily\_data%>%  
 count(ActivityDay, sort=TRUE) %>%  
 ggplot()+geom\_col(mapping=aes(x=ActivityDay,y=n),fill="yellowgreen",color="darkgreen")+labs(title= "Number of Activities per Day")

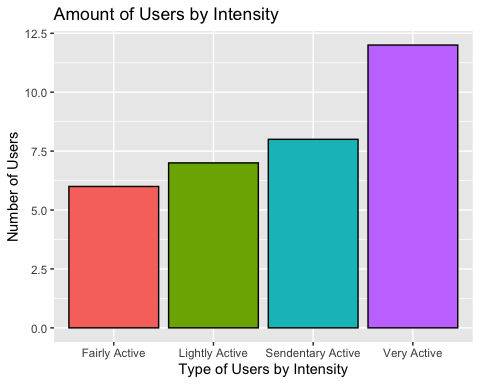


#The graph shows us that at least 12 people have missed an activity throughout the 31 days of this study. It also shows us that users are less dedicated as time goes on. Lets see how many people have gone without their fitbit assuming that 1440 sedentary minutes is equal to 24 sedentary hours.  
fitbit\_off <- new\_daily\_data %>%   
 filter(SedentaryMinutes == 1440) %>%   
 group\_by(Id) %>%   
 summarise(count = n()) %>%   
 print()

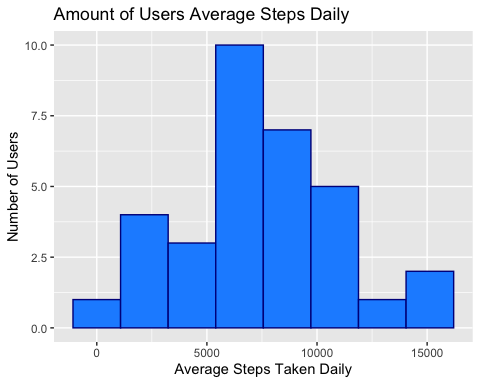
## # A tibble: 17 × 2  
## Id count  
## <dbl> <int>  
## 1 1503960366 1  
## 2 1844505072 9  
## 3 1927972279 13  
## 4 4020332650 14  
## 5 4057192912 1  
## 6 4319703577 1  
## 7 4388161847 1  
## 8 4702921684 1  
## 9 5577150313 2  
## 10 6117666160 5  
## 11 6290855005 4  
## 12 6775888955 9  
## 13 7007744171 1  
## 14 7086361926 1  
## 15 8253242879 1  
## 16 8583815059 6  
## 17 8792009665 9

#The output shows us that 17 users went at LEAST one day without using their fitbit. We want to see at which intensities our users are active while accounting for the number of users who had their fitbit off but still counted as Sedentary Active and skewing our data.  
user\_activity <- new\_daily\_data %>%   
 filter(SedentaryMinutes != 1440) %>%   
 group\_by(Id) %>%   
 summarize(total\_very\_active\_mins = sum(VeryActiveMinutes),  
 total\_fairly\_active\_mins = sum(FairlyActiveMinutes),  
 total\_lightly\_active\_mins = sum(LightlyActiveMinutes),  
 total\_sendentary\_mins = sum(SedentaryMinutes),  
 total\_mins = sum(VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, SedentaryMinutes),  
 percent\_very\_active = (total\_very\_active\_mins/total\_mins)\*100,  
 percent\_fairly\_active = (total\_fairly\_active\_mins/total\_mins)\*100,  
 percent\_lightly\_active = (total\_lightly\_active\_mins/total\_mins)\*100,  
 percent\_sendentary\_active = (total\_sendentary\_mins/total\_mins)\*100) # Calculate percentages of minutes by intensity  
user\_activity <- user\_activity %>%   
 mutate(intensity =  
 case\_when(percent\_very\_active > mean(percent\_very\_active) ~ "Very Active",  
 percent\_fairly\_active > mean(percent\_fairly\_active) ~ "Fairly Active",  
 percent\_lightly\_active > mean(percent\_lightly\_active) ~ "Lightly Active",  
 percent\_sendentary\_active > mean(percent\_sendentary\_active) ~ "Sendentary Active"))  
ds\_activity <- user\_activity %>%   
 group\_by(intensity) %>%   
 summarise(count = n())  
ggplot(ds\_activity, aes(x = intensity, y = count, fill = intensity)) +  
 geom\_histogram(stat = "identity", color= "black") +  
 ylab("Number of Users") +  
 xlab("Type of Users by Intensity") +  
 labs(title = "Amount of Users by Intensity") +  
 theme(legend.position = "none")

