Bellabeat Case Study

Lydia Kim

2023-11-24

Introduction

This is a case study created by the Google Data Analytics Certficate. The company of focus, Bellabeat, is a high-tech company that manufactures health-related smart devices. In this case study, we will be analyzing customer data from Fitbit, a well-known fitness company, to inform Bellabeat's marketing strategy.

Product Time, one of Bellabeat's smart wellness products, is a watch that measures the user's activity, sleep, and stress. We will be applying the insights from our analysis to this product.

Uploading the data

Our dataset is titled FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius). This dataset contains personal fitness information from 30 users.

First, let's install all necessary packages

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

Now, let's upload our csv files

```
daily_activity <- read_csv('Fitabase.16/dailyActivity_merged.csv')

hourly_calories <- read_csv('Fitabase.16/hourlyCalories_merged.csv')
hourly_intensity <- read_csv('Fitabase.16/hourlyIntensities_merged.csv')
hourly_steps <- read_csv('Fitabase.16/hourlySteps_merged.csv')

weight <- read_csv('Fitabase.16/weightLogInfo_merged.csv')
sleep <- read_csv('Fitabase.16/sleepDay_merged.csv')
```

All column names are in camel case, so there is no need to clean the names!

Cleaning data

Before we tidy our data, let's get a preview of our data frames (code not shown)

It looks like there are some inaccuracies in the daily_activity data frame. The average human burns at least 1300 calories a day (source: https://www.verywellfit.com/how-many-calories-do-i-burn-every-day-3495464#:~:text=How%20Active%20Are%20You%3F&text=Your%20body%20is%20always%20burning,1%2C300%20calories%E2%80%94and%20likely%20more.), so we can assume either the user took the fitbit off or there was an error with the data collection for calorie entries less than about 1000. Let's remove these rows to prevent the outliers from messing with our results.

```
daily_activity <- daily_activity[daily_activity$Calories > 1000 & daily_activity$TotalSteps > 0, ]
```

Now that we filtered the data, we can move on!

Since the 3 hourly tables contain corresponding Id's and dates, let's merge them now to make the data cleaning process easier

```
hourly_one <- merge(hourly_calories,hourly_intensity,by=c('Id','ActivityHour'))
hourly_activity <- merge(hourly_one,hourly_steps,by=c('Id','ActivityHour'))
```

Great! Now we have a single table containing all the hourly tracked data.

Each table contains many columns, some of which we will not be needing. Let's keep only the variables that will be useful for our analysis

```
daily_activity <- subset(daily_activity, select=-c(TotalDistance,TrackerDistance,LoggedActivitiesDistanc

weight <- subset(weight, select=c(Id,Date,BMI))

sleep <- subset(sleep, select=-TotalSleepRecords)
```

Now that we have our concise tables, let's check for any duplicates or missing data. The row count and distinct row count must be equal to verify there are no duplicates.

```
# daily_activity
cat('**daily_activity**', '\n\nrow count: ',dim(daily_activity)[1], '\ndistinct row count: ', as.charac
```

```
## **daily_activity**
##
## row count:  858
## distinct row count:  858
## missing values:  0
```

```
# hourly_activity
cat('**hourly_activity**', '\n\nrow count: ',dim(hourly_activity)[1], '\ndistinct row count: ', as.chara
```

```
## **hourly_activity**
##
## row count:  22099
## distinct row count:  22099
## missing values:  0
```

```
# weight
cat('**weight**', '\n\nrow count: ',dim(weight)[1], '\ndistinct row count: ', as.character(count(distin
```

```
## **weight**
##
## row count:  67
## distinct row count:  67
## missing values:  0
```

```
# sleep
cat('**sleep**', '\n\nrow count: ',dim(sleep)[1], '\ndistinct row count: ', as.character(count(distinct
```

```
## **sleep**
##
## row count:  413
## distinct row count:  410
## missing values:  0
```

There are 413 rows in the sleep data, but 410 distinct rows. Let's remove the duplicate data

```
sleep <- sleep %>%
  distinct()
dim(sleep)[1]
```

```
## [1] 410
```

Great! Now all our data is clean and ready for analysis.

