BellabeatCaseStudy

Lori Bettencourt

2022-09-15

The Ask

From Coursera's Google Data Analytics Case Study Assignment: You are a junior data analyst working on the marketing analyst team at Bellabeat, a high-tech manufacturer of health-focused products for women. Urška Sršen believes that analyzing smart device fitness data could help unlock new growth opportunities for the company. You have been asked to focus on one of Bellabeat's products and analyze smart device data to gain insight into how consumers are using their smart devices. Collecting data on activity, sleep, stress, and reproductive health has allowed Bellabeat to empower women with knowledge about their own health and habits. 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.

Business Task

How can fitness smart device trends help influence Bellabeat's marketing strategy for their customers? **Stakeholders to consider:** Urška Sršen, a Co-founder and Chief Creative Officer. Should keep Sando Mur, a Co-founder, Mathematician, and Executive in mind as well.

Include necessary packages

Check for and load required packages - installing first, if necessary.

```
## [1] "Load package: tidyverse"
## [1] "Load package: lubridate"
## [1] "Load package: ggplot2"
## [1] "Load package: ggpubr"
## [1] "Load package: janitor"
```

Prepare

It was recommended that FitBit Fitness Tracker Data be retrieved from Mobius to be analyzed. This Kaggle data set contains personal fitness tracker information from 33 FitBit users and is stored in long format. There are some limitations to the data in that the collection is third-party data; it is for less than of month of activity from 2016, which is quite outdated; and it is far from comprehensive with a limited user base of 33 individuals. Its reliability is therefore, questionable. Even though Mobius cites the origin of the data, this does not do much to improve its credibility. Still, I will use it as instructed to gather insights in a quest to address the business task.

Data preparation discoveries

- Column names in each data frame were examined to see which data are available for analysis and if there are common key fields in each file. The Id and date are common in all tables and are used for joining.

- 33 unique Ids were found in all of the daily files, with the sleepDay_merged.csv file being the exception having 24.
- The weightLogInfo_merged.csv file contained information for only 8 Ids. This file was, therefore, dropped from the study as no significant finding are expected to be found from such a limited sample.
- We will exclude the files dailyCalories_merged.csv and dailyIntensities_merged.csv because the information in those files is already contained in dailyActivity_merged.csv.

Read csv data into data frames

```
daily_act_df <- read.csv("/cloud/project/Datasets/dailyActivity_merged.csv")
daily_cal_df <- read.csv("/cloud/project/Datasets/dailyCalories_merged.csv")
daily_int_df <- read.csv("/cloud/project/Datasets/dailyIntensities_merged.csv")
sleep_day_df <- read.csv("/cloud/project/Datasets/sleepDay_merged.csv",
                         stringsAsFactors = FALSE)
weight_df <- read.csv("/cloud/project/Datasets/weightLogInfo_merged.csv")
```

Compare column names

```
colnames(daily_act_df)
```

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

```
colnames(daily_cal_df)
```

```
## [1] "Id"          "ActivityDay" "Calories"
```

```
colnames(daily_int_df)
```

```
##  [1] "Id"                       "ActivityDay"
##  [3] "SedentaryMinutes"         "LightlyActiveMinutes"
##  [5] "FairlyActiveMinutes"      "VeryActiveMinutes"
##  [7] "SedentaryActiveDistance"  "LightActiveDistance"
##  [9] "ModeratelyActiveDistance" "VeryActiveDistance"
```

```
colnames(sleep_day_df)
```

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

```
colnames(weight_df)
```

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

Determine unique Id count in each data frame

```
## [1] "Unique Ids in daily_act_df: 33"

## [1] "Unique Ids in daily_cal_df: 33"

## [1] "Unique Ids in daily_int_df: 33"

## [1] "Unique Ids in sleep_day_df: 24"

## [1] "Unique Ids in weight_df: 8"
```

