Bellabeat Case Study: Insight On How Consumers Are Using Their Fitness Trackers

Jesenia Rodriguez

12/30/2023

Introduction

This is my capstone project for the Google Data Analytics Certification. I was asked to analyze smart device data in order to gain insight on how consumers use their fitness trackers. I will be analyzing data collected from Kaggle link to demonstrate the skills I acquired during this course. Using the data analysis process to answer questions for Bellabeat, a Wellness Technology Company that produces technology driven fitness products for women.

About Bellabeat

Bellabeat is the company that developed one of the first wearables specifically designed for women and has since gone on to create a portfolio of digital products for tracking and improving the health of women. Bellabeat has collected data on activity, sleep, stress, and reproductive health, products range from an app, a fashionable watch, a water bottle, and a fitness tracker.

Focusing on creating innovative health and wellness products for women, our mission is to empower women to take control of their health by providing them with technology-driven solutions that blend design and function.link

Ask

Business Task:

- Analyze Fitbit user data to understand what ways consumers are using their devices
- Provide recommendations for Bellabeat's marketing strategy

Key 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

Prepare

About the data

This Kaggle dataset is titled "Fitbit Fitness Tracker Data" contains personal fitness tracker from thirty Fitbit users. This dataset generated by respondents to a distributed survey via Amazon Mechanical Turk between 3/12/2016 and 5/12/2016. Thirty Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring, daily calories, weight logs and more, that can be used to explore users' habits. I will clean, analyze and visualize the downloaded data to Rstudio.

Credibility

I will address credibility using the R.O.C.C.C. method:

Reliablilty: This dataset is small with a sample size of 30, I would not consider this to be valid sample size. This will limit the amount of analysis that can be determined from the data.

Original: This dataset was collected from an outside source, so I it is not considered original.

Comprehensive: The data should be more comprehensive. Many factors have been committed such as the gender, age, height, and location.

Current: This dataset is not current. The dataset is 7 years old, technology, lifestyles, have changed and may not reflect what a fitness data tracker may not showcase what they look like now.

Cited: The data is cited but does not ensure that the source is credible.

Sort and Filter Data

I begin with installing packages and library's to clean and plot the data. Package: tidyverse Title: Easily Install and Load the 'Tidyverse' Version: 2.0.0 URL:https://tidyverse.tidyverse.org

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

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
```

```
## -- 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
```

Packages to clean the data

Package: here Title: A Simpler Way to Find Your Files Version: 1.0.1 URL: https://here.r-lib.org/

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

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```r
library(here)
```

```
## here() starts at /cloud/project
```

Package: janitor Title: Simple Tools for Examining and Cleaning Dirty Data Version: 2.2.0 URL: https://github.com/sfirke/janitor

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

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```r
library(janitor)
```

```
##
## Attaching package: 'janitor'

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

Package: skimr Title: Compact and Flexible Summaries of Data Version: 2.1.5 URL: https://docs.ropensci.org/skimr/ (website)

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

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```r
library(skimr)
```

Download the CSV files

Here we'll load and name the data. I decided to work with the dailyActivity_merged.csv, weightLog_merged.csv, and sleepDay_merged.csv files.

Create a dataframe for daily activity, weight log, and sleep day.

```r
daily_activity <- read.csv("dailyActivity_merged.csv")
weight<-read.csv("weightLogInfo_merged.csv")
sleep_day<-read.csv("sleepDay_merged.csv")
```

Process

Take a look at the daily__activity data, head allows us to look at the first few rows in each dataset.

```r
head(daily_activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366    4/12/2016      13162          8.50            8.50
## 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
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               1.88                     0.55
## 2                        0               1.57                     0.69
## 3                        0               2.44                     0.40
## 4                        0               2.14                     1.26
## 5                        0               2.71                     0.41
## 6                        0               3.19                     0.78
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                3.91                       0                30
## 4                2.83                       0                29
## 5                5.04                       0                36
## 6                2.51                       0                38
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  13                  328              728     1985
```

```
## 2                    19                  217                776      1797
## 3                    11                  181               1218      1776
## 4                    34                  209                726      1745
## 5                    10                  221                773      1863
## 6                    20                  164                539      1728
```

Identify all the columns in the daily__activity data.