Analysis

Let's aggregate the weight, sleep, and daily_activity data by the ID and date

```
# setting a consistent date format for aggregation
weight$Date <- weight$Date %>%
  as.Date('%m/%d/%Y')
sleep$SleepDay <- sleep$SleepDay %>%
  as.Date('%m/%d/%Y')
daily_activity$ActivityDate <- daily_activity$ActivityDate %>%
  as.Date('%m/%d/%Y')

# renaming columns for consistency
names(sleep)[2] <- 'ActivityDate'
names(weight)[2] <- 'ActivityDate'

daily_activity <- merge(weight, daily_activity, by=c('Id', 'ActivityDate'), all=TRUE)
daily_activity <- merge(sleep, daily_activity, by=c('Id', 'ActivityDate'), all=TRUE)
```

Considering that the weight and sleep data frames had significantly less rows, it is expected for there to be many missing values in the updated daily_activity data frame.

Let's take a look at the summaries of our data frames to identify any trends, patterns, and observations.

```
summary(daily_activity)
```

```
##        Id              ActivityDate        TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.504e+09   Min.   :2016-04-12   Min.   : 58.0      Min.   : 61.0
##  1st Qu.:2.320e+09   1st Qu.:2016-04-18   1st Qu.:361.0      1st Qu.:403.8
##  Median :4.445e+09   Median :2016-04-26   Median :432.5      Median :463.0
##  Mean   :4.859e+09   Mean   :2016-04-26   Mean   :419.2      Mean   :458.5
##  3rd Qu.:6.962e+09   3rd Qu.:2016-05-03   3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :8.878e+09   Max.   :2016-05-12   Max.   :796.0      Max.   :961.0
##                                           NA's   :452        NA's   :452
##       BMI          TotalSteps    VeryActiveMinutes FairlyActiveMinutes
##  Min.   :21.45   Min.   :    4   Min.   :  0.00    Min.   :  0.00
##  1st Qu.:23.96   1st Qu.: 4936   1st Qu.:  0.00    1st Qu.:  0.00
##  Median :24.39   Median : 8062   Median :  7.00    Median :  8.00
##  Mean   :25.19   Mean   : 8359   Mean   : 23.14    Mean   : 14.84
##  3rd Qu.:25.56   3rd Qu.:11101   3rd Qu.: 35.75    3rd Qu.: 21.00
##  Max.   :47.54   Max.   :36019   Max.   :210.00    Max.   :143.00
##  NA's   :795     NA's   :4       NA's   :4         NA's   :4
##  LightlyActiveMinutes SedentaryMinutes    Calories
##  Min.   :  0          Min.   : 125.0   Min.   :1002
##  1st Qu.:148          1st Qu.: 724.0   1st Qu.:1862
##  Median :209          Median :1024.0   Median :2222
##  Mean   :211          Mean   : 960.9   Mean   :2372
```

```
##  3rd Qu.:272          3rd Qu.:1189.8   3rd Qu.:2835
##  Max.   :518          Max.   :1440.0   Max.   :4900
##  NA's   :4            NA's   :4        NA's   :4
```

```
summary(hourly_activity)
```

```
##        Id            ActivityHour          Calories      TotalIntensity
##  Min.   :1.504e+09   Length:22099       Min.   : 42.00   Min.   :  0.00
##  1st Qu.:2.320e+09   Class :character   1st Qu.: 63.00   1st Qu.:  0.00
##  Median :4.445e+09   Mode  :character   Median : 83.00   Median :  3.00
##  Mean   :4.848e+09                      Mean   : 97.39   Mean   : 12.04
##  3rd Qu.:6.962e+09                      3rd Qu.:108.00   3rd Qu.: 16.00
##  Max.   :8.878e+09                      Max.   :948.00   Max.   :180.00
##  AverageIntensity   StepTotal
##  Min.   :0.0000   Min.   :    0.0
##  1st Qu.:0.0000   1st Qu.:    0.0
##  Median :0.0500   Median :   40.0
##  Mean   :0.2006   Mean   :  320.2
##  3rd Qu.:0.2667   3rd Qu.:  357.0
##  Max.   :3.0000   Max.   :10554.0
```

Right off the bat, a few observations stand out in the daily_activity summary. Among the 30 users that recorded their sleep, the average sleep length is a little less than 7 hours. Considering that the ideal amount is around 7-9 hours, the users are lacking. Additionally, the average BMI is 25.19, which is 0.2 higher than the highest BMI within the healthy range. We also see that a large majority of the users time is spent sedentary, considering the average of around 16 hours.

