How can a wellness company, Bellabeat, play it smart?

A case study in smart device's consumer usage

By Leonie Nutz

01/20/22

Introduction - Ask

In this case study, I will perform data analysis for Bellabeat, a high-tech manufacturer of health-focused products for women. Bellabeat is a successful small company, but they have the potential to become a larger player in the global smart device market. I focused my analysis on one of Bellabeat's products, the Bellabeat app, which provides users with health data related to their activity, sleep, stress, menstrual cycle, and mindfulness habits. The Bellabeat app connects to their line of smart wellness products and accumulates data through those products.

I will analyze smart device data to gain insights into how consumers are using their smart devices. My analysis will help guide future marketing strategies for the Bellabeat team. With the future adjustments made to the marketing strategy, we will ideally see an increase in sales and a higher consumer interest in Bellabeat products than in other smart devices.

The goal of the analysis is to answer the following questions:

- 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?

The key stakeholders for this analysis are:

- **Urška Sršen:** Bellabeat's cofounder and Chief Creative Officer
- **Sando Mur:** Mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team
- **Bellabeat marketing analytics team:** A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy.

Prepare

The dataset used for this analysis is a crowd-sourced public dataset called Fitbit Fitness Tracker Data made available through Mobius. It contains data from thirty eligible Fitbit users that consented to share their personal tracker data. The dataset contains minute-level output for physical activity, heart rate and sleep monitoring. It includes information about daily activity, steps, and heart-rate. The data was collected from March 12th 2016 to May 12th 2016 (total of 61 days).

The data is organized into 18 separate csv files, one of them being all the data merged together. The data is mostly in a long format and the merged daily activity file contains 15 columns and 941 rows. Next, I wanted to answer the following question: Does this data

- **R** eliable
- **O** riginal
- **C** omprehensive
- **C** urrent
- **C** ited

The dataset only contains data from 30 participants, which makes its reliability quite low. Furthermore, the data was collected by a third-party provider, Amazon Mechanical Turk, which makes its originality very low. Next, the data is somewhat comprehensive, seen as the data that was collected mostly matches the data that the Bellabeat products collect. Since the data was collected in 2016, the data is nearly 6 years old, decreasing its relevance significantly. Lastly, the data is not cited, since it was collected from a third-party provider. Concluding, the dataset lacks reliability, originality, is not current and not cited in depth. However, the stakeholders wish for the analysis to be done with this specific dataset.

After assessing the data, I decided to shift my main focus to the dailyActivity_merged.csv and weightLogInfo_merged.csv files. To sort and filter the data I used Microsoft Excel.

First, I inspected the weightLogInfo_merged.csv, which was especially interesting to me, since the Bellabeat app does not support the feature of tracking the consumer's weight. I was interested in how much this feature was utilized.

Count of WeightKg	Column Labels		
	Apr	May	Grand Total
ID			
1503960366		2	2
1927972279	1		1
2873212765	1	1	2
4319703577	1	1	2
4558609924	2	3	5
5577150313	1		1
6962181067	18	12	30
8877689391	16	8	24
Grand Total	**40**	**27**	**67**

Total users
8

After placing the data into a pivot table, it became clear that the weight log was a very unpopular feature among the 30 consumers. Only a total of 8 people used the feature at least once in the 61 days of data collection. A maximum of 1,830 logs could have been created during the collection period, however only a total of 67 were created. This means that only 3.66% of the possible weight logs were

Next, I inspected the dailyActivity_merged.csv file, containing data on total steps, total distance, duration of different activity levels (mins) and calories burned. After scrolling through the spreadsheet, I immediately noticed null values in the total steps column. Assuming that a consumer would not get through the day with taking 0 steps and burning 0 calories, I decided to filter out those rows. This decreased the dataset from 940 rows to 863 rows. I decided to clear the rows with 0 steps and then deleted the sheet rows. Next, I noticed that the TrackerDistance and TotalDistance column were duplicates of each other. I decided to discard the TrackerDistance column for this analysis. This decreased the number of columns to 14 in the dataset.

Process

Next, I will use R to check the data further for errors and clean the dataset in order to work with the data effectively.

Loading the dataset and all necessary packages

```
# import all relevant packages
library(ggplot2)
library(tidyverse)
library(skimr)
library(janitor)
library(lubridate)

# load the csv file
dailyactivity <- read.csv("C:\\Users\\lnutz\\Desktop\\Case Study Bellabeat\\Fitabase Data 4.12.16-5.12.16\\dailyActivity_merged.csv")
```