```r
colnames(daily_activity)
```

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

Take a look at weight log and dentify column names

```r
head(weight)
```

```
##           Id                  Date WeightKg WeightPounds Fat   BMI
## 1 1503960366  5/2/2016 11:59:59 PM     52.6     115.9631  22 22.65
## 2 1503960366  5/3/2016 11:59:59 PM     52.6     115.9631  NA 22.65
## 3 1927972279  4/13/2016 1:08:52 AM    133.5     294.3171  NA 47.54
## 4 2873212765 4/21/2016 11:59:59 PM     56.7     125.0021  NA 21.45
## 5 2873212765 5/12/2016 11:59:59 PM     57.3     126.3249  NA 21.69
## 6 4319703577 4/17/2016 11:59:59 PM     72.4     159.6147  25 27.45
##   IsManualReport        LogId
## 1           True 1.462234e+12
## 2           True 1.462320e+12
## 3          False 1.460510e+12
## 4           True 1.461283e+12
## 5           True 1.463098e+12
## 6           True 1.460938e+12
```

```r
colnames(weight)
```

```
## [1] "Id"             "Date"           "WeightKg"       "WeightPounds"
## [5] "Fat"            "BMI"            "IsManualReport" "LogId"
```

Take a look at sleep day and identify column names

```r
head(sleep_day)
```

```
##           Id              SleepDay TotalSleepRecords TotalMinutesAsleep
## 1 1503960366 4/12/2016 12:00:00 AM                 1                327
## 2 1503960366 4/13/2016 12:00:00 AM                 2                384
## 3 1503960366 4/15/2016 12:00:00 AM                 1                412
## 4 1503960366 4/16/2016 12:00:00 AM                 2                340
## 5 1503960366 4/17/2016 12:00:00 AM                 1                700
## 6 1503960366 4/19/2016 12:00:00 AM                 1                304
##   TotalTimeInBed
## 1            346
## 2            407
## 3            442
```

```
## 4             367
## 5             712
## 6             320
```

```
colnames(sleep_day)
```

```
## [1] "Id"                 "SleepDay"           "TotalSleepRecords"
## [4] "TotalMinutesAsleep" "TotalTimeInBed"
```

Install packages for visualizations

Package: dplyr Title: A Grammar of Data Manipulation Version: 1.1.4 URL: https://dplyr.tidyverse.org

```
install.packages("dplyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
library(dplyr)
```

By using the n_distinct function, it produces a count of how many distinct values are in the dataset using ID columns

```
n_distinct(daily_activity$Id)
```

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

```
n_distinct(weight$Id)
```

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

```
n_distinct(sleep_day$Id)
```

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

There are 3 data sets from the number of distinct ID's.

```
nrow(daily_activity)
```

```
## [1] 940
```

```
nrow(weight)
```

```
## [1] 67
```

```
nrow(sleep_day)
```

```
## [1] 413
```

Check for duplicates

To ensure the data is not skewed, let's check for duplicates in the dataset, using the rows with duplicates function will give us a count true count of rows in the dataset.

```
nrow(daily_activity[duplicated(daily_activity),])
```

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

```
nrow(weight[duplicated(weight),])
```

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

```
nrow(sleep_day[duplicated(sleep_day),])
```

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

There are three duplicates in our sleep log. I am going to create a new sleep day data frame that only includes unique entries.

```
sleep_day_new <- unique(sleep_day)
```

Just to be sure, let's check our new data frame to look for any duplicates.

```
nrow(sleep_day_new[duplicated(sleep_day_new),])
```

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

Check to see that there are 410 observations in our sleep log

```
nrow(sleep_day_new)
```

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

Analyze

Lets look at some summary statistics.

