Bellabeat Capstone Project

Donald Brown

2023-04-30

Case Study 2: How Can a Wellness Technology Company Play It Smart?

About the company

Urška Sršen and Sando Mur founded Bellabeat, a high-tech company that manufactures health-focused smart products. Sršen used her background as an artist to develop beautifully designed technology that informs and inspires women around the world. Collecting data on activity, sleep, stress, and reproductive health has allowed Bellabeat to empower women withknowledge about their own health and habits. Since it was founded in 2013, Bellabeat has grown rapidly and quickly positioned itself as a tech-driven wellness company for women. By 2016, Bellabeat had opened offices around the world and launched multiple products. Bellabeat products became available through a growing number of online retailers in addition to their own e-commerce channel on their website. The company has invested in traditional advertising media, such as radio, out-of-home billboards, print, and television, but focuses on digital marketing extensively. Bellabeat invests year-round in Google Search, maintaining active Facebook and Instagram pages, and consistently engages consumers on Twitter. Additionally, Bellabeat runs video ads on Youtube and display ads on the Google Display Network to support campaigns around key marketing dates. Sršen knows that an analysis of Bellabeat's available consumer data would reveal more opportunities for growth. She has asked the marketing analytics team to focus on a Bellabeat product and analyze smart device usage data in order to gain insight into how people are already using their smart devices. Then, using this information, she would like high-level recommendations for how these trends can inform Bellabeat marketing strategy.

Ask

Sršen asks you to analyze smart device usage data in order to gain insight into how consumers use non-Bellabeat smart devices. She then wants you to select one Bellabeat product to apply these insights to in your presentation. These questions will guide your analysis:

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?**

You will produce a report with the following deliverables:

1. **A clear summary of the business task**
2. **A description of all data sources used**
3. **Documentation of any cleaning or manipulation of data**
4. **A summary of your analysis**
5. **Supporting visualizations and key findings**
6. **Your top high-level content recommendations based on your analysis**

I found a great article on the fitness tracker market at https://www.fortunebusinessinsights.com/fitness-tracker-market-103358 The global fitness tracker market is projected to grow from $36.34 billion in 2020 to $114.36 billion in 2028 at a CAGR of 15.4% in forecast period 2021-2028… Read More at:- https://www.fortunebusinessinsights.com/fitness-tracker-market-103358

Here are two quotes that I found relevant to the potential success of Bellabeat.

###_ "Growing awareness to stay healthy and fit to monitor fitness activities is anticipating the growth of the market. These tracking products have come a long way from being just a basic pedometer to becoming a smarter device with a colored display that tracks the sleep pattern, measures heart rate, activity monitor, and others. Being at their nascent stage in India, the fitness industry foresees a good amount of adoption, particularly amongst the younger generation. People are currently more inclined towards health clubs and gyms to limit the side-effects of a hectic lifestyle. Moreover, a regular workout helps in reducing stress, anxiety, and depression. The growing health issues are pushing people not only towards a healthy diet but also towards fitness activities. The fitness trackers help them track their exercises, thus, propelling the demand for fitness monitoring products."_

Interestingly enough, in the same article came some insights on data privacy. Particularly relevant to Bellabeat's focus on women as their primary customers.

"The data collected by fitness monitoring devices is mainly personal, entailing the user's information, which includes weight, birth date, photos, GPS coordinates, or social data, heart rate, steps, and background data used by the device. Besides the highly personal data, the primary concern is data theft. Thus, leaving the person open to privacy destructions that may cause them harm. Though, fitness tracker users are likely uninformed of the privacy implications of how the data could be misused when collected over time or when linked with other information. For Instance, in 2018, Strava on its website uploaded a heat map of users' unnamed and collective fitness tracking data."

I would suggest Bellabeat take the time to invest in better data security for it's customers, and to market it appropriately. Peace of mind for it's customers cannot be understated.

Prepare

The data I used for this project came from the FitBit Fitness Tracker Data (CC0: Public Domain, data set made available through Mobius): This Kaggle data set contains personal fitness tracker from thirty fitbit users. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and heart rate that can be used to explore users' habits.

The data was in a long format to begin with, and I decided to keep it that way. For this particular case, it would have made looking at the data sets a little harder to understand at a glance.

In order to feel confident in using this data, I first looked to see if it held any bias or credibility issues. For the Google Data Analytics course, we used the acronym ROCCC. Reliable, Original, Comprehensive, Current, and Cited.