Check the datatypes of each column

```
# get an overview of current columns and their data types
glimpse(dailyactivity)
```

```
## Rows: 862
## Columns: 14
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <chr> "4/12/2016", "4/14/2016", "4/15/2016", "4/16/~
## $ TotalSteps               <int> 13162, 10460, 9762, 12669, 9705, 13019, 15506~
## $ TotalDistance            <dbl> 8.50, 6.74, 6.28, 8.16, 6.48, 8.59, 9.88, 6.6~
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance       <dbl> 1.88, 2.44, 2.14, 2.71, 3.19, 3.25, 3.53, 1.9~
## $ ModeratelyActiveDistance <dbl> 0.55, 0.40, 1.26, 0.41, 0.78, 0.64, 1.32, 0.4~
## $ LightActiveDistance      <dbl> 6.06, 3.91, 2.83, 5.04, 2.51, 4.71, 5.03, 4.2~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes        <int> 25, 30, 29, 36, 38, 42, 50, 28, 19, 66, 41, 3~
## $ FairlyActiveMinutes      <int> 13, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21, 5,~
## $ LightlyActiveMinutes     <int> 328, 181, 209, 221, 164, 233, 264, 205, 211, ~
## $ SedentaryMinutes         <int> 728, 1218, 726, 773, 539, 1149, 775, 818, 838~
## $ Calories                 <int> 1985, 1776, 1745, 1863, 1728, 1921, 2035, 178~
```

Immediately we can see that the ActivityDate is not in the correct data format. Using the lubridate package, the all the ActivityDate values are successfully changed to a date type.

```
# change char to date type
dailyactivity$ActivityDate <- mdy(dailyactivity$ActivityDate)

# verify again that the data type was successfully changed
glimpse(dailyactivity)
```

```
## Rows: 862
## Columns: 14
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <date> 2016-04-12, 2016-04-14, 2016-04-15, 2016-04-~
## $ TotalSteps               <int> 13162, 10460, 9762, 12669, 9705, 13019, 15506~
## $ TotalDistance            <dbl> 8.50, 6.74, 6.28, 8.16, 6.48, 8.59, 9.88, 6.6~
```

```
## $ LoggedActivitiesDistance      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance            <dbl> 1.88, 2.44, 2.14, 2.71, 3.19, 3.25, 3.53, 1.9~
## $ ModeratelyActiveDistance      <dbl> 0.55, 0.40, 1.26, 0.41, 0.78, 0.64, 1.32, 0.4~
## $ LightActiveDistance           <dbl> 6.06, 3.91, 2.83, 5.04, 2.51, 4.71, 5.03, 4.2~
## $ SedentaryActiveDistance       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes             <int> 25, 30, 29, 36, 38, 42, 50, 28, 19, 66, 41, 3~
## $ FairlyActiveMinutes           <int> 13, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21, 5,~
## $ LightlyActiveMinutes          <int> 328, 181, 209, 221, 164, 233, 264, 205, 211, ~
## $ SedentaryMinutes              <int> 728, 1218, 726, 773, 539, 1149, 775, 818, 838~
## $ Calories                      <int> 1985, 1776, 1745, 1863, 1728, 1921, 2035, 178~
```

Verify number of users that participated in data collection

According to the description of the data set, 30 users participated in the data collection process. To verify this, I will count the number of unique Ids created.

```
# count number of id
sum_id <- length(unique(dailyactivity$Id))
print(sum_id)

## [1] 33
```

There exist 33 Ids, which might suggest that 3 users created new accounts/Ids during the data collection period. Next, I will count how often each Id occurs.

```
# see number of occurrences per id
table(dailyactivity$Id)

##
## 1503960366 1624580081 1644430081 1844505072 1927972279 2022484408 2026352035
##         29         31         30         21         17         31         31
## 2320127002 2347167796 2873212765 3372868164 3977333714 4020332650 4057192912
##         31         18         31         20         30         17          3
## 4319703577 4388161847 4445114986 4558609924 4702921684 5553957443 5577150313
##         31         31         31         31         30         31         28
## 6117666160 6290855005 6775888955 6962181067 7007744171 7086361926 8053475328
##         23         24         17         31         24         30         31
## 8253242879 8378563200 8583815059 8792009665 8877689391
##         18         31         30         19         31
```

Only one Id has a very low number of occurrence (3). Since there is no Id with an occurrence of less than 2, I decided to not discard any values.

Create new column with dates as weekdays

```
# new column with weekdays
dailyactivity$Weekday <- format(as.Date(dailyactivity$ActivityDate), format = "%A")
glimpse(dailyactivity)

## Rows: 862
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <date> 2016-04-12, 2016-04-14, 2016-04-15, 2016-04-~
## $ TotalSteps               <int> 13162, 10460, 9762, 12669, 9705, 13019, 15506~
## $ TotalDistance            <dbl> 8.50, 6.74, 6.28, 8.16, 6.48, 8.59, 9.88, 6.6~
## $ LoggedActivitiesDistance      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
```