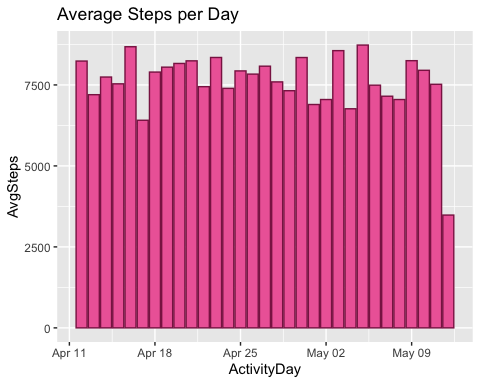
## Warning: Ignoring unknown parameters: binwidth, bins, pad



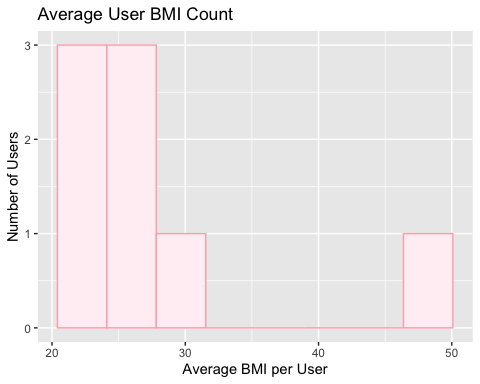
#The graph shows us that 'Very Active' accounts for the highest number of users. Let's see how active our users are by checking total average steps per day.  
new\_daily\_steps <- new\_daily\_data %>%   
 group\_by(Id) %>%   
 summarise(avg\_daily\_steps = mean(TotalSteps))  
ggplot(new\_daily\_steps, aes(x = avg\_daily\_steps)) +  
 geom\_histogram(bins = 8, fill = "dodgerblue", color = "blue4") +  
 ylab("Number of Users") +  
 xlab("Average Steps Taken Daily") +  
 labs(title = "Amount of Users Average Steps Daily")



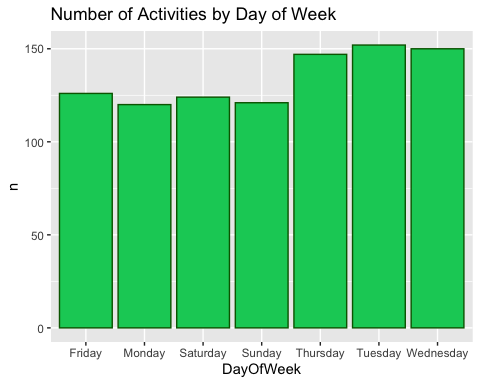
#avg. steps per day  
new\_daily\_data%>%  
 group\_by(ActivityDay) %>%  
 summarise\_at(vars(TotalSteps), list(AvgSteps = mean))%>%  
 ggplot()+geom\_col(mapping = aes(x=ActivityDay, y=AvgSteps),fill="hotpink2", color="violetred4")+ labs(title="Average Steps per Day")



#Users tend to take 5,000 to 10,000 steps daily on average. As we see in the second graph, this stays pretty consistent over time as well. This is not a high number since the CDC recommends that a person takes 10,000 steps per day. This makes sense since most users probably purchased a Fitbit with the desire to become more active. Let's check with the information that we have, how fit the users are.  
  