In the hourly_activity summary, there appears to be a large range in the StepTotal column. This could possibly mean the users are inconsistent with their movement, or the difference between each user is very drastic.

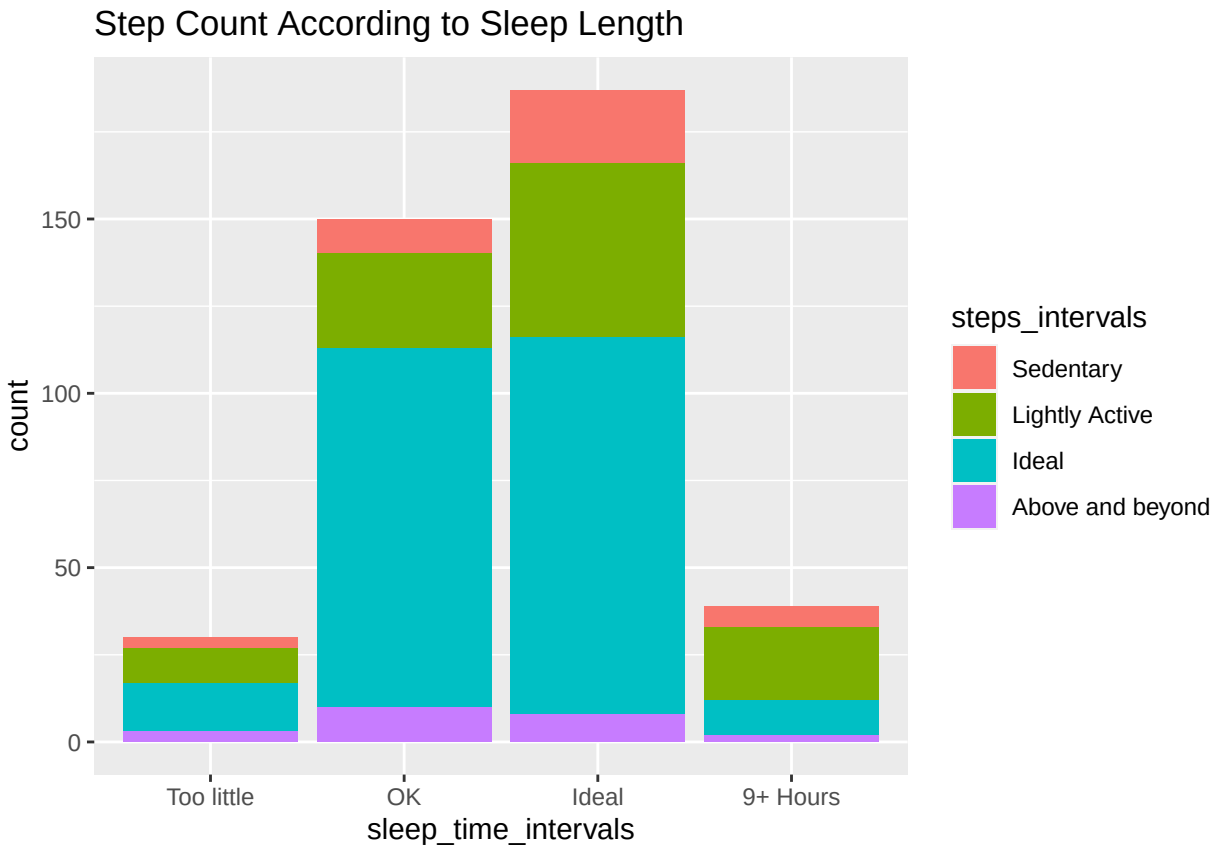
Using these observations, we can formulate some questions that will guide our analysis:

- Does heavier activity level correlate with amount of time asleep?
- Do BMI and total steps have a correlation?
- What does the step count look like for a user with a healthy BMI vs a user with an unhealthy BMI?
- Is there a specific time of the day that people are most/least active?

