Bellabeat Case Study Notebook

1. ASK

Purpose

The purpose of this is to hypothetically assist Bellabeat in adjusting their marketing strategy through the use of Fitbit data. My job was to look for trends in the data.

Fitbit data retrieved from here

Business Question

1. What are some trends in smart device usage?
2. How could these trends apply to Bellabeat customers?
3. How could these trends help influence Bellabeat marketing strategy?

Stakeholders

- 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.

2. PREPARE

Installing Packages

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- 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 errors
```

```
library(ggplot2)
```

Now that I have my packages installed and loaded, I need to upload the data and begin to explore it.

First, I explored each of the data through Kaggle and picked Daily Activity as the best option for a few reasons.

1. Not all of the data sets started at the same time which meant it would be better to dive into one deeply and then suppliment with other data if the need arose.
2. Not all of the data sets would be useful given what Bellabeat is already doing and not all of the data sets could be helpful given their names.
3. Daily Activity gave a more comprehensive amount of data that could make for a better analysis and for finding trends.

Though I picked Daily Activity, other data sets could be briefly analyzed later on, especially sleep.

Importing Data and Deeper Exploration

```
act1<-data.frame(read_csv("dailyActivity_merged1.csv"))
```

```
## Rows: 457 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.
```

```
act2<-data.frame(read_csv('dailyActivity_merged2.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.
```

```
colnames(act1)
```

```
##  [1] "Id"                       "ActivityDate"
##  [3] "TotalSteps"               "TotalDistance"
##  [5] "TrackerDistance"          "LoggedActivitiesDistance"
##  [7] "VeryActiveDistance"       "ModeratelyActiveDistance"
##  [9] "LightActiveDistance"      "SedentaryActiveDistance"
## [11] "VeryActiveMinutes"        "FairlyActiveMinutes"
## [13] "LightlyActiveMinutes"     "SedentaryMinutes"
## [15] "Calories"
```

```
colnames(act2) #Double checking that column names match
```

```
##  [1] "Id"                       "ActivityDate"
##  [3] "TotalSteps"               "TotalDistance"
##  [5] "TrackerDistance"          "LoggedActivitiesDistance"
##  [7] "VeryActiveDistance"       "ModeratelyActiveDistance"
##  [9] "LightActiveDistance"      "SedentaryActiveDistance"
## [11] "VeryActiveMinutes"        "FairlyActiveMinutes"
## [13] "LightlyActiveMinutes"     "SedentaryMinutes"
## [15] "Calories"
```

```
ID1<-unique(act1$"Id")
list(ID1)
```

```
## [[1]]
##  [1] 1503960366 1624580081 1644430081 1844505072 1927972279 2022484408
##  [7] 2026352035 2320127002 2347167796 2873212765 2891001357 3372868164
## [13] 3977333714 4020332650 4057192912 4319703577 4388161847 4445114986
## [19] 4558609924 4702921684 5553957443 5577150313 6117666160 6290855005
## [25] 6391747486 6775888955 6962181067 7007744171 7086361926 8053475328
## [31] 8253242879 8378563200 8583815059 8792009665 8877689391
```

```
ID2<-unique(act2$"Id") #creating a new name for the unique IDs
list(ID2)
```

```
## [[1]]
##  [1] 1503960366 1624580081 1644430081 1844505072 1927972279 2022484408
##  [7] 2026352035 2320127002 2347167796 2873212765 3372868164 3977333714
## [13] 4020332650 4057192912 4319703577 4388161847 4445114986 4558609924
## [19] 4702921684 5553957443 5577150313 6117666160 6290855005 6775888955
## [25] 6962181067 7007744171 7086361926 8053475328 8253242879 8378563200
## [31] 8583815059 8792009665 8877689391
```

During my deeper exploration, I noted that two of the ID numbers were not included in the second dataset (2891001357 and 6391747486) meaning that there were 33 instead of 35 unique identification numbers.

I left this alone and continued working while understanding that 2 people did not use their fitbit for the entire period.

```
act_bind <- rbind(act1, act2)

act_merged <- act_bind %>%
  arrange(Id, ActivityDate)

head(act_merged) #Viewing the headings to ensure it merged right
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366    3/25/2016      11004          7.11            7.11
## 2 1503960366    3/26/2016      17609         11.55           11.55
## 3 1503960366    3/27/2016      12736          8.53            8.53
## 4 1503960366    3/28/2016      13231          8.93            8.93
## 5 1503960366    3/29/2016      12041          7.85            7.85
## 6 1503960366    3/30/2016      10970          7.16            7.16
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0              2.57                     0.46
## 2                        0              6.92                     0.73
## 3                        0              4.66                     0.16
## 4                        0              3.19                     0.79
## 5                        0              2.16                     1.09
## 6                        0              2.36                     0.51
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                4.07                       0                33
## 2                3.91                       0                89
## 3                3.71                       0                56
## 4                4.95                       0                39
## 5                4.61                       0                28
## 6                4.29                       0                30
```

```
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  12                  205              804     1819
## 2                  17                  274              588     2154
## 3                   5                  268              605     1944
## 4                  20                  224             1080     1932
## 5                  28                  243              763     1886
## 6                  13                  223             1174     1820
```

```
str(act_merged) #Viewing the structure to make sure data types are correct
```