```
daily_activity %>%
  select(TotalSteps,
         TotalDistance,
         VeryActiveMinutes,
         FairlyActiveMinutes,
         LightlyActiveMinutes,
         SedentaryMinutes,
         Calories) %>%
  summary()
```

```
##    TotalSteps    TotalDistance    VeryActiveMinutes FairlyActiveMinutes
##  Min.   :    0   Min.   : 0.000   Min.   :  0.00    Min.   :  0.00
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.:  0.00    1st Qu.:  0.00
##  Median : 7406   Median : 5.245   Median :  4.00    Median :  6.00
##  Mean   : 7638   Mean   : 5.490   Mean   : 21.16    Mean   : 13.56
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.: 32.00    3rd Qu.: 19.00
##  Max.   :36019   Max.   :28.030   Max.   :210.00    Max.   :143.00
##  LightlyActiveMinutes SedentaryMinutes    Calories
##  Min.   :  0.0        Min.   :   0.0   Min.   :   0
##  1st Qu.:127.0        1st Qu.: 729.8   1st Qu.:1828
##  Median :199.0        Median :1057.5   Median :2134
##  Mean   :192.8        Mean   : 991.2   Mean   :2304
##  3rd Qu.:264.0        3rd Qu.:1229.5   3rd Qu.:2793
##  Max.   :518.0        Max.   :1440.0   Max.   :4900
```

```
weight %>%
  select(BMI,
         WeightPounds,
         Fat) %>%
  summary()
```

```
##       BMI         WeightPounds        Fat
##  Min.   :21.45   Min.   :116.0   Min.   :22.00
##  1st Qu.:23.96   1st Qu.:135.4   1st Qu.:22.75
##  Median :24.39   Median :137.8   Median :23.50
##  Mean   :25.19   Mean   :158.8   Mean   :23.50
##  3rd Qu.:25.56   3rd Qu.:187.5   3rd Qu.:24.25
```

```
##  Max.   :47.54   Max.   :294.3   Max.   :25.00
##                                  NA's   :65
```

```
sleep_day_new %>%
  select(TotalSleepRecords,
  TotalMinutesAsleep,
  TotalTimeInBed) %>%
  summary()
```