The data is properly cited and verified. These two are critical for unbiased analysis. It is original. It isn't very current, but again, for this project it's fine. I don't expect the course to update these things regularly. We're learning how to clean data and offer analysis. For the scope of this project the data is comprehensive. You will see that we ended up with 34 unique Ids to study.

Process

I chose to use excel and R to process the data. I chose them because they are the tools I am most comfortable with. Also, I like the fact that I can do some data visualization in R and not have to move my data around. It isn't shown here in this study, but I fiddled around with the data in excel to get a feel for it. I've been using excel for years and am pretty comfortable with using it, and it's limitations. If I included all of this preliminary analysis this document would be too long and boring!

Below are the steps I took to look at the data, figure out my starting point, verify and clean as needed, and make some initial analysis as I figured out which data would be most useful to me.

Installing and loading the relevant packages:

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

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

```
library("tidyverse")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
```

```
## -- 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("readxl")
library("lubridate")
```

I had some trouble with R Studio on my computer and I had to set the working directory.

```
setwd("/cloud/project/FitBit")
```

Importing the useful data sets. After looking through all the data sets provided. I saw that ten of them were most useful for this analysis.

```
Daily_Activity <- read_csv("FitBit/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("FitBit/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_Intensity <- read_csv("FitBit/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.
```

```r
Daily_Steps <- read_csv("FitBit/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.
```

```r
Heartrate <- read_csv("FitBit/heartrate_seconds_merged.csv")
```

```
## Rows: 2483658 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Time
## dbl (2): Id, Value
##
## 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.
```

```r
Hourly_Calories <- read_csv("FitBit/hourlyCalories_merged.csv")
```

```
## Rows: 22099 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityHour
## 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.
```

```r
Hourly_Intensity <- read_csv("FitBit/hourlyIntensities_merged.csv")
```

```
## Rows: 22099 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityHour
## dbl (3): Id, TotalIntensity, AverageIntensity
##
## 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.
```

```r
Hourly_Steps <- read_csv("FitBit/hourlySteps_merged.csv")
```

```
## Rows: 22099 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityHour
## 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.
```

```r
Sleep <- read_csv("FitBit/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 <- read_csv("FitBit/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.
```