Data Visualization

Sleep Time Intervals vs Step Intervals

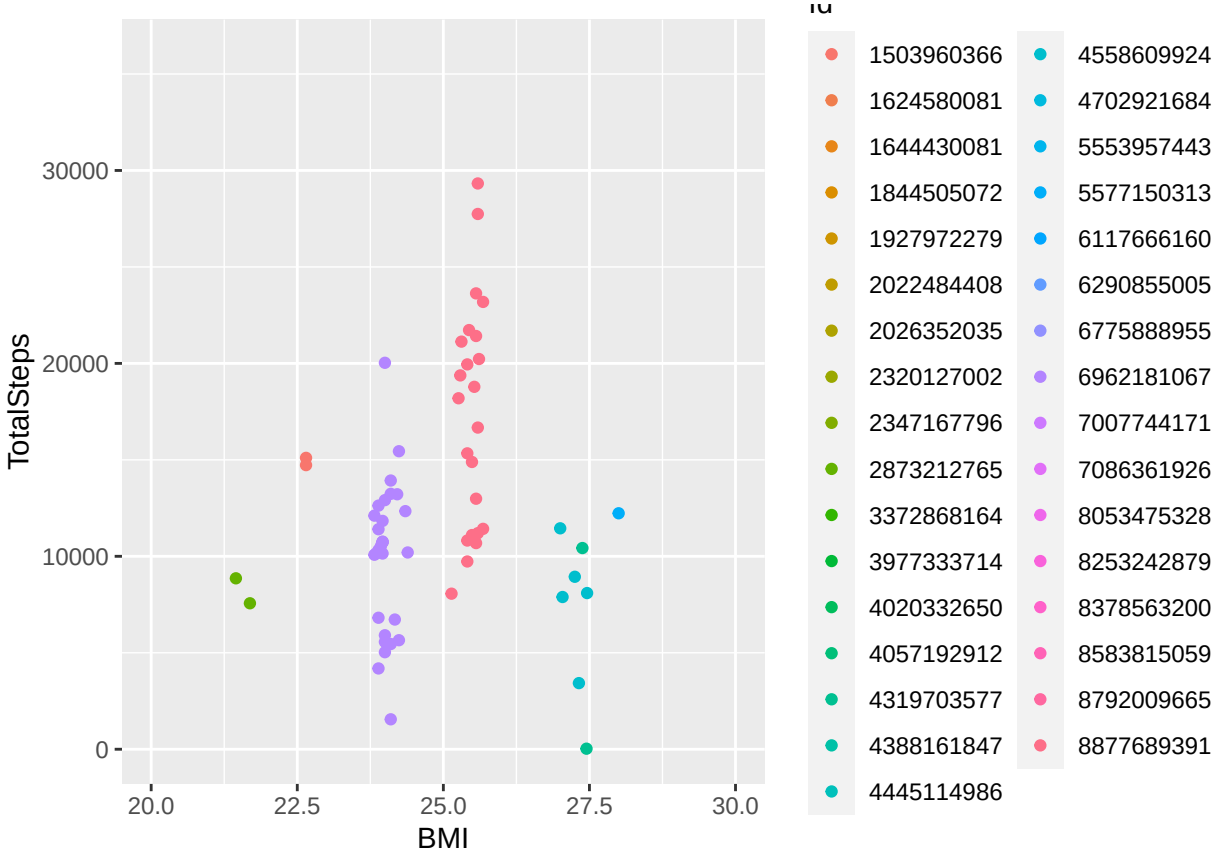
```
daily_activity %>%
  drop_na(c(TotalMinutesAsleep, TotalSteps)) %>%
  mutate(sleep_time_intervals=cut(TotalMinutesAsleep, breaks=c(0,240,420,540,800),
                                  labels=c('Too little','OK','Ideal','9+ Hours'))) %>%
  mutate(steps_intervals=cut(TotalSteps, breaks=c(0,3000,7000,15000,30000),
                             labels=c('Sedentary','Lightly Active','Ideal','Above and beyond'))) %>%
  ggplot() +
  geom_bar(aes(x=sleep_time_intervals, fill=steps_intervals)) +
  labs(title='Step Count According to Sleep Length')
```



Based on our graph, we can see that a greater proportion of those who had an OK or ideal sleep took an ideal amount of steps during the day. Using this data, we can advise Bellabeat to add a sleep coach feature on the Time product to assist users in getting a better sleep.