```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.00      Min.   : 58.0      Min.   : 61.0
##  1st Qu.:1.00      1st Qu.:361.0      1st Qu.:403.8
##  Median :1.00      Median :432.5      Median :463.0
##  Mean   :1.12      Mean   :419.2      Mean   :458.5
##  3rd Qu.:1.00      3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :3.00      Max.   :796.0      Max.   :961.0
```

Analysis Observations

daily_activity:

- Average calories burned is 2,304.
- Average total daily steps is 7,638.
- Average very active minutes is 21.16 minutes.
- Average fairly active minutes is 13.56 minutes.
- Average light active minutes is 192.8 mintues or 3.21 hours.
- Average sedentary minutes is 991.2 minutes or 16.52 hours.
- On average people spend a large amount of time lightly active.
- On average people spend a significant amount of time sitting and not active at all.

sleep_day_new:

- Average minutes in bed 463.0 minutes or 7.71 hours.

- Average minutes alseep 419.2 minutes or 6.98 hours.

- Users time in bed vs asleep is fairly similar.

- On average the amount of sleep is considered healthy and reports states 7 hours helps live longer link.

weight:

- Average weight 158.8lbs.

- Average BMI 25.19.

- Average fat 23.5.

- The amount of entries is significantly lower than the other data sets.

- Users are not entering their weight data into the app.

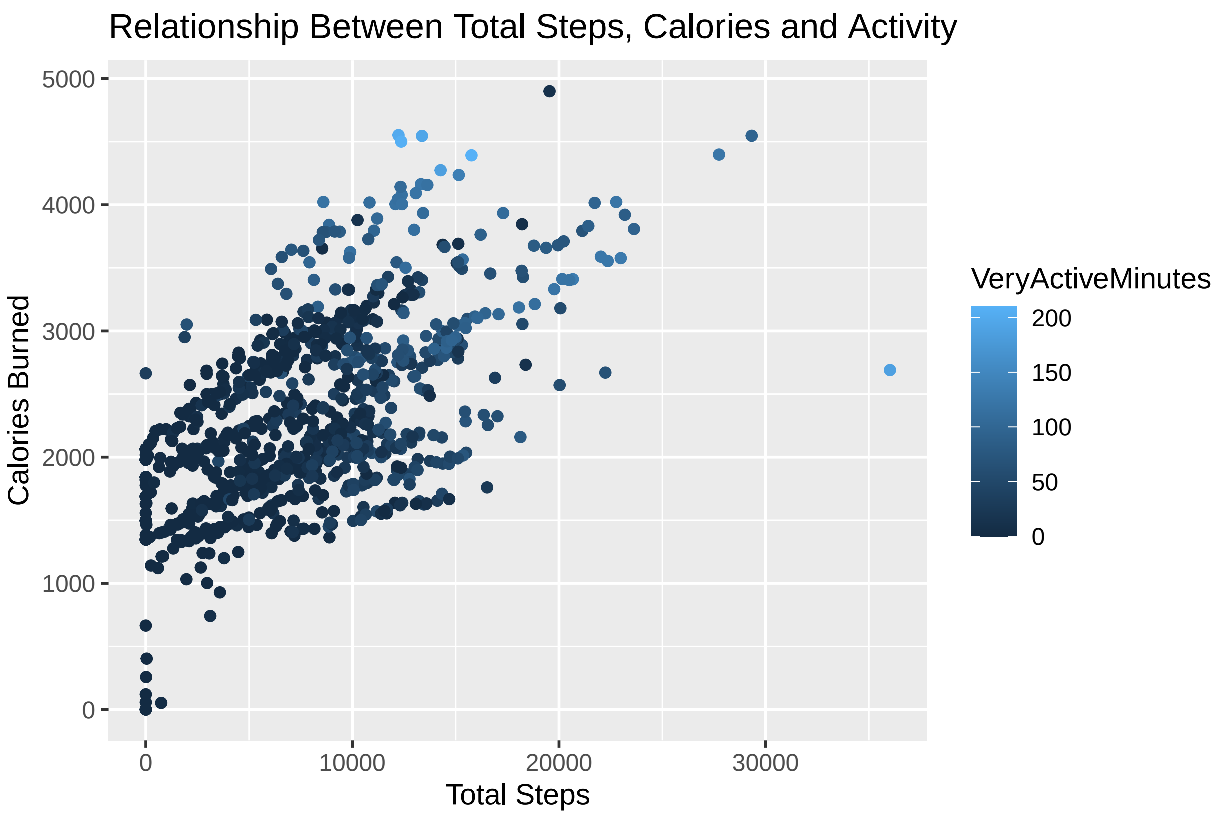
- There is data missing to calculate correct BMI.

Share: Visualizations

Let's begin by creating visualizations using plot charts.

First, the relationship between Total Steps, Calories and Activity