Now to check each data set for errors.

```
head(Daily_Activity)
```

```
## # A tibble: 6 x 15
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
##        <dbl> <chr>             <dbl>         <dbl>           <dbl>
## 1 1503960366 4/12/2016         13162          8.5             8.5
## 2 1503960366 4/13/2016         10735          6.97            6.97
## 3 1503960366 4/14/2016         10460          6.74            6.74
## 4 1503960366 4/15/2016          9762          6.28            6.28
## 5 1503960366 4/16/2016         12669          8.16            8.16
## 6 1503960366 4/17/2016          9705          6.48            6.48
## # i 10 more variables: LoggedActivitiesDistance <dbl>,
## #   VeryActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,
## #   LightActiveDistance <dbl>, SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>
```

```
head(Daily_Calories)
```

```
## # A tibble: 6 x 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_Intensity)
```

```
## # A tibble: 6 x 10
##         Id ActivityDay SedentaryMinutes LightlyActiveMinutes FairlyActiveMinutes
##      <dbl> <chr>                  <dbl>                <dbl>               <dbl>
## 1   1.50e9 4/12/2016                728                  328                  13
## 2   1.50e9 4/13/2016                776                  217                  19
## 3   1.50e9 4/14/2016               1218                  181                  11
## 4   1.50e9 4/15/2016                726                  209                  34
## 5   1.50e9 4/16/2016                773                  221                  10
## 6   1.50e9 4/17/2016                539                  164                  20
## # i 5 more variables: VeryActiveMinutes <dbl>, SedentaryActiveDistance <dbl>,
## #   LightActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,
## #   VeryActiveDistance <dbl>
```

```
head(Daily_Steps)
```

```
## # A tibble: 6 x 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(Heartrate)
```

```
## # A tibble: 6 x 3
##           Id Time                 Value
##        <dbl> <chr>                <dbl>
## 1 2022484408 4/12/2016 7:21:00 AM    97
## 2 2022484408 4/12/2016 7:21:05 AM   102
## 3 2022484408 4/12/2016 7:21:10 AM   105
## 4 2022484408 4/12/2016 7:21:20 AM   103
## 5 2022484408 4/12/2016 7:21:25 AM   101
## 6 2022484408 4/12/2016 7:22:05 AM    95
```

```
head(Hourly_Calories)
```

```
## # A tibble: 6 x 3
##           Id ActivityHour          Calories
##        <dbl> <chr>                    <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM       81
## 2 1503960366 4/12/2016 1:00:00 AM        61
## 3 1503960366 4/12/2016 2:00:00 AM        59
## 4 1503960366 4/12/2016 3:00:00 AM        47
## 5 1503960366 4/12/2016 4:00:00 AM        48
## 6 1503960366 4/12/2016 5:00:00 AM        48
```

```
head(Hourly_Intensity)
```

```
## # A tibble: 6 x 4
##           Id ActivityHour          TotalIntensity AverageIntensity
##        <dbl> <chr>                          <dbl>            <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM             20            0.333
## 2 1503960366 4/12/2016 1:00:00 AM               8            0.133
## 3 1503960366 4/12/2016 2:00:00 AM               7            0.117
## 4 1503960366 4/12/2016 3:00:00 AM               0            0
## 5 1503960366 4/12/2016 4:00:00 AM               0            0
## 6 1503960366 4/12/2016 5:00:00 AM               0           0
```

```
head(Hourly_Steps)
```

```
## # A tibble: 6 x 3
##           Id ActivityHour          StepTotal
##        <dbl> <chr>                     <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM       373
## 2 1503960366 4/12/2016 1:00:00 AM        160
## 3 1503960366 4/12/2016 2:00:00 AM        151
## 4 1503960366 4/12/2016 3:00:00 AM          0
## 5 1503960366 4/12/2016 4:00:00 AM          0
## 6 1503960366 4/12/2016 5:00:00 AM          0
```

```
head(Sleep)
```

```
## # A tibble: 6 x 5
##           Id SleepDay        TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##        <dbl> <chr>                       <dbl>              <dbl>          <dbl>
## 1 1503960366 4/12/2016 12:0~                 1                327            346
## 2 1503960366 4/13/2016 12:0~                 2                384            407
## 3 1503960366 4/15/2016 12:0~                 1                412            442
## 4 1503960366 4/16/2016 12:0~                 2                340            367
## 5 1503960366 4/17/2016 12:0~                 1                700            712
## 6 1503960366 4/19/2016 12:0~                 1                304            320
```

```
head(Weight)
```

```
## # A tibble: 6 x 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 4/13/2016~    134.          294.    NA  47.5 FALSE          1.46e12
## 4 2873212765 4/21/2016~     56.7         125.    NA  21.5 TRUE           1.46e12
## 5 2873212765 5/12/2016~     57.3         126.    NA  21.7 TRUE           1.46e12
## 6 4319703577 4/17/2016~     72.4         160.    25  27.5 TRUE           1.46e12
```

They all look good.

I then looked at them and compared. I found that all the information from Daily_Calories, Daily_Intensity, and Daily_Steps, were all included in the Daily_Activitiy data set. To keep things neat, I removed them.

```
rm(Daily_Calories, Daily_Intensity, Daily_Steps)
```

Next, I wanted to look at how many unique ids there were in each data set.

```
n_distinct(Daily_Activity$Id)
```

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

```
n_distinct(Heartrate$Id)
```

```
## [1] 14
```

```
n_distinct(Hourly_Calories$Id)
```

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

```
n_distinct(Hourly_Steps$Id)
```

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

```
n_distinct(Sleep$Id)
```

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

```
n_distinct(Weight$Id)
```

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

```
n_distinct(Hourly_Intensity$Id)
```

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

The data set containing information about weight has only 8 unique users. Not enough for any reliable statistical analysis. I removed this data set as well.

```
rm(Weight)
```

I wanted to make sure there were no duplicate Ids in the data sets, so I checked them. The Heartrate data set kept causing R to crash, so I had to remove it from this function. From my own work with Garmin, I do know that this data is indirectly recorded in all columns with an intensity component. The intensisty is based on heart rate. So this won't hinder any insights for this project.

```
rm(Heartrate)
sum(duplicated(Daily_Activity))
```

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

```
sum(duplicated(Hourly_Calories))
```

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

```
sum(duplicated(Hourly_Steps))
```

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

```
sum(duplicated(Sleep))
```

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

```
sum(duplicated(Hourly_Intensity))
```

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

The Sleep data set has three duplicates. For the analysis, I'll remove them. Then I'll check it to see if any duplicates are returned.

```
Sleep <- (unique(Sleep))
sum(duplicated(Sleep))
```