body\_weight <- new\_weight\_log %>%   
 group\_by(Id) %>%   
 summarise(avg\_bmi = mean(BMI))  
ggplot(body\_weight, aes(x =avg\_bmi)) +  
 geom\_histogram(bins = 8, fill = "lavenderblush", color = "lightpink1") +  
 ylab("Number of Users") +  
 xlab("Average BMI per User") +  
 labs(title = "Average User BMI Count")

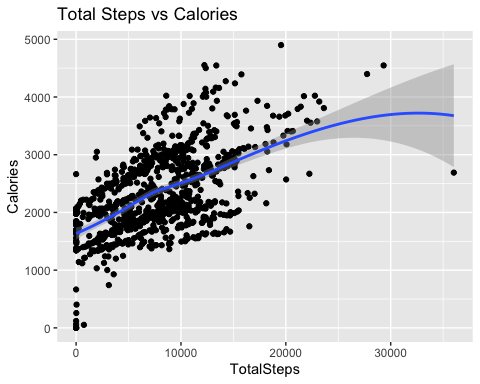


#Though only 8 people logged their weights, 5 of those 8 fall within a BMI higher than 25 which, according to the CDC, is considered overweight. One person has a BMI over 30 which is considered obese.   
  
#most active days of the week  
new\_daily\_data$DayOfWeek<-weekdays(new\_daily\_data$ActivityDay)   
new\_daily\_data%>%  
 count(DayOfWeek,sort=TRUE)%>%  
 ggplot()+geom\_col(mapping= aes(x=DayOfWeek, y=n),fill="springgreen3" ,color="darkgreen")+labs(title="Number of Activities by Day of Week")

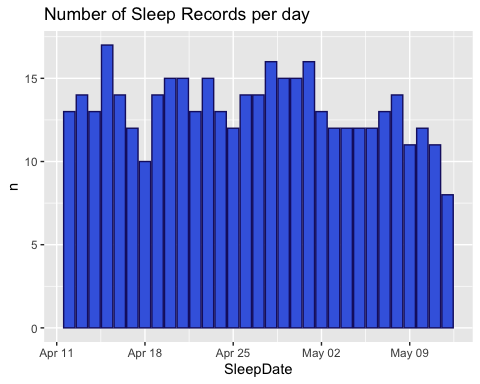


#Tuesday, Wednesday and Thursday are the days of the week with the most activity. The graph shows that people are less active on the weekends.  
  
#whats the relationship between steps and calories  
ggplot(data=new\_daily\_data, aes(x=TotalSteps, y=Calories))+geom\_point()+  
geom\_smooth() + labs(title="Total Steps vs Calories")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



#we see a positive correlation between total steps and calories. The more active they are, the more calories they are burning.  
  
#how many users logged their sleep per day  
new\_sleep\_day%>%  
 count(SleepDate, sort=TRUE) %>%  
 ggplot()+geom\_col(aes(x=SleepDate,y=n),fill="royalblue", color="midnightblue")+labs(title= "Number of Sleep Records per day")



#We can see that their are sudden drops in sleep records within the week. These days fall on Mondays. Not only do Mondays account for the lowest activity but also lowest sleep records.  
  
skim(new\_sleep\_day)

Data summary

|  |  |
| --- | --- |
| Name | new\_sleep\_day |
| Number of rows | 410 |
| Number of columns | 6 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| Date | 1 |
| difftime | 1 |
| numeric | 4 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: Date**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| SleepDate | 0 | 1 | 2016-04-12 | 2016-05-12 | 2016-04-27 | 31 |