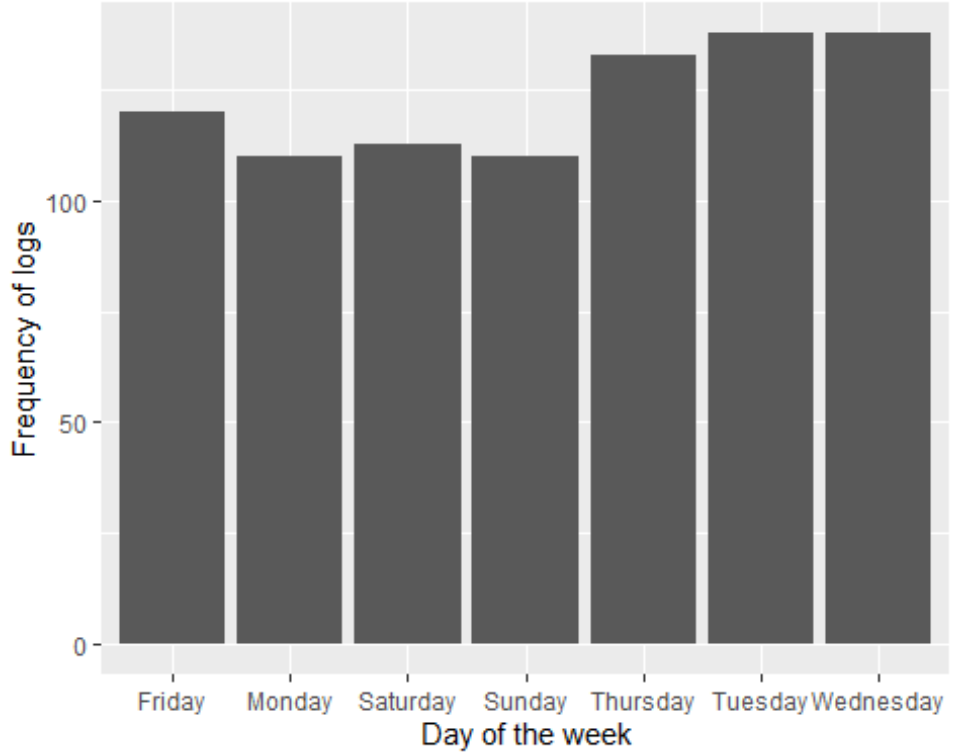
```
## $ VeryActiveDistance          <dbl> 1.88, 2.44, 2.14, 2.71, 3.19, 3.25, 3.53, 1.9~
## $ ModeratelyActiveDistance    <dbl> 0.55, 0.40, 1.26, 0.41, 0.78, 0.64, 1.32, 0.4~
## $ LightActiveDistance         <dbl> 6.06, 3.91, 2.83, 5.04, 2.51, 4.71, 5.03, 4.2~
## $ SedentaryActiveDistance     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes           <int> 25, 30, 29, 36, 38, 42, 50, 28, 19, 66, 41, 3~
## $ FairlyActiveMinutes         <int> 13, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21, 5,~
## $ LightlyActiveMinutes        <int> 328, 181, 209, 221, 164, 233, 264, 205, 211, ~
## $ SedentaryMinutes            <int> 728, 1218, 726, 773, 539, 1149, 775, 818, 838~
## $ Calories                    <int> 1985, 1776, 1745, 1863, 1728, 1921, 2035, 178~
## $ Weekday                     <chr> "Tuesday", "Thursday", "Friday", "Saturday",~
```

Analyze

Next, I will use R again to analyze the data, identify trends and patterns, as well as create some visualizations.

Frequency of logs per weekday

```
ggplot(data = dailyactivity, mapping = aes(x = DayOfWeek)) + geom_bar() + labs(x='Day of the week', y='Frequency of logs')
```



We can immediately see that Sunday, Monday and Saturday are the days where people track their data the least and Tuesday and Wednesday are the days they track more frequently.