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

Now that all Ids are unique throughout the datasets I'll be using the next step is to clean the data. I will check the structure of the data.

```
str(Daily_Activity)
```

```
## spc_tbl_ [940 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id                      : num [1:940] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps              : num [1:940] 13162 10735 10460 9762 12669 ...
##  $ TotalDistance           : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : num [1:940] 728 776 1218 726 773 ...
##  $ Calories                : num [1:940] 1985 1797 1776 1745 1863 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDate = col_character(),
##   ..   TotalSteps = col_double(),
##   ..   TotalDistance = col_double(),
##   ..   TrackerDistance = col_double(),
##   ..   LoggedActivitiesDistance = col_double(),
##   ..   VeryActiveDistance = col_double(),
##   ..   ModeratelyActiveDistance = col_double(),
##   ..   LightActiveDistance = col_double(),
##   ..   SedentaryActiveDistance = col_double(),
##   ..   VeryActiveMinutes = col_double(),
##   ..   FairlyActiveMinutes = col_double(),
##   ..   LightlyActiveMinutes = col_double(),
##   ..   SedentaryMinutes = col_double(),
##   ..   Calories = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(Hourly_Steps)
```

```
## spc_tbl_ [22,099 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id          : num [1:22099] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour: chr [1:22099] "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00 AM"
##  $ StepTotal   : num [1:22099] 373 160 151 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityHour = col_character(),
##   ..   StepTotal = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(Hourly_Calories)
```

```
## spc_tbl_ [22,099 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id          : num [1:22099] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour: chr [1:22099] "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00 AM"
##  $ Calories    : num [1:22099] 81 61 59 47 48 48 48 47 68 141 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityHour = col_character(),
##   ..   Calories = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(Hourly_Intensity)
```

```
## spc_tbl_ [22,099 x 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id              : num [1:22099] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour    : chr [1:22099] "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00
##  $ TotalIntensity  : num [1:22099] 20 8 7 0 0 0 0 0 13 30 ...
##  $ AverageIntensity: num [1:22099] 0.333 0.133 0.117 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityHour = col_character(),
##   ..   TotalIntensity = col_double(),
##   ..   AverageIntensity = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(Sleep)
```

```
## tibble [410 x 5] (S3: tbl_df/tbl/data.frame)
##  $ Id                : num [1:410] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : chr [1:410] "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:
##  $ TotalSleepRecords : num [1:410] 1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: num [1:410] 327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : num [1:410] 346 407 442 367 712 320 377 364 384 449 ...
```