BMI vs Total Steps

```
daily_activity$Id <- factor(daily_activity$Id)

daily_activity %>%
  ggplot() +
  geom_point(aes(x=BMI, y=TotalSteps, color=Id)) +
  scale_color_discrete() +
  xlim(20,30)
```

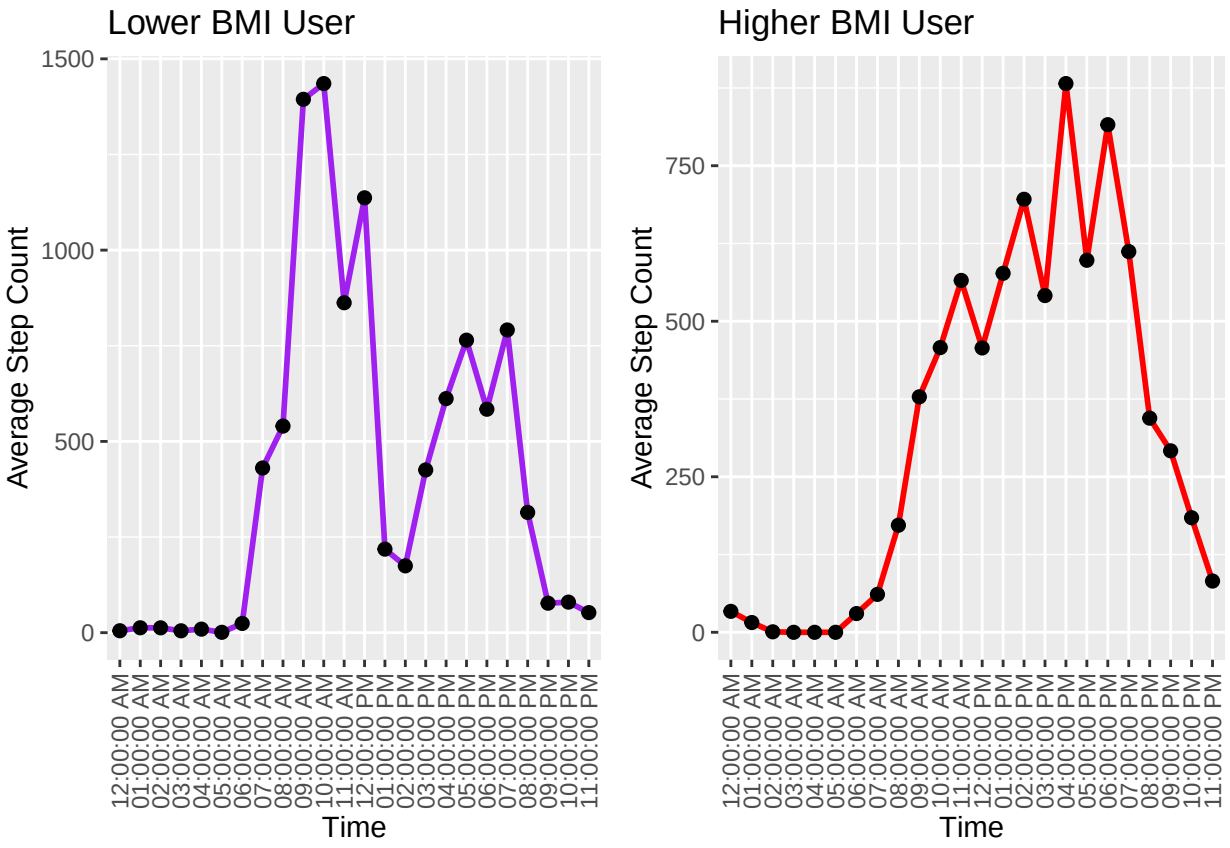


Considering we have such little data on BMI, we cannot see if there is much correlation between BMI and total steps. However, we can see data on each individual. Knowing that the healthy BMI range is from 18.5-24.9, we see an individual who has a lower step count and high BMI. We also see an individual within the healthy BMI range who has a lot of daily steps. Let's analyze the hourly activity of these two individuals to see if there are reasons behind their measurements.