```
ggplot(data=daily_activity)+
  geom_point(mapping=aes(x=TotalSteps, y=Calories, color=VeryActiveMinutes)) +
  labs(title="Relationship Between Total Steps, Calories and Activity", caption="Correlation between ste
```



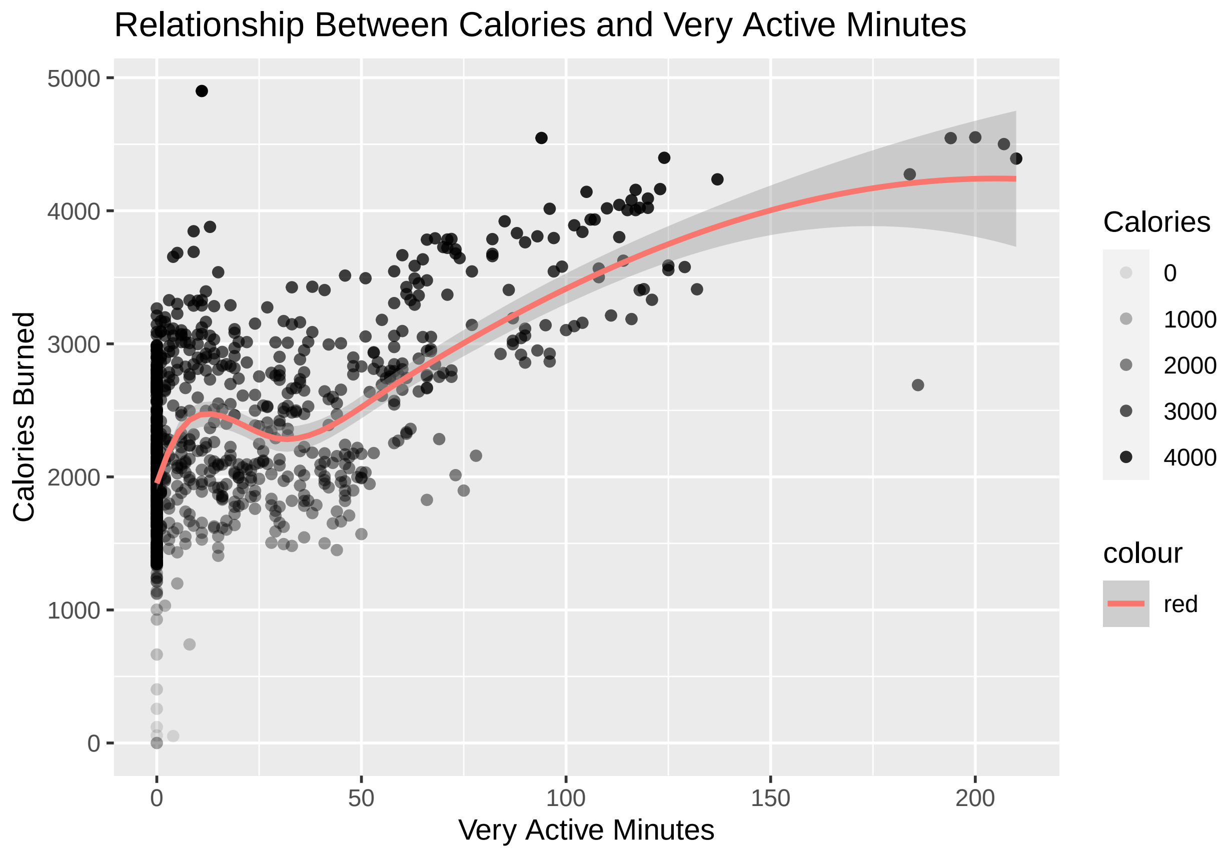
Correlation between steps and calories

Majority of calories burned consistently burned is for a