All the colums that have a date and a time in them are in character format. I'll change them to date time formatting, and check the column names. This is a common thing to run into during data analysis. When entering the date, time, or both in a single frame, the formatting doesn't always conform to whatever software you're using for analysis.

```
Daily_Activity <- Daily_Activity %>%
  rename(date= ActivityDate) %>%
  mutate(date= as_date(date, format= "%m/%d/%Y"))

Hourly_Calories <- Hourly_Calories %>%
  rename(date_time= ActivityHour) %>%
 mutate(date_time= as.POSIXct(date_time, format="%m/%d/%Y %I:%M:%S %p", tz= Sys.timezone()))

Hourly_Intensity <- Hourly_Intensity %>%
  rename(date_time= ActivityHour) %>%
 mutate(date_time= as.POSIXct(date_time, format="%m/%d/%Y %I:%M:%S %p", tz= Sys.timezone()))
```

```
Hourly_Steps <- Hourly_Steps %>%
  rename(date_time= ActivityHour) %>%
   mutate(date_time= as.POSIXct(date_time, format="%m/%d/%Y %I:%M:%S %p", tz= Sys.timezone()))

Sleep <- Sleep %>%
  rename(date= SleepDay) %>%
  mutate(date= as_date(date, format= "%m/%d/%Y  %I:%M:%S %p", tz= Sys.timezone()))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `date = as_date(date, format = "%m/%d/%Y %I:%M:%S %p", tz =
##   Sys.timezone())`.
## Caused by warning:
## ! `tz` argument is ignored by `as_date()`
```

```
head(Daily_Activity)
```

```
## # A tibble: 6 x 15
##           Id date       TotalSteps TotalDistance TrackerDistance
##        <dbl> <date>          <dbl>         <dbl>           <dbl>
## 1 1503960366 2016-04-12      13162          8.5             8.5
## 2 1503960366 2016-04-13      10735          6.97            6.97
## 3 1503960366 2016-04-14      10460          6.74            6.74
## 4 1503960366 2016-04-15       9762          6.28            6.28
## 5 1503960366 2016-04-16      12669          8.16            8.16
## 6 1503960366 2016-04-17       9705          6.48            6.48
## # i 10 more variables: LoggedActivitiesDistance <dbl>,
## #   VeryActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,
## #   LightActiveDistance <dbl>, SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>
```

```
head(Hourly_Calories)
```

```
## # A tibble: 6 x 3
##           Id date_time           Calories
##        <dbl> <dttm>                 <dbl>
## 1 1503960366 2016-04-12 00:00:00       81
## 2 1503960366 2016-04-12 01:00:00       61
## 3 1503960366 2016-04-12 02:00:00       59
## 4 1503960366 2016-04-12 03:00:00       47
## 5 1503960366 2016-04-12 04:00:00       48
## 6 1503960366 2016-04-12 05:00:00       48
```

```
head(Hourly_Intensity)
```

```
## # A tibble: 6 x 4
##           Id date_time           TotalIntensity AverageIntensity
##        <dbl> <dttm>                       <dbl>            <dbl>
## 1 1503960366 2016-04-12 00:00:00             20            0.333
## 2 1503960366 2016-04-12 01:00:00              8            0.133
## 3 1503960366 2016-04-12 02:00:00              7            0.117
## 4 1503960366 2016-04-12 03:00:00              0            0
## 5 1503960366 2016-04-12 04:00:00              0            0
## 6 1503960366 2016-04-12 05:00:00              0            0
```

```
head(Hourly_Steps)
```

```
## # A tibble: 6 x 3
##           Id date_time           StepTotal
##        <dbl> <dttm>                  <dbl>
## 1 1503960366 2016-04-12 00:00:00       373
## 2 1503960366 2016-04-12 01:00:00       160
## 3 1503960366 2016-04-12 02:00:00       151
## 4 1503960366 2016-04-12 03:00:00         0
## 5 1503960366 2016-04-12 04:00:00         0
## 6 1503960366 2016-04-12 05:00:00         0
```

```
head(Sleep)
```

```
## # A tibble: 6 x 5
##           Id date       TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##        <dbl> <date>                 <dbl>              <dbl>          <dbl>
## 1 1503960366 2016-04-12                 1                327            346
## 2 1503960366 2016-04-13                 2                384            407
## 3 1503960366 2016-04-15                 1                412            442
## 4 1503960366 2016-04-16                 2                340            367
## 5 1503960366 2016-04-17                 1                700            712
## 6 1503960366 2016-04-19                 1                304            320
```

Now I need to separate the date and the time in the data frames Hourly__Calories, Hourly__Steps, and Hourly__Intensity.

```
separate(Hourly_Calories, col=date_time, into=c('date','time'), sep=' ')
```

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

```
## # A tibble: 22,099 x 4
##            Id date       time     Calories
##         <dbl> <chr>      <chr>       <dbl>
##  1 1503960366 2016-04-12 <NA>           81
##  2 1503960366 2016-04-12 01:00:00       61
##  3 1503960366 2016-04-12 02:00:00       59
##  4 1503960366 2016-04-12 03:00:00       47
##  5 1503960366 2016-04-12 04:00:00       48
##  6 1503960366 2016-04-12 05:00:00       48
##  7 1503960366 2016-04-12 06:00:00       48
##  8 1503960366 2016-04-12 07:00:00       47
##  9 1503960366 2016-04-12 08:00:00       68
## 10 1503960366 2016-04-12 09:00:00      141
## # i 22,089 more rows
```

```
separate(Hourly_Steps, col=date_time, into=c('date','time'), sep=' ')
```

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

```
## # A tibble: 22,099 x 4
##            Id date       time     StepTotal
##          <dbl> <chr>      <chr>         <dbl>
##  1 1503960366 2016-04-12 <NA>            373
##  2 1503960366 2016-04-12 01:00:00        160
##  3 1503960366 2016-04-12 02:00:00        151
##  4 1503960366 2016-04-12 03:00:00          0
##  5 1503960366 2016-04-12 04:00:00          0
##  6 1503960366 2016-04-12 05:00:00          0
##  7 1503960366 2016-04-12 06:00:00          0
##  8 1503960366 2016-04-12 07:00:00          0
##  9 1503960366 2016-04-12 08:00:00        250
## 10 1503960366 2016-04-12 09:00:00       1864
## # i 22,089 more rows
```

```
separate(Hourly_Intensity, col=date_time, into=c('date','time'), sep=' ')
```