Average and median Steps taken in relationship to the weekday

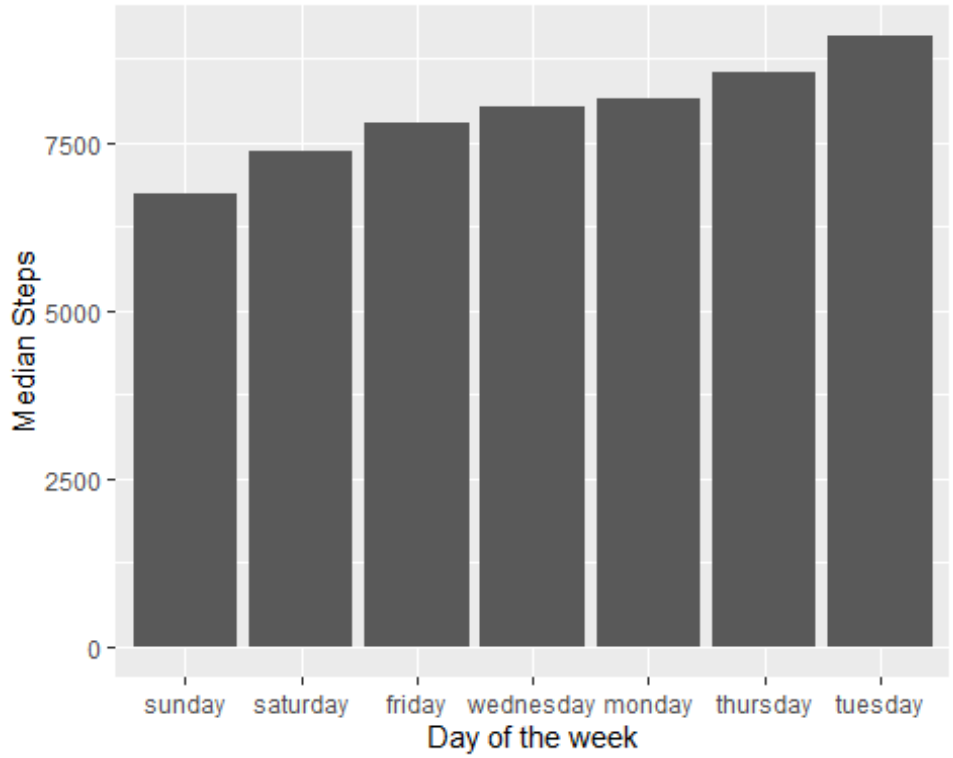
```
# average and median steps Monday
filter_monday <- filter(dailyactivity, DayOfWeek == "Monday")
monday_avg <- mean(filter_monday$TotalSteps)
monday_median <- median(filter_monday$TotalSteps)
sprintf("Monday average steps: %d", round(monday_avg, digits = 0))

## [1] "Monday average steps: 8488"

sprintf("Monday average steps: %d", round(monday_median, digits = 0))

## [1] "Monday average steps: 8164"

# average and median steps Tuesday
filter_tuesday <- filter(dailyactivity, DayOfWeek == "Tuesday")
tuesday_avg <- mean(filter_tuesday$TotalSteps)
tuesday_median <- median(filter_tuesday$TotalSteps)
sprintf("Tuesday average steps: %d", round(tuesday_avg, digits = 0))
```

```
## [1] "Tuesday average steps: 8949"

sprintf("Tuesday average steps: %d", round(tuesday_median, digits = 0))

## [1] "Tuesday average steps: 9090"

# average and median steps Wednesday
filter_wednesday <- filter(dailyactivity, DayOfWeek == "Wednesday")
wednesday_avg <- mean(filter_wednesday$TotalSteps)
wednesday_median <- median(filter_wednesday$TotalSteps)
sprintf("Wednesday average steps: %d", round(wednesday_avg, digits = 0))

## [1] "Wednesday average steps: 8139"

sprintf("Wednesday average steps: %d", round(wednesday_median, digits = 0))

## [1] "Wednesday average steps: 8027"

# average and median steps Thursday
filter_thursday <- filter(dailyactivity, DayOfWeek == "Thursday")
thursday_avg <- mean(filter_thursday$TotalSteps)
thursday_median <- median(filter_thursday$TotalSteps)
sprintf("Thursday average steps: %d", round(thursday_avg, digits = 0))

## [1] "Thursday average steps: 8185"

sprintf("Thursday average steps: %d", round(thursday_median, digits = 0))

## [1] "Thursday average steps: 8538"

# average and median steps Friday
filter_friday <- filter(dailyactivity, DayOfWeek == "Friday")
friday_avg <- mean(filter_friday$TotalSteps)
friday_median <- median(filter_friday$TotalSteps)
sprintf("Friday average steps: %d", round(friday_avg, digits = 0))

## [1] "Friday average steps: 7821"

sprintf("Friday average steps: %d", round(friday_median, digits = 0))

## [1] "Friday average steps: 7800"

# average and median steps Saturday
filter_saturday <- filter(dailyactivity, DayOfWeek == "Saturday")
saturday_avg <- mean(filter_saturday$TotalSteps)
saturday_median <- median(filter_saturday$TotalSteps)
sprintf("Saturday average steps: %d", round(saturday_avg, digits = 0))

## [1] "Saturday average steps: 8947"

sprintf("Saturday average steps: %d", round(saturday_median, digits = 0))

## [1] "Saturday average steps: 7379"

# average and median steps Sunday
filter_sunday <- filter(dailyactivity, DayOfWeek == "Sunday")
sunday_avg <- mean(filter_sunday$TotalSteps)
```

```
sunday_median <- median(filter_sunday$TotalSteps)
sprintf("Sunday average steps: %d", round(sunday_avg, digits = 0))

## [1] "Sunday average steps: 7627"

sprintf("Sunday average steps: %d", round(sunday_median, digits = 0))
```

Since there is such a big discrepancy between the median and average steps taken on Saturday and Sunday, there is reason to suspect that high and low anomalous results are affecting the average too much. Therefore, the median is more suited for the further analysis.