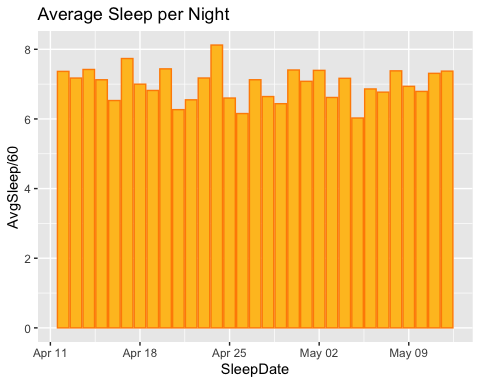
**Variable type: difftime**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| SleepTime | 0 | 1 | 0 secs | 0 secs | 0 secs | 1 |

**Variable type: numeric**

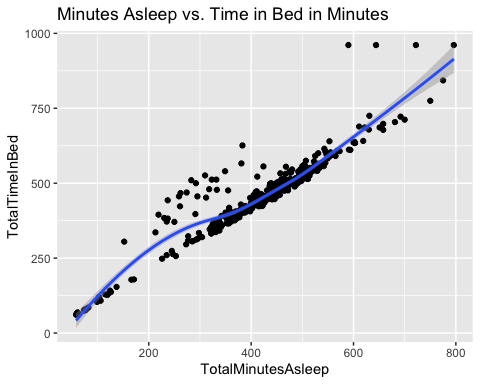
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| Id | 0 | 1 | 4.994963e+09 | 2.060863e+09 | 1503960366 | 3.977334e+09 | 4702921684.0 | 6962181067 | 8792009665 | ▆▆▇▅▃ |
| TotalSleepRecords | 0 | 1 | 1.120000e+00 | 3.500000e-01 | 1 | 1.000000e+00 | 1.0 | 1 | 3 | ▇▁▁▁▁ |
| TotalMinutesAsleep | 0 | 1 | 4.191700e+02 | 1.186400e+02 | 58 | 3.610000e+02 | 432.5 | 490 | 796 | ▁▂▇▃▁ |
| TotalTimeInBed | 0 | 1 | 4.584800e+02 | 1.274600e+02 | 61 | 4.037500e+02 | 463.0 | 526 | 961 | ▁▃▇▁▁ |

#The average TotalMinutesAsleep equates to about 7 hours a night.  
  
#average sleep  
new\_sleep\_day%>%  
 group\_by(SleepDate) %>%  
 summarise\_at(vars(TotalMinutesAsleep), list(AvgSleep = mean))%>%  
 ggplot()+geom\_col(mapping = aes(x=SleepDate, y=AvgSleep/60),fill="goldenrod1", color="darkorange")+ labs(title="Average Sleep per Night")

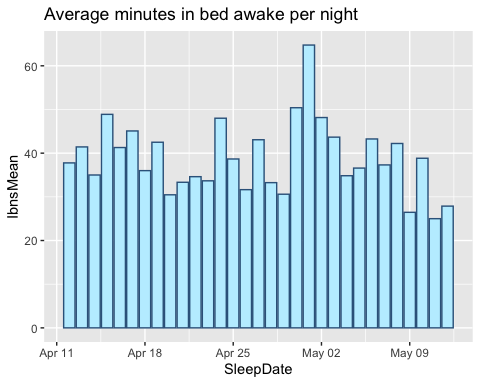


#CDC recommends at least 7 hours of sleep per night for those over the age of 18. The younger you are, the more sleep is recommended.  
  
##relationship between minutes asleep and time in bed  
ggplot(data=new\_sleep\_day, aes(x=TotalMinutesAsleep, y=TotalTimeInBed))+ geom\_point()+  
geom\_smooth()+labs(title="Minutes Asleep vs. Time in Bed in Minutes")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

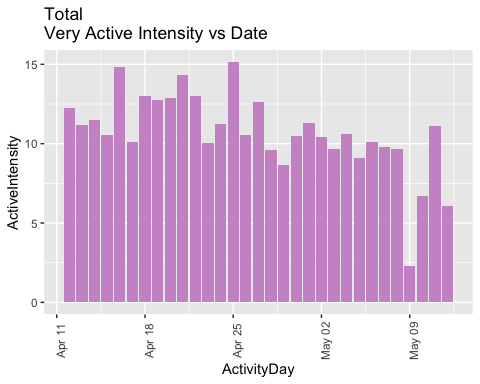


#As expected, there is an almost completely linear trend between minutes asleep and time in bed. To help users improve their sleep schedules, the company should consider using notification to go to sleep.  
  