longer period of time. The Very Active Minutes is used as the scale, that the more active the user is, the more calories they burn.

Next, let's look deeper at the relationship between calories and very active minutes, calories and fairly active minutes and calories and lightly active minutes.

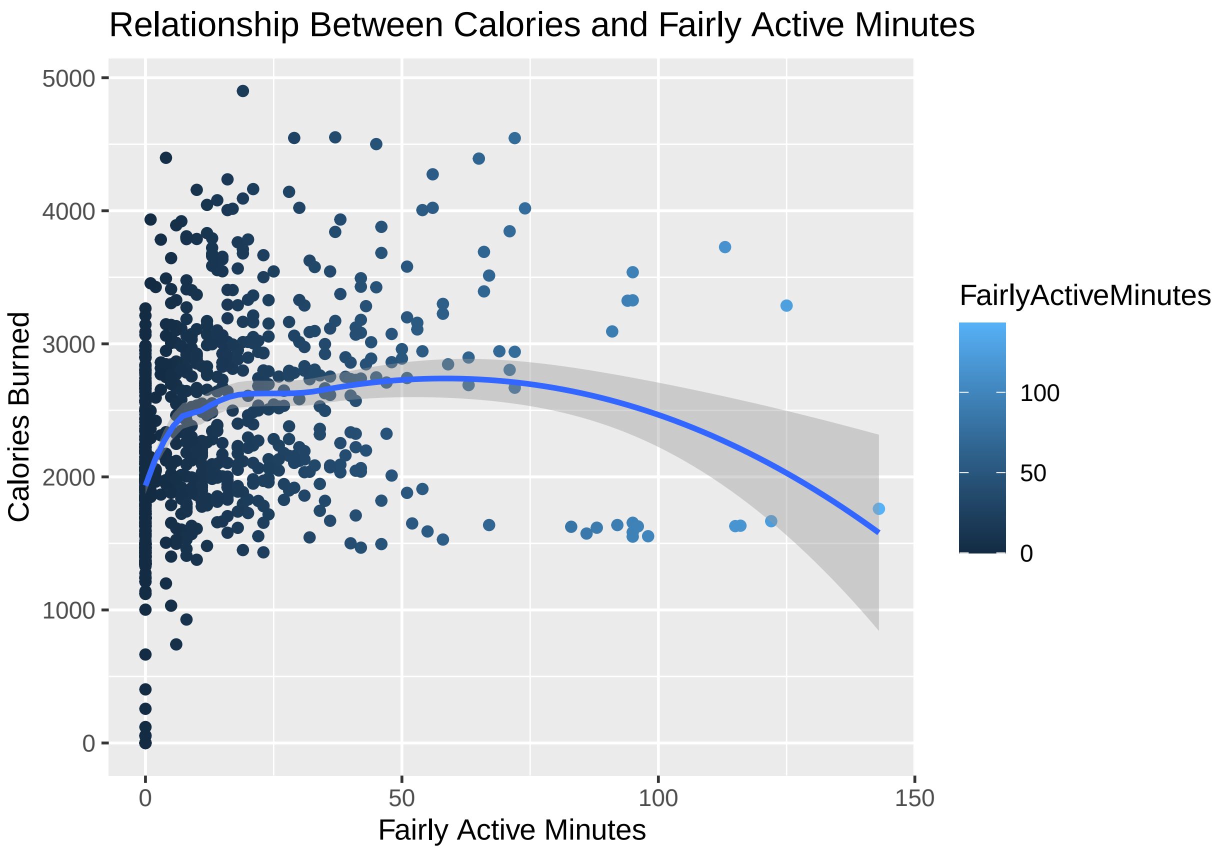
```
ggplot(data=daily_activity)+