```
## 'data.frame':    1397 obs. of  15 variables:
##  $ Id                       : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate             : chr  "3/25/2016" "3/26/2016" "3/27/2016" "3/28/2016" ...
##  $ TotalSteps               : num  11004 17609 12736 13231 12041 ...
##  $ TotalDistance            : num  7.11 11.55 8.53 8.93 7.85 ...
##  $ TrackerDistance          : num  7.11 11.55 8.53 8.93 7.85 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance       : num  2.57 6.92 4.66 3.19 2.16 ...
##  $ ModeratelyActiveDistance: num  0.46 0.73 0.16 0.79 1.09 ...
##  $ LightActiveDistance      : num  4.07 3.91 3.71 4.95 4.61 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes        : num  33 89 56 39 28 30 33 47 44 26 ...
##  $ FairlyActiveMinutes      : num  12 17 5 20 28 13 12 21 13 14 ...
##  $ LightlyActiveMinutes     : num  205 274 268 224 243 223 239 200 168 216 ...
##  $ SedentaryMinutes         : num  804 588 605 1080 763 ...
##  $ Calories                 : num  1819 2154 1944 1932 1886 ...
```

```
glimpse(act_merged)
```

```
## Rows: 1,397
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <chr> "3/25/2016", "3/26/2016", "3/27/2016", "3/28/~
## $ TotalSteps               <dbl> 11004, 17609, 12736, 13231, 12041, 10970, 122~
## $ TotalDistance            <dbl> 7.11, 11.55, 8.53, 8.93, 7.85, 7.16, 7.86, 7.~
## $ TrackerDistance          <dbl> 7.11, 11.55, 8.53, 8.93, 7.85, 7.16, 7.86, 7.~
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance       <dbl> 2.57, 6.92, 4.66, 3.19, 2.16, 2.36, 2.29, 3.3~
## $ ModeratelyActiveDistance <dbl> 0.46, 0.73, 0.16, 0.79, 1.09, 0.51, 0.49, 0.8~
## $ LightActiveDistance      <dbl> 4.07, 3.91, 3.71, 4.95, 4.61, 4.29, 5.04, 3.6~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes        <dbl> 33, 89, 56, 39, 28, 30, 33, 47, 44, 26, 0, 25~
## $ FairlyActiveMinutes      <dbl> 12, 17, 5, 20, 28, 13, 12, 21, 13, 14, 0, 13,~
## $ LightlyActiveMinutes     <dbl> 205, 274, 268, 224, 243, 223, 239, 200, 168, ~
## $ SedentaryMinutes         <dbl> 804, 588, 605, 1080, 763, 1174, 820, 866, 737~
## $ Calories                 <dbl> 1819, 2154, 1944, 1932, 1886, 1820, 1889, 186~
```

Data Cleaning

Through glimpse and str, I was able to determine that "ActivityDate" was set to "character" rather than date/time, so I needed to fix that through data cleaning and manipulation. I started there and went through the rest of my cleaning process.

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

str(act_merged)
```

```
## 'data.frame':    1397 obs. of  15 variables:
##  $ Id                       : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate             : Date, format: "2016-03-25" "2016-03-26" ...
##  $ TotalSteps               : num  11004 17609 12736 13231 12041 ...
##  $ TotalDistance            : num  7.11 11.55 8.53 8.93 7.85 ...
##  $ TrackerDistance          : num  7.11 11.55 8.53 8.93 7.85 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance       : num  2.57 6.92 4.66 3.19 2.16 ...
##  $ ModeratelyActiveDistance: num  0.46 0.73 0.16 0.79 1.09 ...
##  $ LightActiveDistance      : num  4.07 3.91 3.71 4.95 4.61 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes        : num  33 89 56 39 28 30 33 47 44 26 ...
##  $ FairlyActiveMinutes      : num  12 17 5 20 28 13 12 21 13 14 ...
##  $ LightlyActiveMinutes     : num  205 274 268 224 243 223 239 200 168 216 ...
##  $ SedentaryMinutes         : num  804 588 605 1080 763 ...
##  $ Calories                 : num  1819 2154 1944 1932 1886 ...
```

```
#to double check that it was changed appropriately and to see if there is anything else mistaken
```

```
# Count missing values in each column
colSums(is.na(act_merged)) # gave me 0
```

```
##                       Id             ActivityDate                TotalSteps
##                        0                        0                         0
##            TotalDistance          TrackerDistance LoggedActivitiesDistance
##                        0                        0                         0
##       VeryActiveDistance ModeratelyActiveDistance      LightActiveDistance
##                        0                        0                         0
##  SedentaryActiveDistance        VeryActiveMinutes      FairlyActiveMinutes
##                        0                        0                         0
##     LightlyActiveMinutes         SedentaryMinutes                  Calories
##                        0                        0                         0
```

```
sum(duplicated(act_merged)) # gave me 0
```

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

```
summary(act_merged$TotalSteps) # no one had any wild amount of steps
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    3146    6999    7281   10544   36019
```

```
act_merged %>%
  group_by(Id) %>%
  summarise(zero_days = sum(TotalSteps == 0)) # checking for 0 step days by Id
```

```
## # A tibble: 35 x 2
##            Id zero_days
##         <dbl>     <int>
##  1 1503960366         1
##  2 1624580081         0
##  3 1644430081         0
##  4 1844505072        13
##  5 1927972279        14
##  6 2022484408         0
##  7 2026352035         0
##  8 2320127002         4
```

```
##  9 2347167796         1
## 10 2873212765         1
## # i 25 more rows
```

Through the cleaning process, I noted that the data set was mostly clean. Earlier I noted that there were some people who did not participate for the full time, but that was fine as it related to trends. Additionally, there were no repeats and no empty columns which was good. Lastly, the maximum steps were not a crazy discrepency. Yes, there was an outlier of over 36,000, but it isn't outside the realm of possibility that someone hit that count. I noted that as an area to explore later.