#average time in bed but awake  
new\_sleep\_day%>%  
 mutate(InBedNoSleep=TotalTimeInBed-TotalMinutesAsleep)%>%  
 group\_by(SleepDate) %>%  
 summarise\_at(vars(InBedNoSleep), list(IbnsMean = mean))%>%  
 ggplot()+geom\_col(mapping = aes(x=SleepDate, y=IbnsMean),fill="lightblue1", color="steelblue4")+ labs(title="Average minutes in bed awake per night")



#on average, it took the participants 40 minutes to fall asleep after laying in bed. Going off the previous idea, the company should consider setting the notification to go to sleep to 40 minutes before the user plans on actually sleeping.  
  
#Lets see if being active helps or hurts when trying to get a good nights sleep.  
##intensities data  
new\_daily\_data$ActiveIntensity <- (daily\_intensities$VeryActiveMinutes/60)  
ggplot(data=new\_daily\_data, aes(x=ActivityDay, y=ActiveIntensity))+ geom\_histogram(stat= "identity", fill="plum3")+ theme(axis.text.x = element\_text(angle=90))+labs(title="Total   
Very Active Intensity vs Date")

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



#When comparing the two graphs, we can see that there is no visible correlation between intensity and time in bed awake.

**Recap of my findings**

As time goes by, users tend to be less active on their Fitbits though only 8 people logged their weights, 5 of those 8 fall within a BMI higher than 25 which, according to the CDC, is considered overweight. One person has a BMI over 30 which is considered obese.

‘Very Active’ accounts for the highest number of users participants tend to take 5000 to 10000 steps daily on average .

Tuesday, Wednesday and Thursday are the days of the week with the most activity, people are less active on the weekends.

Positive correlation between total steps and calories. The more active they are, the more calories they are burning.

Not only do Mondays account for the lowest activity but also lowest sleep records.

There is an almost completely linear trend between minutes asleep and time in bed.

On average, it took the participants 40 minutes to fall asleep after laying in bed.

There is no visible correlation between intensity and time in bed awake or sleep.

**Recommendations**

Of all of their products, Bellabeat’s smart watch device, Time, is the most comparable to a Fitbit. Both devices are worn on your wrist to track personalized activity and sleep. I recommend that the marketing team focuses on promoting the categories with high user activity, which are intensities, steps, and sleep. We know that only 8 of 33 users were tracking their weight, showing that weight loss might not be a priority for them. Instead of being marketed as a device that assists weight-loss, Bellabeat should emphasize body positivity and promote a healthy lifestyle. We see that there is a large discrepancy between how many steps your taking/how much sleep you’re getting and the day of the week. By targeting women who live sedentary lifestyles but want to become more active, we can use goal-oriented reminders to take 10,000 steps per day and sleep at least 8 hours a night to improve or maintain their current health habits. Since we found that it takes user an average of 40 minutes to fall asleep, by pushing 8 hours of sleep we are making sure that users will get the CDC recommended 7 hours. Especially since user activity decreased over time, regular reminders can help keep activity consistent. Since our targeted market is specifically geared towards women, our Time device should really emphasize the fact that this product allows you to do all the things a Fitbit can and more. The Time watch is capable of tracking your stress, where you are in your cycle, as well as an option for women who are pregnant. The discreet design also makes it perfect to keep on all day long, including work hours, unlike the Fitbit which may look unprofessional.