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

```
## # A tibble: 22,099 x 5
##            Id date       time     TotalIntensity AverageIntensity
##         <dbl> <chr>      <chr>             <dbl>            <dbl>
##  1 1503960366 2016-04-12 <NA>                 20            0.333
##  2 1503960366 2016-04-12 01:00:00              8            0.133
##  3 1503960366 2016-04-12 02:00:00              7            0.117
##  4 1503960366 2016-04-12 03:00:00              0            0
##  5 1503960366 2016-04-12 04:00:00              0            0
##  6 1503960366 2016-04-12 05:00:00              0            0
##  7 1503960366 2016-04-12 06:00:00              0            0
##  8 1503960366 2016-04-12 07:00:00              0            0
##  9 1503960366 2016-04-12 08:00:00             13            0.217
## 10 1503960366 2016-04-12 09:00:00             30            0.5
## # i 22,089 more rows
```

I will need to separate date and time in the data frames Hourly_Calories, Hourly_Steps, and Hourly_Intensity.

```
 Hourly_Calories <- separate(Hourly_Calories, date_time, into=c('date', 'time'), sep=" ")
```

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

```
Hourly_Steps <- separate(Hourly_Steps, date_time, into=c('date', 'time'), sep=" ")
```

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

```
Hourly_Intensity <- separate(Hourly_Intensity, date_time, into=c('date', 'time'), sep=" ")
```

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

I'll merge these data frames to make analysis a bit easier. The Hourly_Metric I created gave me some trouble at first. I had to merge two data sets, and then merge in the last one. For some reason if you did it the other way, the time and date columns did not merge correctly. I had to go back and look at the data sets manually and check them. It took a couple of tries to get them to merge correctly. Daily_Metrics had no trouble at all.

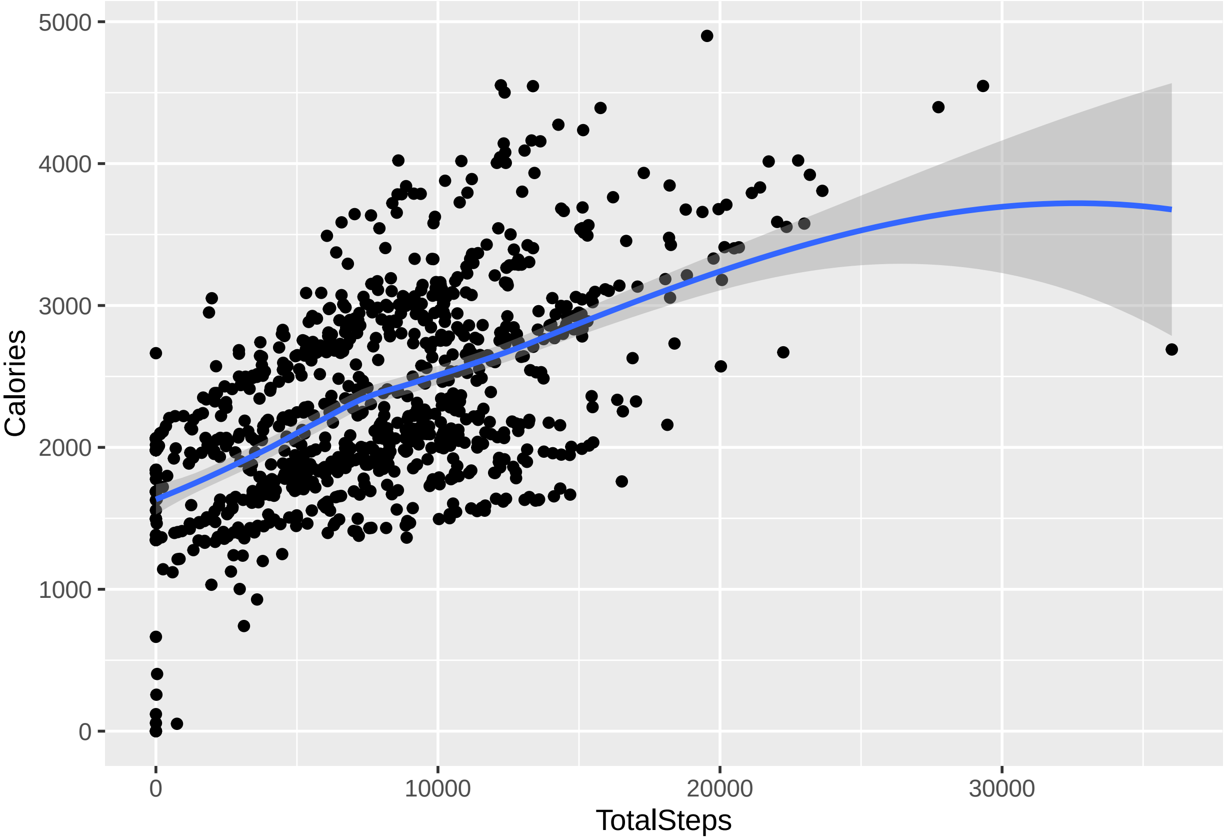
```
Daily_Metrics <- merge(Daily_Activity, Sleep, by= c("Id", "date"), all.x = TRUE)
Hourly_Metrics <- merge(Hourly_Calories, Hourly_Steps, by= c("Id", "date", "time"), all.x = TRUE)
Hourly_Metrics <- merge(Hourly_Metrics, Hourly_Intensity, by= c("Id", "date", "time"), all.x = TRUE)
```