  geom_point(mapping=aes(x=VeryActiveMinutes, y=Calories, alpha=Calories))+
  geom_smooth(mapping=aes(x=VeryActiveMinutes, y=Calories, color="red"))+
  labs(title="Relationship Between Calories and Very Active Minutes", x="Very Active Minutes",y="Calori
```

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



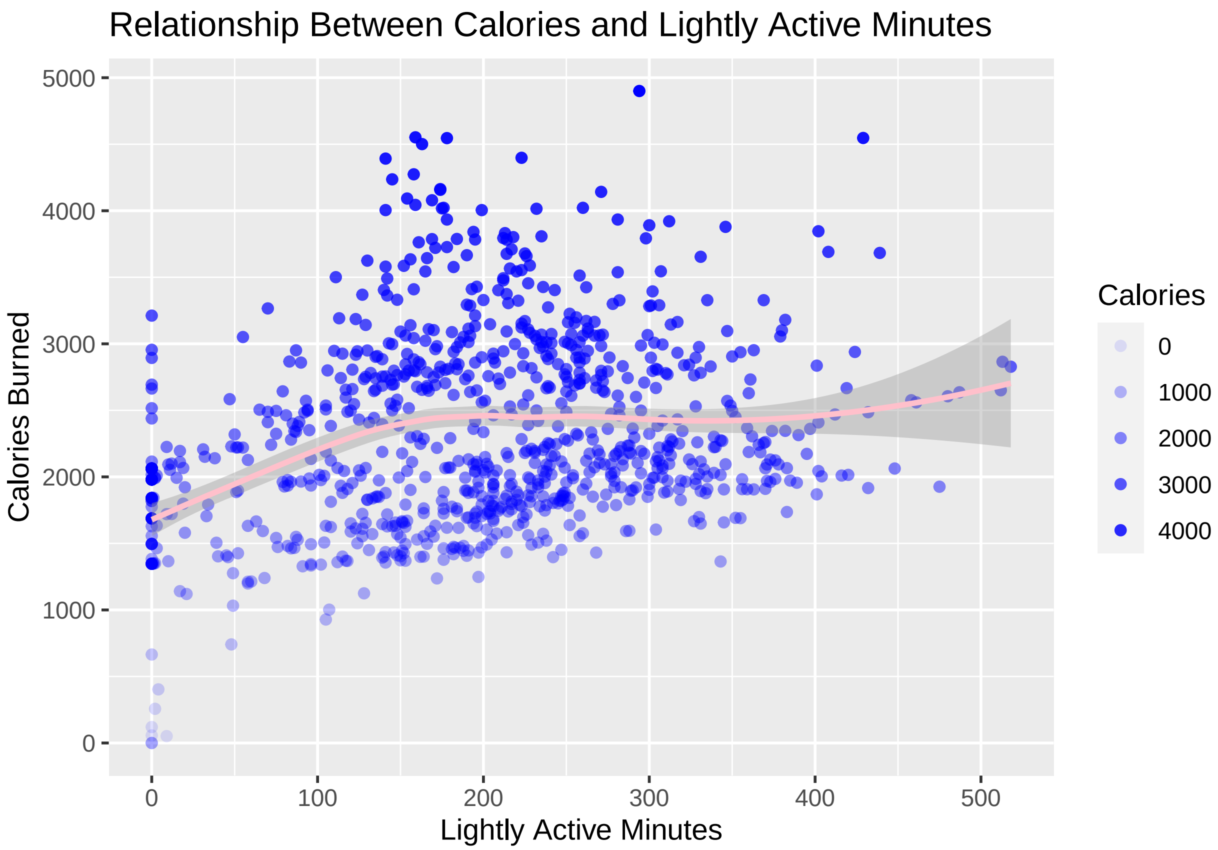