Lastly, some users noted 0 step days. I believed those were days they didn't use the device, or the device was dead, rather than being fully sedentary. However, I kept them in because they reflect real-life usage of smart devices.

Though I already noted inconsistencies in the data in terms of unique user IDs, I wanted to check, just to be sure, now that the data was merged.

```
act_merged %>%
  group_by(Id) %>%
  summarise(days_recorded = n())
```

```
## # A tibble: 35 x 2
##            Id days_recorded
##         <dbl>         <int>
##  1 1503960366            50
##  2 1624580081            50
##  3 1644430081            40
##  4 1844505072            43
##  5 1927972279            43
##  6 2022484408            43
##  7 2026352035            43
##  8 2320127002            43
##  9 2347167796            33
## 10 2873212765            43
## # i 25 more rows
```

This further proved that there were usage inconsistencies. Just like the 0 days, I kept it in becuase it could show real-life usage patterns.

3. PROCESS

Determining What I Will Focus On

Now with all of that out of the way, it was time for me to determine the columns I will be specifically focusing on. However, upon re-looking, I noticed that Total Distance and Tracker Distance were (nearly) identical. With a quick glance and head() it all looked the same, but I did not understand why the device would repeat information without reason. So, I needed to explore that.

```
sum((act_merged$TotalDistance==act_merged$TrackerDistance)==FALSE
    )
```

```
## [1] 31
```

```
#This will count the instances of FALSE that appear
```

The result was 31. Now I knew that 31 of the instances within Tracker Distance and Total Distance did not match. This told me that the information was different and potentially relevant. So, since the first few rows

were exactly the same, I assumed that it had to either be Logged Activities or Sedentary Activities. If it was neither, then it was a calculation I would need to ask about.

```
sum((act_merged$TrackerDistance==(act_merged$TotalDistance+act_merged$LoggedActivitiesDistance))==FALSE)
```

```
## [1] 56
```

```
sum((act_merged$TrackerDistance==(act_merged$TotalDistance+act_merged$SedentaryActiveDistance))==FALSE)
```

```
## [1] 145
```

The exploration didn't prove anything that I could concretely use, so I didn't probe any further.

Instead, I decided to get a sense of when people were most active

```
act_merged$day_of_week<-weekdays(act_merged$ActivityDate)
head(act_merged) # To make sure that the column is there
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366   2016-03-25      11004          7.11            7.11
## 2 1503960366   2016-03-26      17609         11.55           11.55
## 3 1503960366   2016-03-27      12736          8.53            8.53
## 4 1503960366   2016-03-28      13231          8.93            8.93
## 5 1503960366   2016-03-29      12041          7.85            7.85
## 6 1503960366   2016-03-30      10970          7.16            7.16
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               2.57                     0.46
## 2                        0               6.92                     0.73
## 3                        0               4.66                     0.16
## 4                        0               3.19                     0.79
## 5                        0               2.16                     1.09
## 6                        0               2.36                     0.51
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                4.07                       0                33
## 2                3.91                       0                89
## 3                3.71                       0                56
## 4                4.95                       0                39
## 5                4.61                       0                28
## 6                4.29                       0                30
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  12                  205              804     1819
## 2                  17                  274              588     2154
## 3                   5                  268              605     1944
## 4                  20                  224             1080     1932
## 5                  28                  243              763     1886
## 6                  13                  223             1174     1820
##   day_of_week
## 1      Friday
## 2    Saturday
## 3      Sunday
## 4      Monday
## 5     Tuesday
## 6   Wednesday
```

This created a new column for days of the week because I was curious if people were more active on certain days than others. I had assumptions, but the data would be able to provide some clarity despite the sample size being small (and likely biased).

4. ANALYZE

Now I had a better idea of where to explore to get a better understanding of user trends and behaviors. * User habits and meeting baseline steps counts * Sedentary time * And most active days of the week

```
min(act_merged$TotalSteps) #gave me 0 (not surprising)
```

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

```
max(act_merged$TotalSteps) #gave me 36019
```

```
## [1] 36019
```

```
mean(act_merged$TotalSteps, na.rm = TRUE) #7280.898 is the average amount of steps
```

```
## [1] 7280.898
```

```
median(act_merged$TotalSteps, na.rm = TRUE) #6999 is the median, which is pretty close to the mean
```

```
## [1] 6999
```