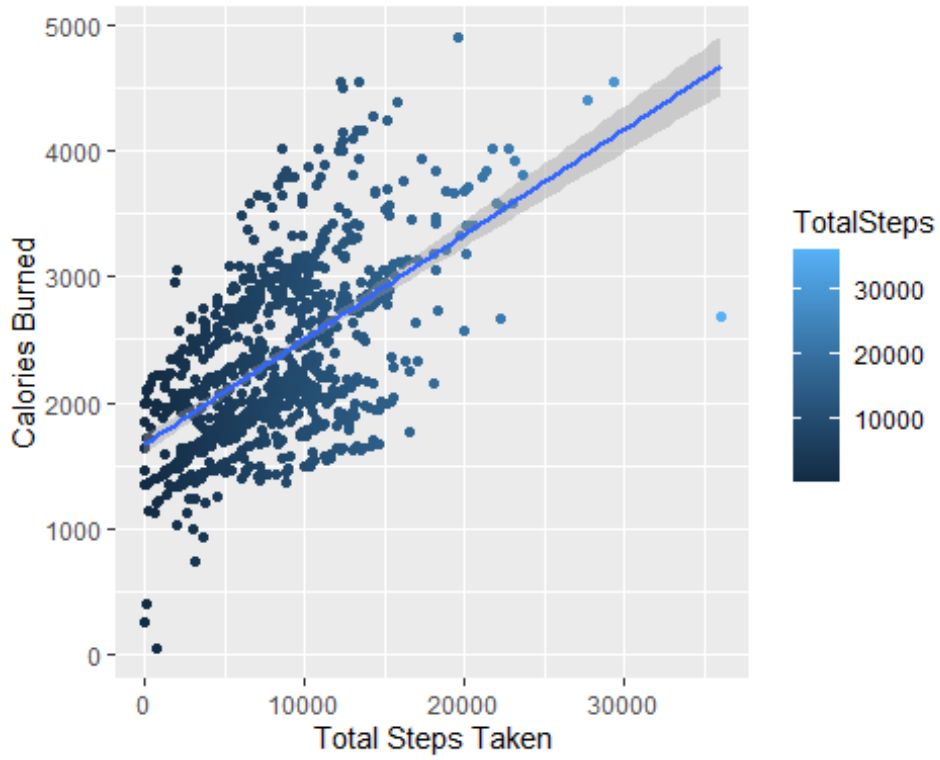
```
# create new data frame with newly calculated data
days <- c("monday", "tuesday", "wednesday", "thursday", "friday", "saturday", "sunday")
median_steps <- c(monday_median, tuesday_median, wednesday_median, thursday_median, friday
_median, saturday_median, sunday_median)
df <- data.frame(days, median_steps)
# plot avg  steps per day
ggplot(data = df, mapping = aes(x = reorder(days, +median_steps), y = median_steps)) + geom_col() +
labs(x='Day of the week', y='Median Steps')
```



Tuesday and Thursday are the days with the most average steps taken, whereas Saturday and Sunday are the days with the least average steps taken.

Is there a correlation between calories and total steps taken?

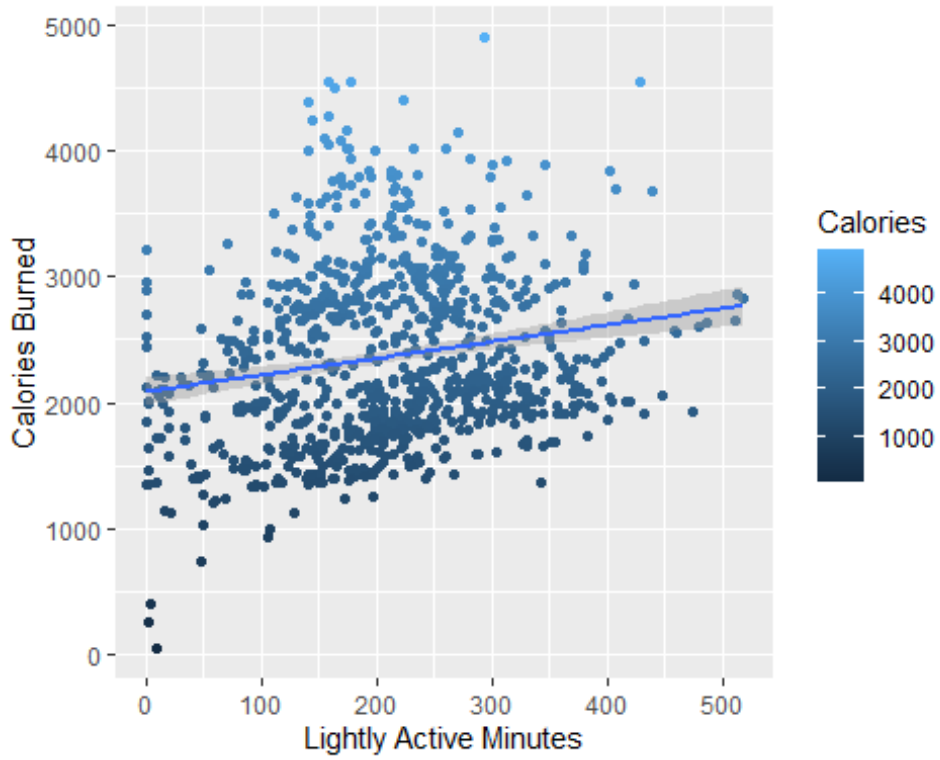
```
ggplot(data = dailyactivity, mapping = aes(x = TotalSteps, y = Calories, color = TotalSteps)) + geom_po
int() + geom_smooth(method = lm) + labs(x='Total Steps Taken', y='Calories Burned')
```



We can see that there is a positive correlation between calories burned and total steps taken. We have some outliers at the 5000-calorie area and 0 calorie area. The outliers may have been due to people taking off their fitbit during the day and never putting it back on or errors in the data collection. Concluding, the more steps that were taken, the more calories were burned.

Is there a correlation between light activity and calories burned?