Analyze

Now let's look for some relationships within the data.

```
ggplot(data=Daily_Metrics, mapping= aes(x=TotalSteps, y=Calories))+ geom_point()+ geom_smooth()
```

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

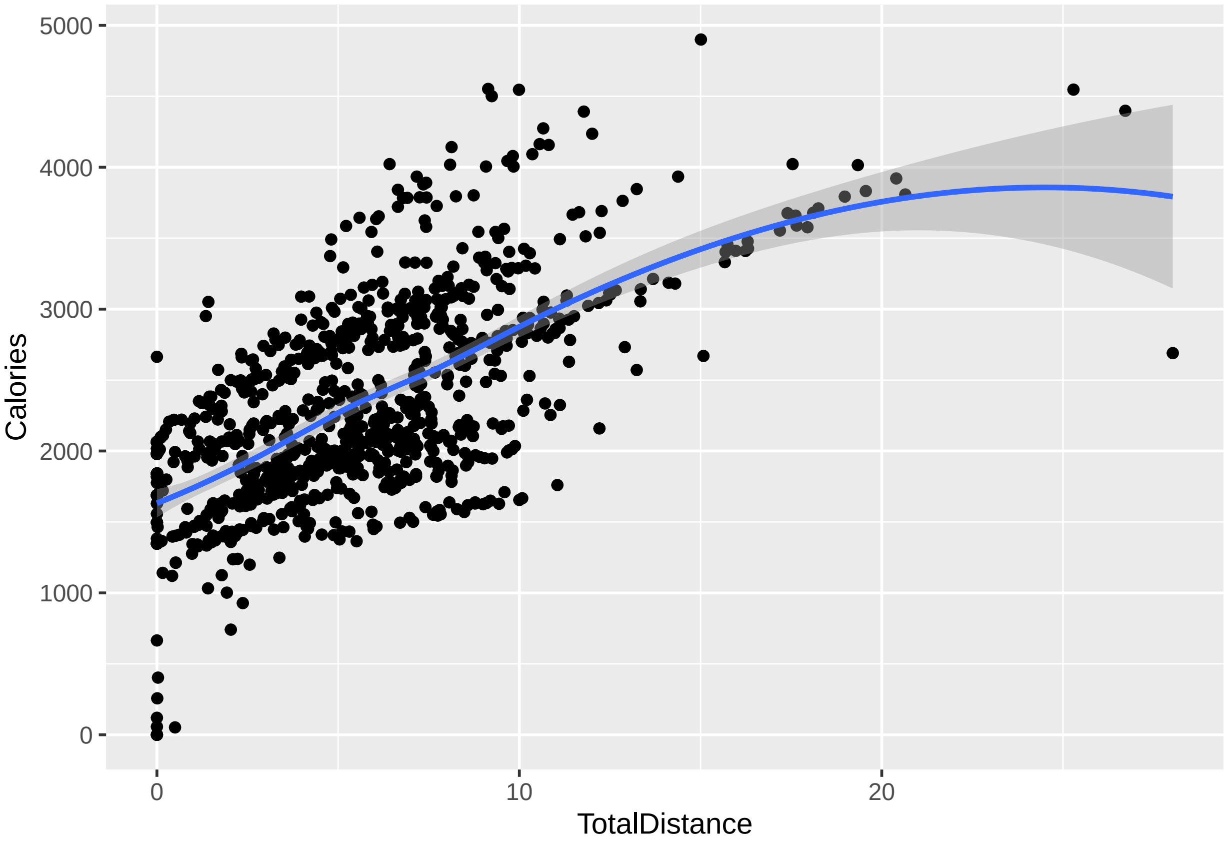


As total steps go up, the amount of calories burned also goes up. No surprise there. It does help confirm the viability of the data. If I didn't see this sort of relationship I would be worried about the sourse of the data as well as it's reliability and comprehensivenss.