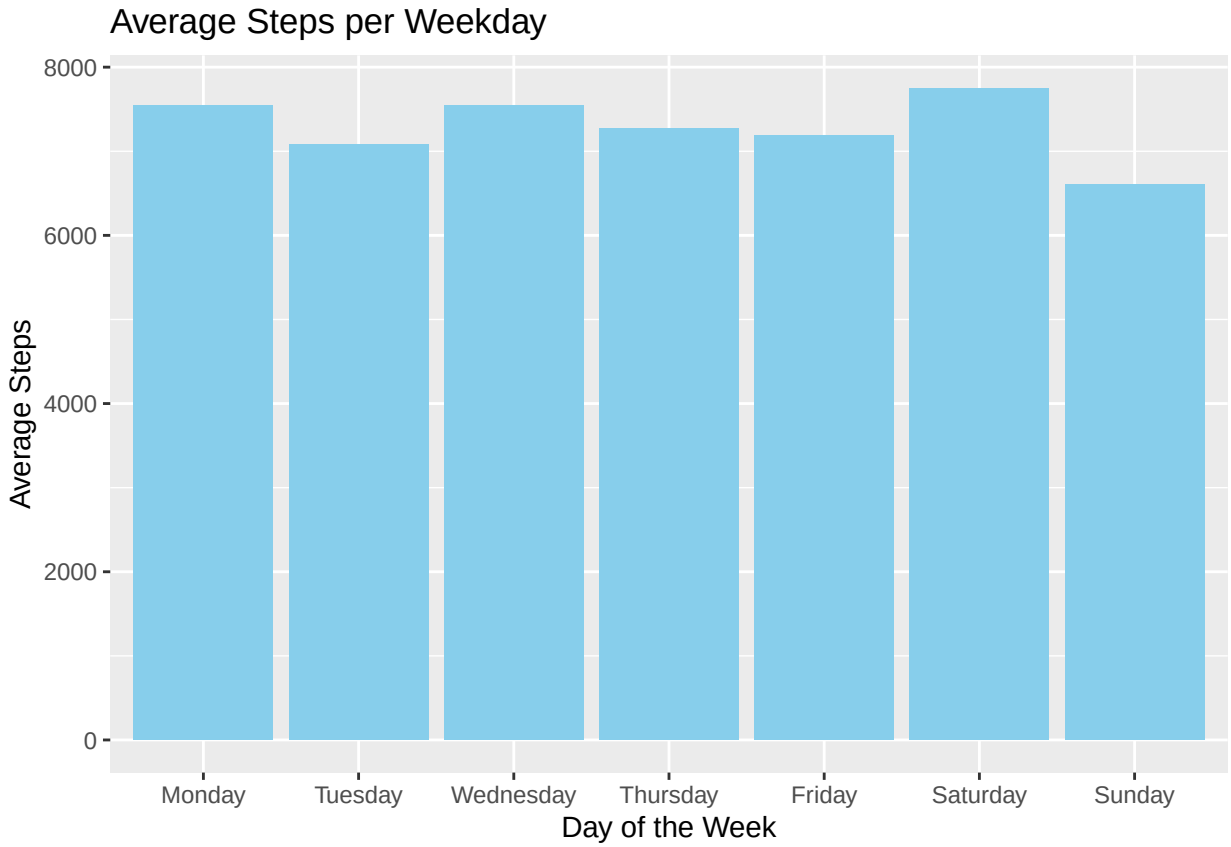
Since I had some specific pieces of information, I wanted to see a visualization to give me a better idea of what I was seeing. Additionally, I assumed that weekends were more popular for "gaining steps" and a visualization would allow me to see that.

```
library(ggplot2)
ggplot(data=act_merged)+
    geom_boxplot(mapping=aes(x=TotalSteps, y=day_of_week, fill=day_of_week)) +
    labs(title = "Total Step by Day of the Week",
         x = "Total Steps", y = "Day of the Week")
```

Based on this box plot, the outlier that exceeded 36,000 happened on a Sunday. However, since it is one outlier among many others, it would need a deeper dive. Ignoring that specific data point, I could still see that Saturday generally had more steps. Friday, Saturday, and Sunday also have more incidents of outliers.

But, we need to dive closer to be sure.

```
library(dplyr)
library(lubridate)
avg_steps_by_day <- act_merged %>%
  group_by(day_of_week) %>%
  summarise(avg_steps = mean(TotalSteps, na.rm = TRUE)) %>%
  ungroup()

avg_steps_by_day$day_of_week<-factor(avg_steps_by_day$day_of_week, levels
  = c("Monday","Tuesday","Wednesday","Thursday","Friday", "Saturday", "Sunday"))

ggplot(avg_steps_by_day, aes(x = day_of_week, y = avg_steps)) +
  geom_col(fill = "skyblue") +
  labs(title = "Average Steps per Weekday", x = "Day of the Week", y = "Average Steps")
```