Lower BMI User vs Higher BMI User

```
## lower bmi user

lower_bmi <- hourly_activity[hourly_activity$Id==6962181067, ]
lower_bmi$ActivityHour <- as.POSIXct(lower_bmi$ActivityHour, format = "%m/%d/%Y %I:%M:%S %p")
lower_bmi$Time <- format(lower_bmi$ActivityHour, "%I:%M:%S %p")

# order the time column
lower_bmi$Time <- factor(lower_bmi$Time, levels = unique(lower_bmi$Time[order(strptime(lower_bmi$Time,

lower_bmi_plot <- lower_bmi %>%
  ggplot(aes(x=Time, y=StepTotal)) +
  stat_summary(fun = "mean", geom = "line", aes(group = 1), color = 'purple', size = 1) +
  stat_summary(fun = "mean", geom = "point", size = 2) +
  theme(axis.text.x = element_text(angle=90, vjust=.5, hjust=1)) +
  labs(x = "Time", y = "Average Step Count", title = "Lower BMI User")


## higher bmi user
```

```
higher_bmi <- hourly_activity[hourly_activity$Id==4558609924, ]
higher_bmi$ActivityHour <- as.POSIXct(higher_bmi$ActivityHour, format = "%m/%d/%Y %I:%M:%S %p")
higher_bmi$Time <- format(higher_bmi$ActivityHour, "%I:%M:%S %p")

# order the time column
higher_bmi$Time <- factor(higher_bmi$Time, levels = unique(higher_bmi$Time[order(strptime(higher_bmi$Ti

higher_bmi_plot <- higher_bmi %>%
  ggplot(aes(x=Time, y=StepTotal)) +
  stat_summary(fun = "mean", geom = "line", aes(group = 1), color = 'red', size = 1) +
  stat_summary(fun = "mean", geom = "point", size = 2) +
  theme(axis.text.x = element_text(angle=90, vjust=.5, hjust=1)) +
  labs(x = "Time", y = "Average Step Count", title = "Higher BMI User")

# plot both on same window
ggarrange(lower_bmi_plot, higher_bmi_plot)
```