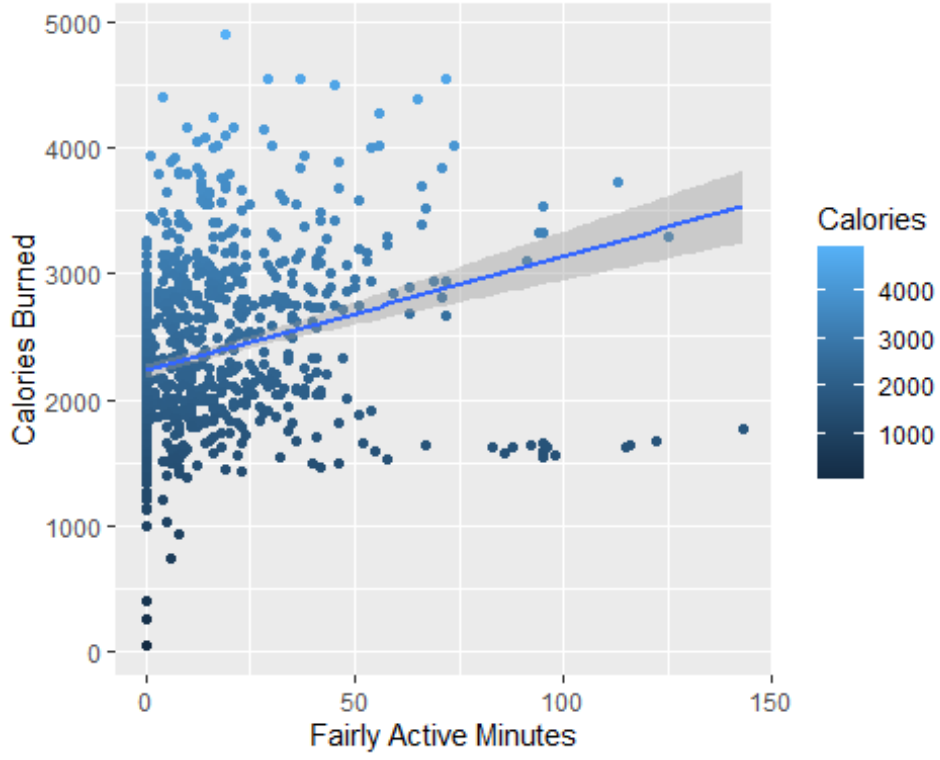
```
ggplot(data = dailyactivity, mapping = aes(x = LightlyActiveMinutes, y = Calories, color = Calories)) + g
eom_point() + geom_smooth(method = lm) +
labs(x='Lightly Active Minutes', y='Calories Burned')
```



We can see some potential outliers in the 5000-calorie area and some in the 0-calorie area. All in all, there is no clear correlation between the calories burned and lightly active minutes.

Is there a correlation between moderate activity and calories burned?

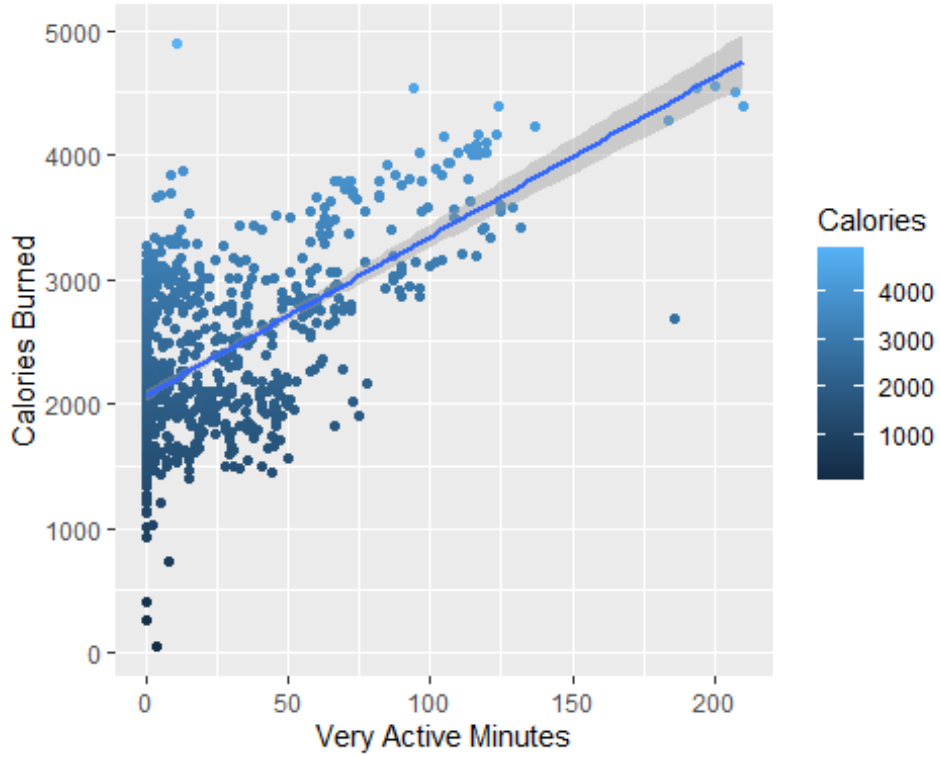
```
ggplot(data = dailyactivity, mapping = aes(x = FairlyActiveMinutes, y = Calories, color = Calories)) + ge
om_point() + geom_smooth(method = lm) +
labs(x='Fairly Active Minutes', y='Calories Burned')
```



There is no clear correlation between fairly active minutes and calories burned. The trend line indicates a positive correlation; however, it is not very clear. Again, there are some outliers visible in the 5000-calorie area and the 0-calorie area.