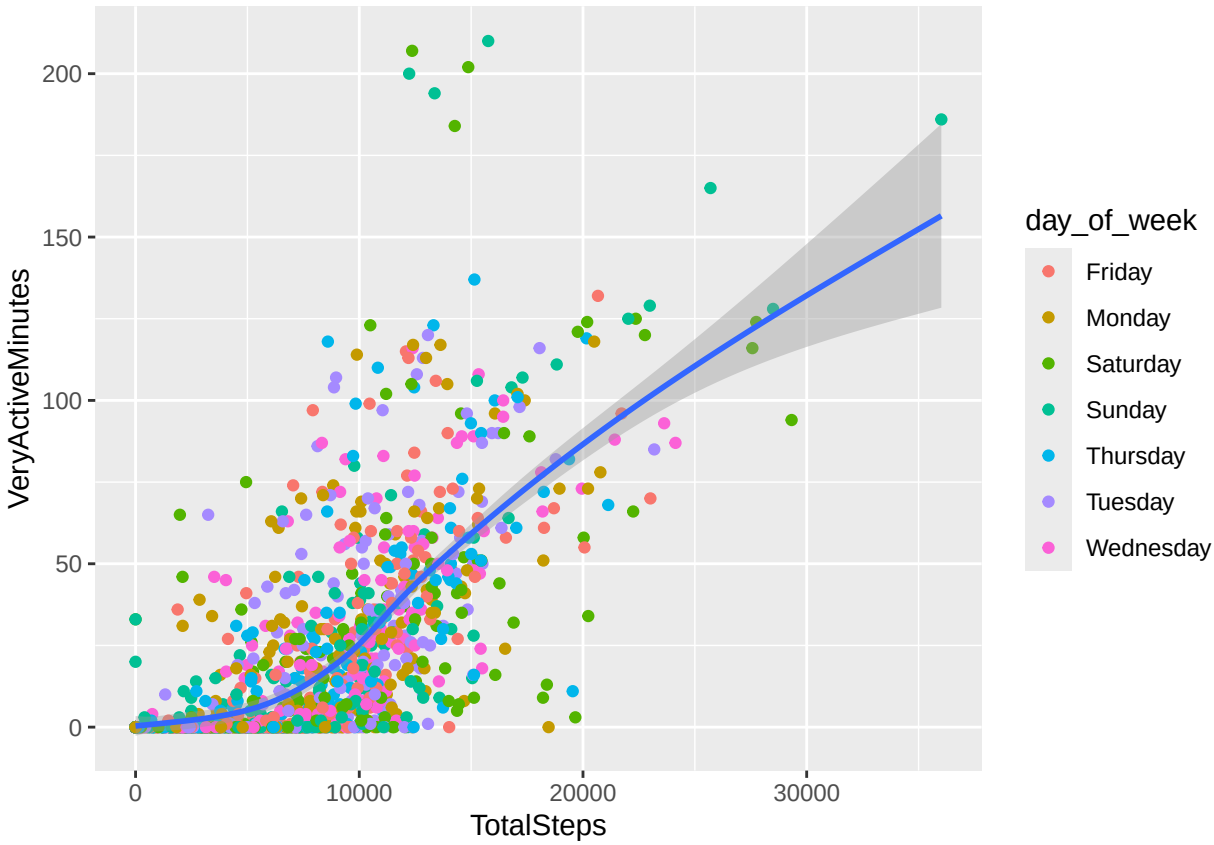
This allowed me to see the average steps per day of the week through columns. Unlike the total number, it instead shows that on average, Mondays, Wednesdays, and Saturdays have more steps with Sunday having the least on average.

It was an interesting distinction when looking at the outliers.

While I was processing that information, I decided to dive deeper into activity level and steps using the raw information.

```
ggplot(data=act_merged)+
  geom_point(mapping=aes(x=TotalSteps, y=VeryActiveMinutes, color=day_of_week))+
  geom_smooth(mapping=aes(x=TotalSteps, y=VeryActiveMinutes))
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



Taking a look at the chart, we could see that typically, as people were more active, then they were more likely to take a lot of steps. Again, this was an assumption I already had, but the visualization helped.

Next I needed to categorize the users into Above Average, Average, and Below Average to better understand the average user habits for this small sample size.

- Above Average was *over 10,000* on average.
- Average was *7,000 to 10,000* (7000 being 1 above the median) on average and this article said, "Approximately 7000 steps per day was associated with risk reductions for all outcomes examined and might serve as a practical quantitative public health target."
- Below Average was *below 7,000* steps on average.

```
user_steps <- act_merged %>% #pipe designed to create user_steps
  group_by(Id) %>% #which I then group by Id
  summarise(avg_steps = mean(TotalSteps, na.rm = TRUE)) #and then find the average
user_steps #checking to make sure it worked
```

```
## # A tibble: 35 x 2
##            Id avg_steps
##         <dbl>     <dbl>
##  1 1503960366    11936.
##  2 1624580081     5167.
##  3 1644430081     7781.
##  4 1844505072     2876.
##  5 1927972279     1269.
##  6 2022484408    11595.
##  7 2026352035     4960.
##  8 2320127002     4276.
```

```
##  9 2347167796     9647.
## 10 2873212765     7299.
## # i 25 more rows
```