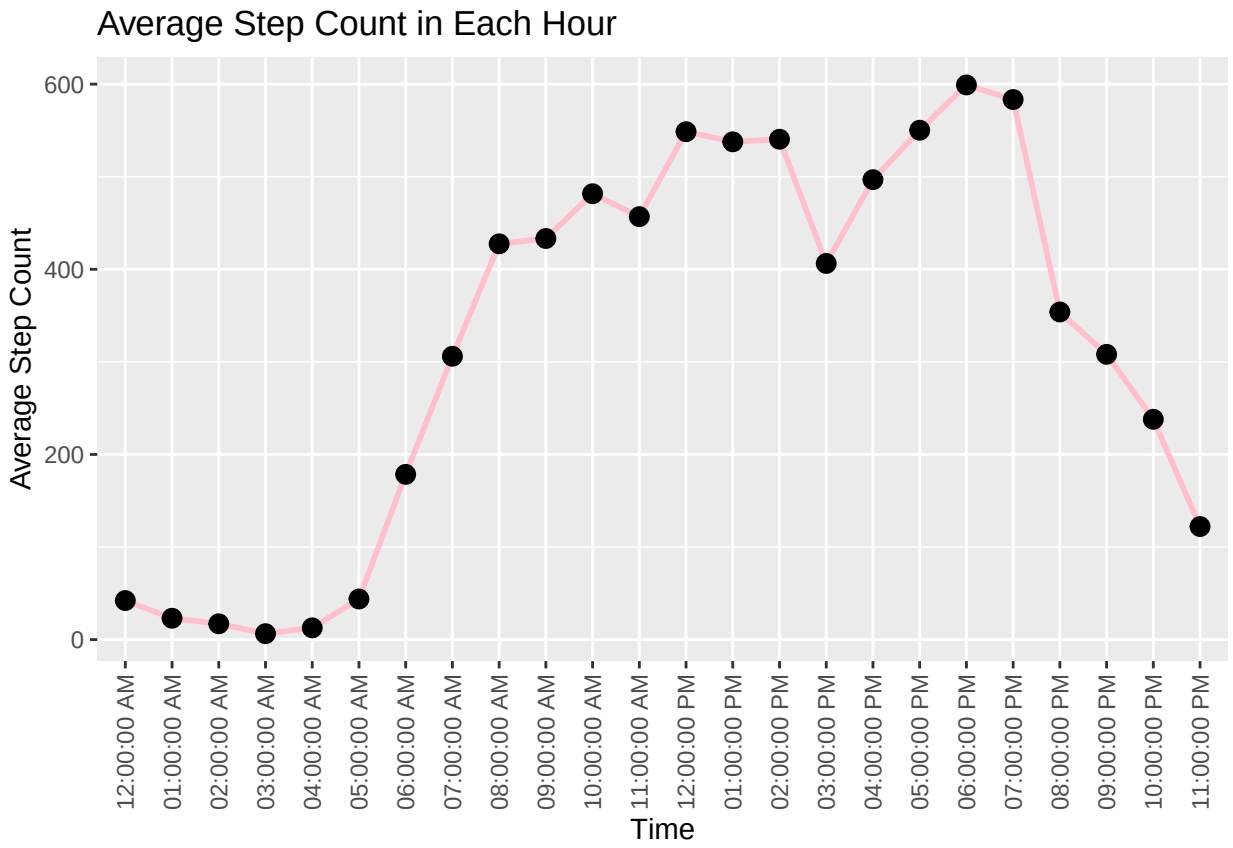


As we can see from our two plots, the user with a healthy BMI tends to get more steps in from 9 AM - 12 PM, while the user with the higher BMI gets more steps in from 2 PM - 6 PM. Since they are just two people, we cannot make any definite assumptions. However, if it is the case that moving more in the earlier part of the day is correlated with a healthier BMI, then we can advise Bellabeat to add a feature that analyzes users data to recommend when they should perform their workouts.

Average Step Count Per Hour

```
new_df <- subset(hourly_activity, select=c(ActivityHour, StepTotal)) %>%
  group_by(ActivityHour)
```

```r
new_df$ActivityHour <- as.POSIXct(new_df$ActivityHour, format = "%m/%d/%Y %I:%M:%S %p")

# average step count per hour
new_df$Time <- format(new_df$ActivityHour, "%I:%M:%S %p")
new_df$Time <- factor(new_df$Time, levels = unique(new_df$Time[order(strptime(new_df$Time, "%I:%M:%S %p

new_df %>%
  ggplot(aes(x=Time, y=StepTotal)) +
  stat_summary(fun = "mean", geom = "line", aes(group = 1), color = 'pink', size = 1) +
  stat_summary(fun = "mean", geom = "point", size = 3) +
  theme(axis.text.x = element_text(angle=90, vjust=.5, hjust=1)) +
  labs(x = "Time", y = "Average Step Count", title = "Average Step Count in Each Hour")
```



According to our plot, the most average steps are taken between 8 AM and 7 PM, which is reasonable. There seems to be a dip at 3 PM. Since there are only 30 users in this dataset, we can't assume the majority of people are less active at 3 PM. However, we can use these results to assume there are certain times in the day where people tend not to move as much.

Thus, in order to improve Bellabeat's time product, there could be a feature that reminds users to be active if they are still for too long. That way, the smartwatch can help encourage balanced movement throughout the day.

Conclusion

In this case study, we used two main data frames: daily_activity and hourly_activity. The former consists of the user's sleep records, weight, step count, calories burned, and detailed activity measurements. The

latter consisted of hourly records of calories burned, steps, and intensity measurements. Using this data, we identified relationships between variables in order to inform Bellabeat's product strategy. We found relationships through data visualization, such as step count and time asleep. We also looked deeper into individual data to determine why one user may have a lower BMI than the other despite a similar activity level. Of course, there are limitations because many factors outside of our data, such as food intake, could affect BMI. With the data we have, we found a pattern of the lower BMI user moving around more in the earlier half of their day, while the higher BMI user moves around more in the later half of their day.

Time Product Feature Recommendations

- A digital sleep coach that suggests tips on how to achieve better sleep based on the user's daily activity and previous sleep data
- A digital health coach that recommends the best times to be active according to the user's previous patterns and BMI
- A reminder feature that alerts the user to move around when they remain sedentary for too long

Next steps

Due to our limited data, it is difficult to draw certain conclusions. Thus, I would like to conduct research on a larger pool of users to test my theories from this case study. I would like to see if the time users are active during the day has any effect on their health. Additionally, I'd want to compare Fitbit user data to another smart watch company's data to see if there are patterns amongst the two.