Is there a correlation between high activity and calories burned?

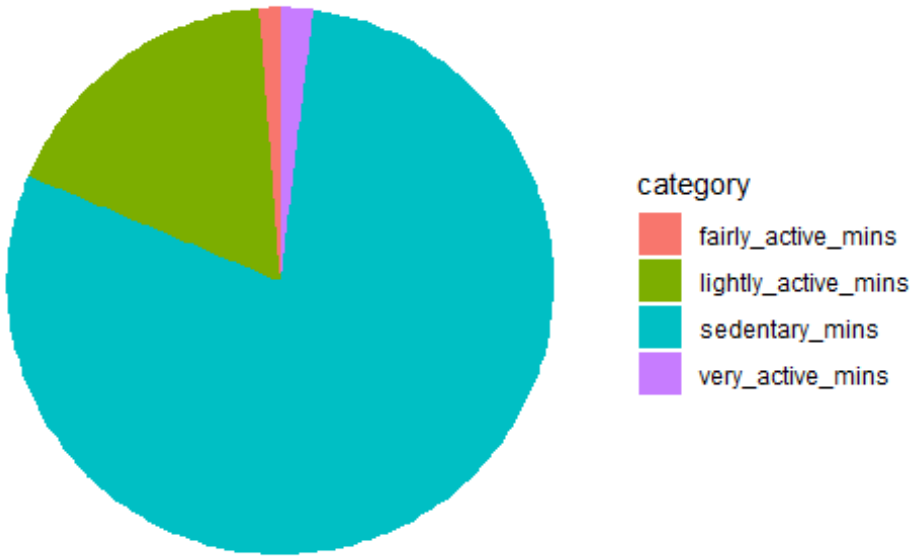
```
ggplot(data = dailyactivity, mapping = aes(x = VeryActiveMinutes, y = Calories, color = Calories)) + geo
m_point() + geom_smooth(method = lm) +
labs(x='Very Active Minutes', y='Calories Burned')
```



We can see there is a positive correlation between very active minutes and calories burned. Again, there are some outliers visible in the 5000-calorie area and the 0-calorie area. Concluding, the higher the number of active minutes, the more calories were burned.

Different Activity Levels

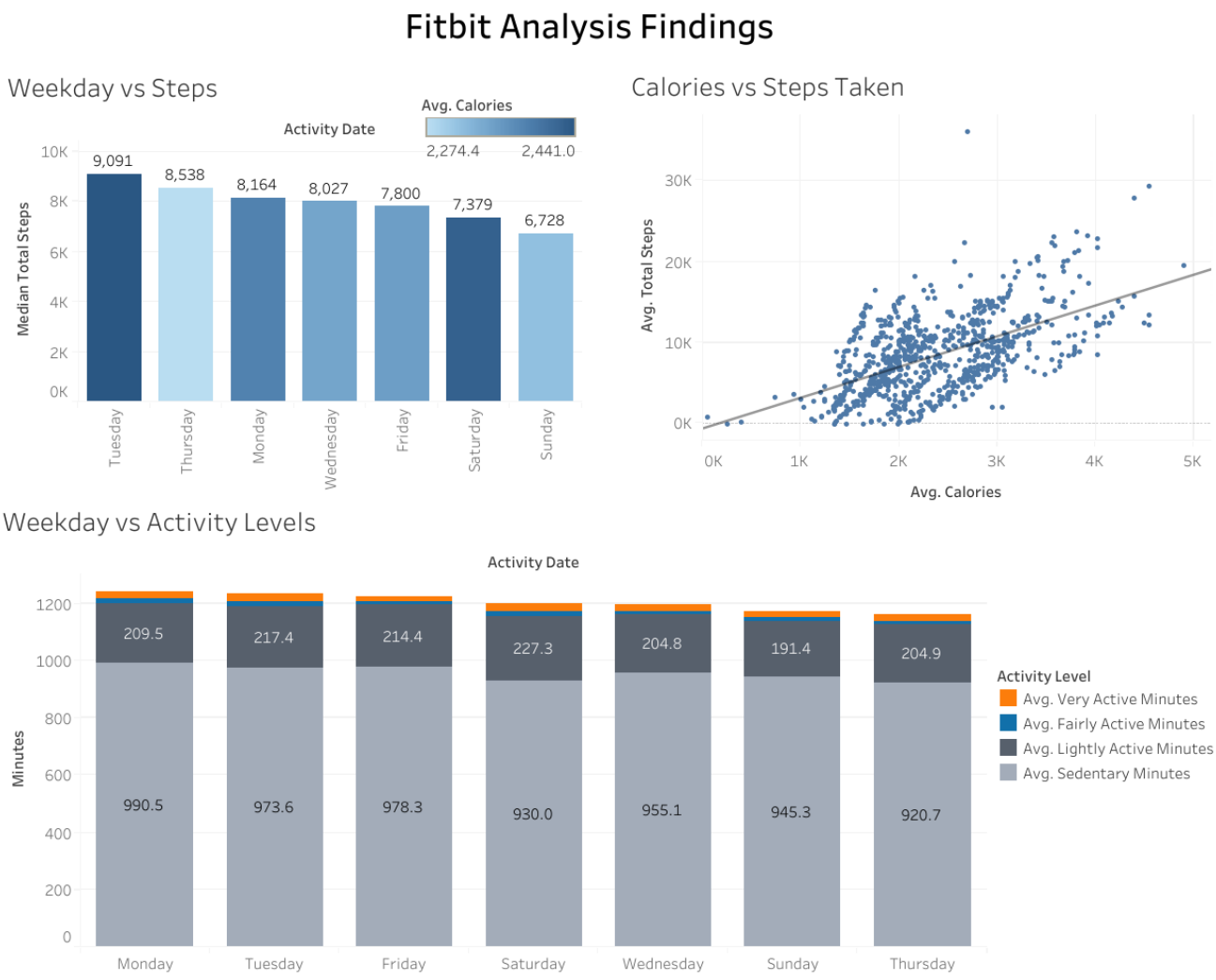
```
very_active_mins <- sum(dailyactivity$VeryActiveMinutes)
fairly_active_mins <- sum(dailyactivity$FairlyActiveMinutes)
lightly_active_mins <- sum(dailyactivity$LightlyActiveMinutes)
sedentary_mins <- sum(dailyactivity$SedentaryMinutes)

sums <- c(very_active_mins, fairly_active_mins, lightly_active_mins, sedentary_mins)
category <- c("very_active_mins", "fairly_active_mins", "lightly_active_mins", "sedentary_mins")
df2 <- data.frame(sums, category)

#create pie chart
ggplot(data = df2, aes(x="", y=sums, fill=category)) +
 geom_bar(stat="identity", width=1) +
 coord_polar("y", start=0) +
 theme_void()
```