```
user_steps <- user_steps %>%
  mutate(step_category = case_when(
    avg_steps < 7000 ~ "Below Average",
    # insufficient according to data and at the median or below
    avg_steps >= 7000 & avg_steps <= 10000 ~ "Average",
    # meets minimum set by 1 above the median
    avg_steps > 10000 ~ "Above Average"))
# exceeds recommended minimum
```

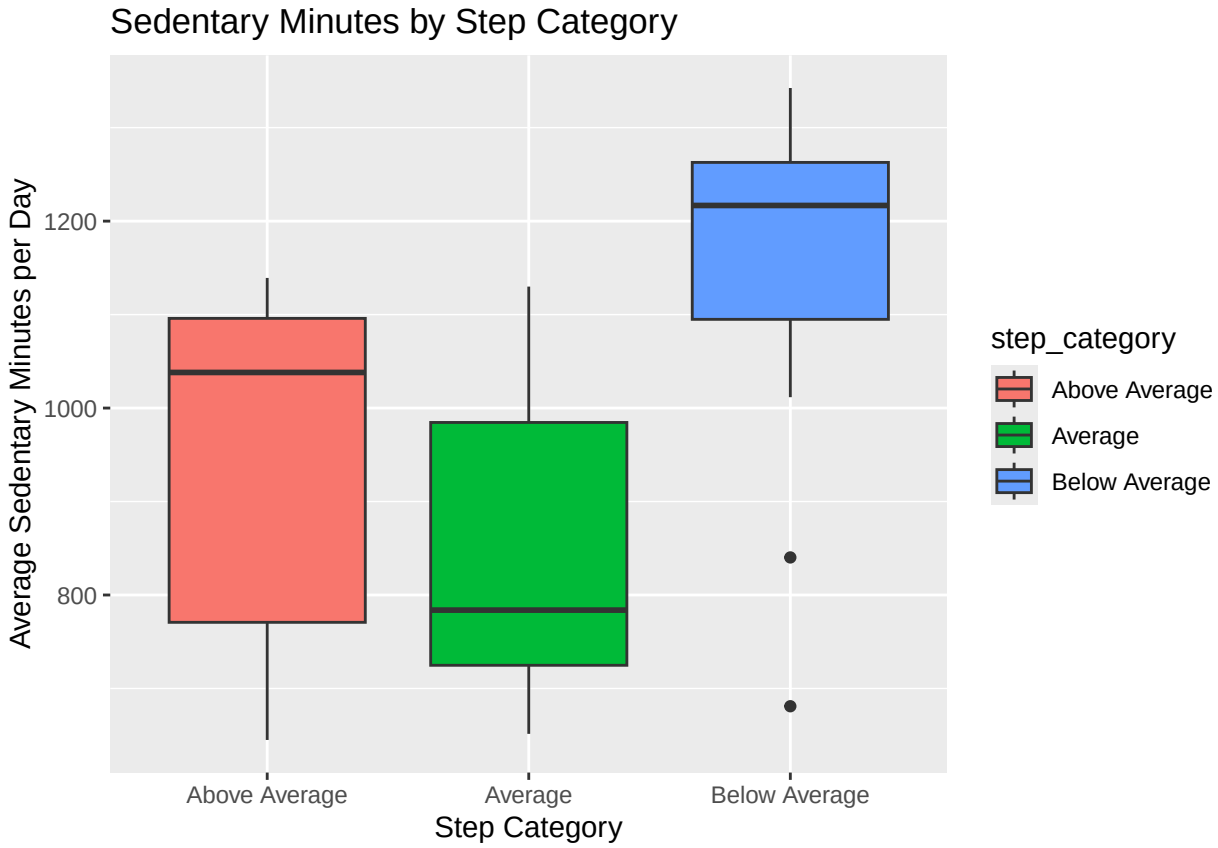
This allowed me to set things up to continue working. I wanted to see how many minutes per user were spent sedatary next.

```
user_sedentary <- act_merged %>%
  # Average sedentary minutes per user
  group_by(Id) %>%
  summarise(avg_sedentary = mean(SedentaryMinutes, na.rm = TRUE))

# Merge with step categories to better compare the two
user_analysis <- user_steps %>%
  left_join(user_sedentary, by = "Id")

# Compare sedentary time across groups to see how they compare
sed_summary <- user_analysis %>%
  group_by(step_category) %>%
  summarise(mean_sedentary = mean(avg_sedentary, na.rm = TRUE),
            sd_sedentary = sd(avg_sedentary, na.rm = TRUE),
            n = n())

print(sed_summary)
```

```
## # A tibble: 3 x 4
##   step_category mean_sedentary sd_sedentary     n
##   <chr>                  <dbl>        <dbl> <int>
## 1 Above Average           937.         204.     7
## 2 Average                 853.         175.    12
## 3 Below Average          1148.         176.    16
```

The summary showed me that some strayed from the mean in each of the categories.