```
ggplot(data=daily_activity)+
  geom_point(mapping=aes(x=FairlyActiveMinutes, y=Calories, color=FairlyActiveMinutes))+
  geom_smooth(mapping=aes(x=FairlyActiveMinutes, y=Calories))+
  labs(title="Relationship Between Calories and Fairly Active Minutes", x="Fairly Active Minutes",y="Cal
```

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




```r
ggplot(data=daily_activity) +
  geom_point(mapping=aes(x=LightlyActiveMinutes, y=Calories, alpha=Calories), color='blue') +
  geom_smooth(mapping=aes(x=LightlyActiveMinutes, y=Calories), color='pink')+
  labs(title="Relationship Between Calories and Lightly Active Minutes", x="Lightly Active Minutes",y="C
```

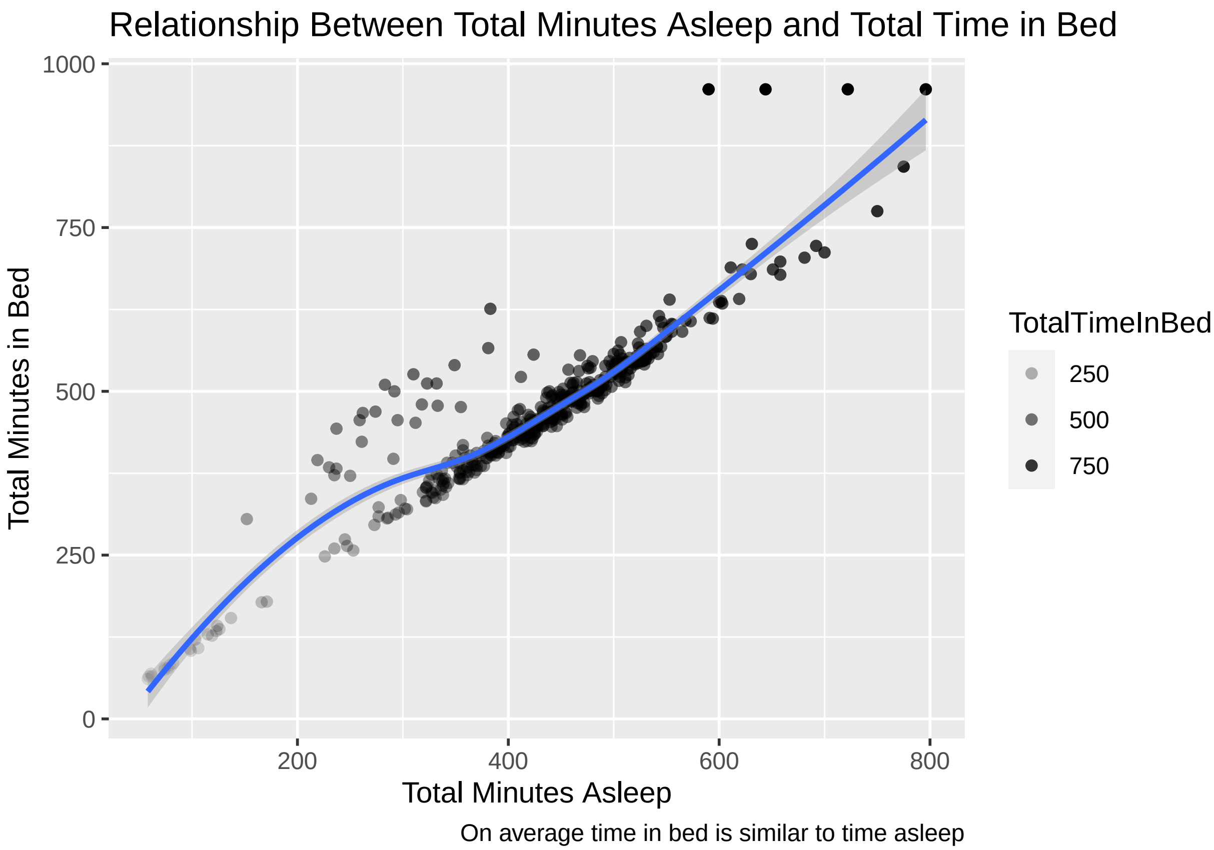
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Last, the relationship between minutes slept and time in bed.

```
ggplot(data=sleep_day_new)+
  geom_point(mapping=aes(x=TotalMinutesAsleep, y=TotalTimeInBed, alpha=TotalTimeInBed))+
  geom_smooth(mapping=aes(x=TotalMinutesAsleep, y=TotalTimeInBed)) +
  labs(title="Relationship Between Total Minutes Asleep and Total Time in Bed", caption="On average tim
```

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



Act: Key Findings

The amount of users entering their information manually is low.

- Some users are not sleeping with their Fitbit.
- The number of distinct user ID's is not consistent in all of the data sets.
- lightly active users are buring calories consistently for longer periods of time.
- Very active users burn calories similar to lightly active users.
- Bellabeat should not focus on gathering data manually, if they do it should be implemented as a requirement, in order to get more accurate data.

Users are spending large amount of time being inactive.

- Bellabeat should promote that users don't need very active workouts to burn calories.
- Bellabeat should promote a positive message about body image to help users feel more comfortable entering their weight. This will lead to more accurate data.
- Promoting the comfort and battery life, along with style of the product, to ensure more sleep data is recorded.
- Bellabeat should consider collecting data from a larger sample.