We can see that the majority of the tracked activity minutes are sedentary minutes (79%). Around 17% are made up from the lightly active minutes. Very little of the total tracked active minutes stems from fairly active (1.9%) and very active minutes (1.2%).

Share

I summarized my findings from my analysis into a Tableau dashboard.
https://public.tableau.com/app/profile/leonie.nutz/viz/FitbitAnalysis_16430382368950/Dashboard1



These three data plots are the most insightful and will help influence Bellabeat's marketing strategy. The specific suggestions will be discussed in the next section of this report.

According to the CDC, people aged 18-64 should at least do 150 minutes of moderate-intensity aerobic activity. Summing up the average fairly active and very active minutes per day, we get an average of 37 minutes. So, the average for the week of moderate activity would be 259 minutes.

According to Health Quote First, the average steps people aged 20-65 should take per day lies between 7,000-11,500 steps. The maximum median steps tracked from the Fitbit users were 9,091. The minimum median steps tracked were 6,728 steps.

It is difficult to give an estimate on the recommended number of calories one should burn per day, since it depends on the body composition, gender and personal weight goals (maintain/loose/gain). As a general guideline, women burn around 2,000 calories a day and men burn around 2,500 calories a day.

Act

What are some trends in smart device usage?
The type of activity level that was tracked the most were sedentary minutes (79%).
The more steps that were taken, the more calories were burned.
The least steps were taken during the weekend.
The frequency of tracking logs was the highest during the middle of the week.

How could these trends apply to Bellabeat customers?
Bellabeat customers are interested in their health and tracking their health-related data. They also track the number of steps they took, calories they burned and their active minutes. The trends and patterns discovered in the Fitbit customer's data apply to the Bellabeat customers because both user types have the same interest in health, wellness and fitness.

How could these trends help influence Bellabeat marketing strategy?
Bellabeat could market their products with regards to the global health and movement recommendations. The Fitbit users lie under the recommended steps, so a concern regarding our health could be presented as the problem and the Bellabeat products could be presented as the solution to the problem. To encourage users to walk more, the Bellabeat App could integrate a Daily Step Goal feature, which rewards the users with virtual badges every time they hit their step goal and also give extra rewards for hitting consecutive goals (streaks).

The frequency of logs of the Fitbit users is also extremely low. Only 3.66% of the possible logs were actually tracked in the data collection period. Daily reminders/push notifications from the Bellabeat App could remind the users to track their daily activity. Especially reminders on the weekend would be very beneficial for the users, because that is the time frame the lowest number of steps were recorded.

Lastly, Bellabeat could promote short workout videos for their customers on their app, to increase the number of very active minutes. Although combined with the fairly active minutes the average moderate intensity activity lies over the recommended amount, the number of very active minutes contributes the least to this.

References

https://www.cdc.gov/physicalactivity/basics/adults/index.htm

https://firstquotehealth.com/health-insurance-news/recommended-steps-day

https://www.nike.com/a/how-many-calories-should-you-burn-daily