```
# visualization using a boxplot
library(ggplot2)
ggplot(user_analysis, aes(x = step_category, y = avg_sedentary, fill = step_category)) +
  geom_boxplot() +
  labs(title = "Sedentary Minutes by Step Category",
       x = "Step Category", y = "Average Sedentary Minutes per Day")
```



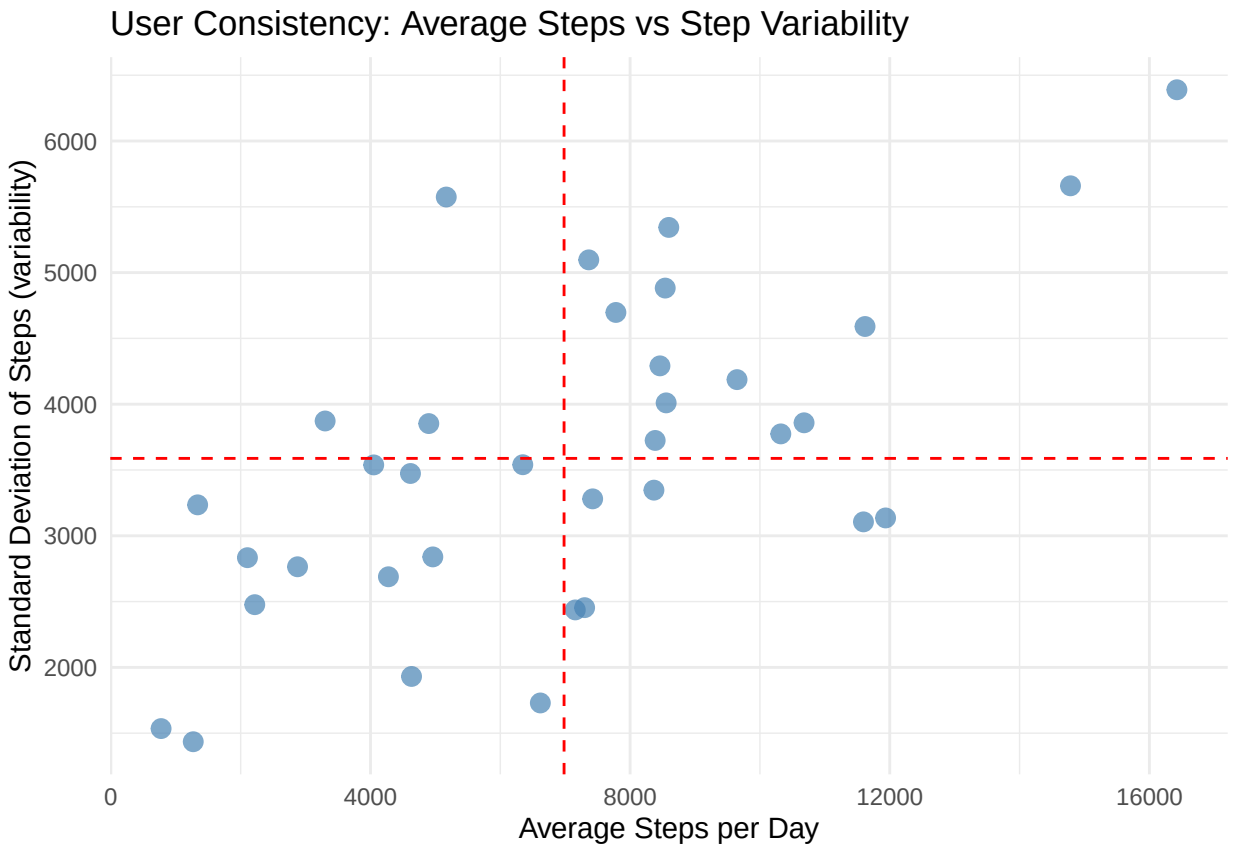
The boxplot shows the distribution of sedentary minutes by step category for the users. The line in each box indicates the median sedentary time, while the box itself represents the middle 50% of values. Longer boxes or wider spread indicate more variation in sedentary behavior among users while shorter boxes mean less variation in sedentary behavior.

Now to check the Standard Deviation.

```
consistency <- act_merged %>%
  group_by(Id) %>%
  summarise(step_sd = sd(TotalSteps, na.rm = TRUE),
            avg_steps = mean(TotalSteps, na.rm = TRUE))
tibble(consistency) #to check if it worked.
```

```
## # A tibble: 35 x 3
##            Id step_sd avg_steps
##         <dbl>   <dbl>     <dbl>
##  1 1503960366   3136.    11936.
##  2 1624580081   5574.     5167.
##  3 1644430081   4697.     7781.
##  4 1844505072   2765.     2876.
##  5 1927972279   1435.     1269.
##  6 2022484408   3106.    11595.
##  7 2026352035   2840.     4960.
##  8 2320127002   2689.     4276.
##  9 2347167796   4187.     9647.
## 10 2873212765   2454.     7299.
## # i 25 more rows
```