Let's look at Distance over Calories too.

```
ggplot(data=Daily_Metrics, mapping= aes(x=TotalDistance, y=Calories))+ geom_point()+ geom_smooth()
```

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



### This is basically the same thing as the first graph, just substituting total distance for total steps. But again, this helps confirm the data. I do like the three points to the far right of the graph. From my own backround I can tell you a little about what I think aboout these three users. The very farthest right point is someone who gets in a lot of steps (obviously). They are in great shape, probably a runner. Why? Because as their total distance and steps goes up, their calories burned stays lower than the rest. As you get in better shape your body becomes more efficient and tends to burn less calories during litght to moderate excercise. This user gets in a lot of easy to light intensity minutes. The other two, for lack of a better explanation, go out and hammer it! Lots of steps/ distance at a high intensity. The difference is the overall calories burned. It's higher than the fist user I talked about.

Without getting into the weeds too much, lets look at these three metrics and see what we see.

```
Daily_Metrics %>%
  select(TotalSteps, Calories, FairlyActiveMinutes, LightlyActiveMinutes, TotalMinutesAsleep) %>%
  drop_na() %>%
  summary()
```

```
##    TotalSteps       Calories    FairlyActiveMinutes LightlyActiveMinutes
##  Min.   :   17   Min.   : 257   Min.   :  0.00      Min.   :  2.0
##  1st Qu.: 5189   1st Qu.:1841   1st Qu.:  0.00      1st Qu.:158.0
##  Median : 8913   Median :2207   Median : 11.00      Median :208.0
##  Mean   : 8515   Mean   :2389   Mean   : 17.92      Mean   :216.5
##  3rd Qu.:11370   3rd Qu.:2920   3rd Qu.: 26.75      3rd Qu.:263.0
##  Max.   :22770   Max.   :4900   Max.   :143.00      Max.   :518.0
##  TotalMinutesAsleep
##  Min.   : 58.0
##  1st Qu.:361.0
##  Median :432.5
##  Mean   :419.2
##  3rd Qu.:490.0
##  Max.   :796.0
```

The mean amount of sleep is around 7 hours. Calories is just shy of 2400. Steps are 8515.

According to the CDC, 7 hours is right at the low end of their recommendations. According the American Heart Association, getting 150 minutes of moderate intensity aerobic activity is recommended, so the 216.5 is a good sign for these users.

Calories are tricky to analyze in this project because we can only see the approximate number of calories burned and have no data on calories taken in. With this in mind a suggestion for the team would be enabling the app so that a user could input their daily caloric intake, or at least the foods they ate. Remembering too that not all calories are created equal. What you eat is more important than how much you eat. I remind the people I coach is: Ounces are lost during the workout and pounds are lost in the kitchen.

The data trends we see in the Bellabeat data are in line with data sets from companies like Garmin and Fitbit, both companies cited in the article I refrenced at the start of this project.

I downloaded the Bellabeat App to get a better sence of what Bellabeat has to offer. They do offer many features that help promote a healthy life style. From menue planning, mediation and mindfulness strategies, exercise suggestions, and menstrual tacking and insights.

What I am most concerned about are the poor reviews the technology gets. From innacurate step counting, to glitches in the app and it's software. My biggest recommendation to the company would be to correct these things first. People won't mind paying for this brand if the products are reliable. Any other quest for gaining market share should be secondary to making the product more relaible.

About me

My name is Donald Brown. I am a gardener and a track and field coach. I have three kids, all college aged, and am looking to change careers. I have always been a bit of a data nerd, so when I saw the opportunity to take a cartificate program in data analytics, I took it.

If you've made it to the end of this long winded case study, then thank you! It's not the most conventional analysis I saw when I went looking for examples, but I stand by it. Perhaps my age and perspective as a runner gave me a different vantage point to view this product.

Again, thank you all for your time.

Best,

Don.