```r
ggplot(consistency, aes(x = avg_steps, y = step_sd)) +
  geom_point(color = "steelblue", size = 3, alpha = 0.7) +
  geom_hline(yintercept = mean(consistency$step_sd), linetype = "dashed", color = "red") +
  geom_vline(xintercept = mean(consistency$avg_steps), linetype = "dashed", color = "red") +
  labs(
    title = "User Consistency: Average Steps vs Step Variability",
    x = "Average Steps per Day",
    y = "Standard Deviation of Steps (variability)"
  ) +
  theme_minimal()
```



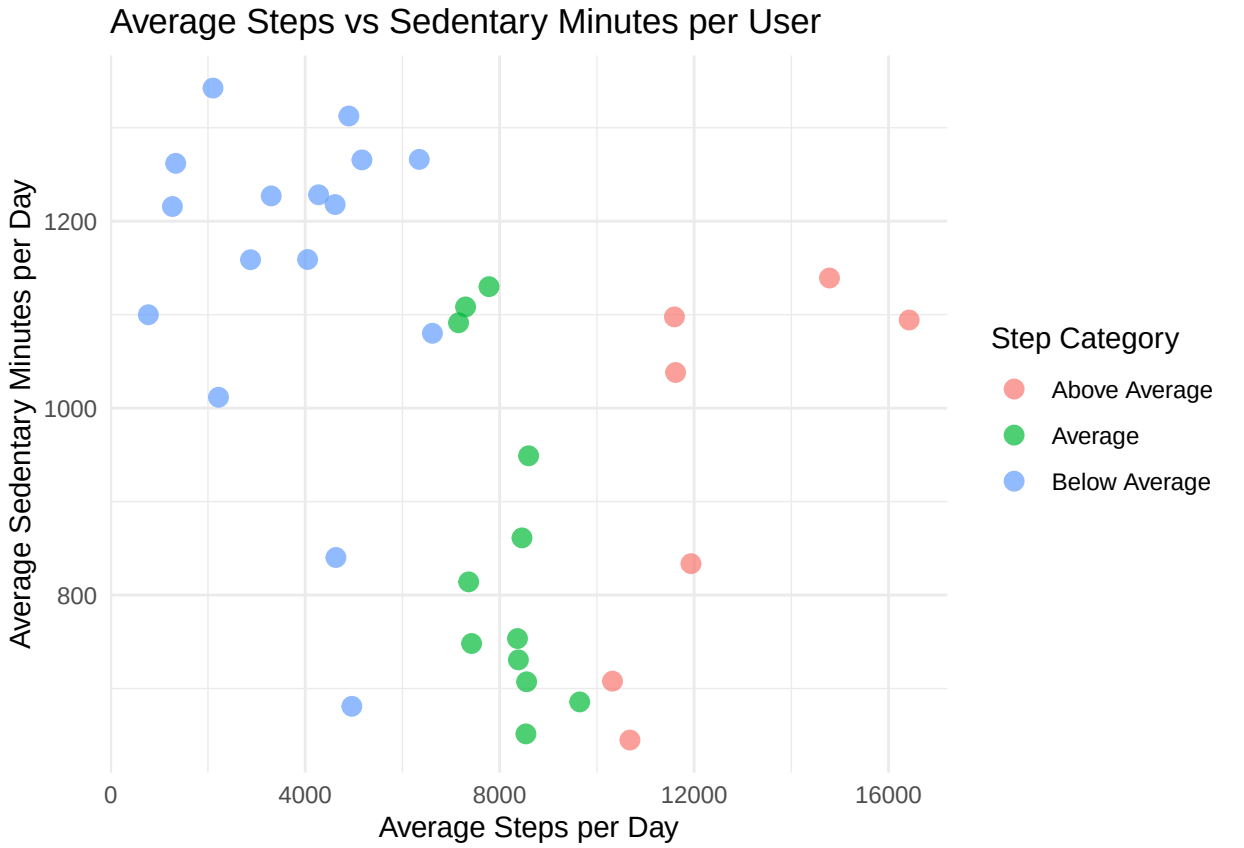
This visual gave me an understanding out how many users were consstent v who was inconsistent. The bottom showed consistency while the top showed inconsistency and the left showed low step count while the right showed high step count. Combined, the bottom-right showed high step count users and low standard deviation meaning they were consistently getting high steps, but they were not the majority.

Most of the people in the data set were either consistently low step takers or were inconsistent in being a high stepper.

Note that there were people who did not have the full set of data, but given the time they did not wear the Fitbit it would add to them being an inconsistent user of the watch, not just someone who takes fewer steps.

```r
ggplot(data=user_analysis, aes(x = avg_steps, y = avg_sedentary, color = step_category)) +
  geom_point(size = 3, alpha = 0.7) +
  labs(
    title = "Average Steps vs Sedentary Minutes per User",
    x = "Average Steps per Day",
    y = "Average Sedentary Minutes per Day",
```

```
    color = "Step Category"
  ) +
  theme_minimal()
```




```
cor(user_analysis$avg_steps, user_analysis$avg_sedentary, use = "complete.obs")
```

```
## [1] -0.4142664
```

The scatterplot showed that as people took more steps, their amount of sedetary minutes went down. When I looked into the corelation, I got -0.4142664. The negative sign means that they have a negative relationship (one going up means the other goes down) which was what I suspected from the scatterplot.

However, the correlation being closer to 0 than -1 meant that they have a weak linear relationship (or one that doesn't exist). The data was limited, but with this sample, that is the information we have.

So, we cannot concretely say that more steps meant less sedentary time, but it seemed to indicate a downward trend. With more data, it could have been possible.

5.RECOMMENDATIONS

Given everything, my suggestions are

1. **Consistency Rewards:** Focus on rewarding consistency of using the product, not just attaining their goal. If there are rewards or badges that are provided, then that would make it easier for people who are inconsistent with using their products to strive for something. This can look like a "wear-streak" similar to those seen in phone app games and other applications.
2. **Weekend Warriors or Weekend Winners:** Rewards for people who use them on the weekend. Weekends are the best times for people to strive for higher counts, shown by the larger incidence of

outliers.

3. **Time-Sensistive Collaborative and Personal Events:** Promoting competition-like events, collaborative events, and competitions. This could look like an event where people connect with friends and make a "team" for a global competition. Individual competitions and badges or time-limited rewards (similar to those of video games) would increase motivation to at least use the product. This method would promote socialization, community, and competition. These events could happen on the weekend or during popular vacation weeks.

Additional Thoughts Social Impact Potential in Marketing: Though the data did not include demographic information, upon looking at the Bellabeat site, I noticed that they frequently used the word "women" as that is their target audience. However, if they expanded it to be more inclusive with language like "people who have periods" or "people seeking wholistic wellness," it would open up the potential of a larger audience. Not all women have periods, so they may not be interested in those features. Specifically, women who have gone through menopause, women who have other health conditions that stopped their periods, and trans women do not have periods. But, there are trans men and non binary people who do still have periods. With the small change to inclusive language, it has the potential to increase the potential customer base while also appealing to inclusivity and visibility.