Clean (process) the data

- Column names are cleaned so that they are unique and consistent in each data frame and so these names only include characters, numbers, and underscores. This also casts them to lower-case, as is standard for variable naming conventions.
- Row data and data types for each data frame were inspected using the str function.
- Each data frame was examined to compare its number of observations before and after removing duplicates. This comparison was made to check if a significant amount of observations were dropped, which would compromise its integrity.
- The sleep_day_df had three duplicate rows, which were removed. No other data frames had duplicate observations.
- Leading and trailing white space was removed from character data types.
- Date columns in each data frame were renamed to 'date' and cast as a date type rather than chr.
- Dropped null values from sedentary_minutes, lightly_active_minutes, fairly_active_minutes, very_active_minutes, total_steps fields of daily_act_df data frame and confirmed the row count was not impacted.
- Dropped null values from id, date, total_minutes_asleep, total_time_in_bed fields of sleep_day_df data frame and confirmed the row count was not impacted.

Clean column names

```r
daily_act_df <- clean_names(daily_act_df)
sleep_day_df <- clean_names(sleep_day_df)
```

Inspect row data and data types

```
## [1] "daily_act_df"
```

```
## 'data.frame':    940 obs. of  15 variables:
##  $ id                        : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ activity_date             : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ total_steps               : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ total_distance            : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ tracker_distance          : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ logged_activities_distance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ very_active_distance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ moderately_active_distance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ light_active_distance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ sedentary_active_distance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ very_active_minutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ fairly_active_minutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ lightly_active_minutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ sedentary_minutes         : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ calories                  : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```
## [1] "sleep_day_df"
```

```
## 'data.frame':    413 obs. of  5 variables:
##  $ id                  : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ sleep_day           : chr  "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:00 AM
##  $ total_sleep_records : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ total_minutes_asleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ total_time_in_bed   : int  346 407 442 367 712 320 377 364 384 449 ...
```

Determine number of observations in each data frame after removing duplicates

```
## [1] "Row count daily_act_df after removing existing duplicates: 940"
```

```
## [1] "Row count sleep_day_df after removing existing duplicates: 410"
```

Remove leading and trailing white space(s) from strings

```
daily_act_df <- daily_act_df %>%
  mutate_if(is.character, str_trim)
sleep_day_df <- sleep_day_df %>%
  mutate_if(is.character, str_trim)
```

Rename date columns and change data type to date

```
## [1] "The data type of the daily_act_df 'date' column is now Date."
```

```
## [1] "The data type of the sleep_day_df 'date' column is now Date."
```

Confirm there are no null values in the fields we use for joining and graphing.

```
## [1] "daily_act_df row count before dropping rows with null(s): 940"
```

```
## [1] "daily_act_df row count after dropping rows with null(s): 940"
```

```
## [1] "sleep_day_df row count before dropping rows with null(s): 410"
```

```
## [1] "sleep_day_df row count after dropping rows with null(s): 410"
```

Analyze the data with the use of visualizations

Examining summary statistics for key values in the daily_act_df, we get a sense of the range of data as well as some measures of central tendency and quartile data. We created a column in this data frame to store the sum of all active minutes called total_active_minutes.

Visualizations were created to examine the relationship between calories burned each day and sedentary_minutes, lightly_active_minutes, fairly_active_minutes, very_active_minutes, and total_active minutes for that day. We can see from each scatter plot that as the intensity and duration of activity both increase, the higher the correlation between the activity intensity and the number of calories burned, and the greater the slope of the regression line. The calories burned with increased sedentary time shows a negative correlation and negative slope. The last visualization mapping calories burned against total activity minutes shows a steady increase overall. The data show a positive correlation between calories burned and activity time.

Looking more specifically at calories burned against steps on a given day, we also see a positive correlation and the slope of our regression line is also positive for our scatter plot. When we break down step count into categories using quartile ranges, we can see that each quartile has a higher minimum, q1, median, q3, and maximum calorie burn than the previous quartile with fewer steps that day.

We also examined summary statistics in the sleep_day_df before taking a look at sleep data. We checked to see if either total_active_minutes or total_steps has a relationship with total_minutes_asleep and it is negligible. We also inspected total_minutes_asleep against total_time_in_bed, as instructed. As we would expect, the data are nearly perfectly linear and have a 0.93 correlation with a regression line slope of 0.87.

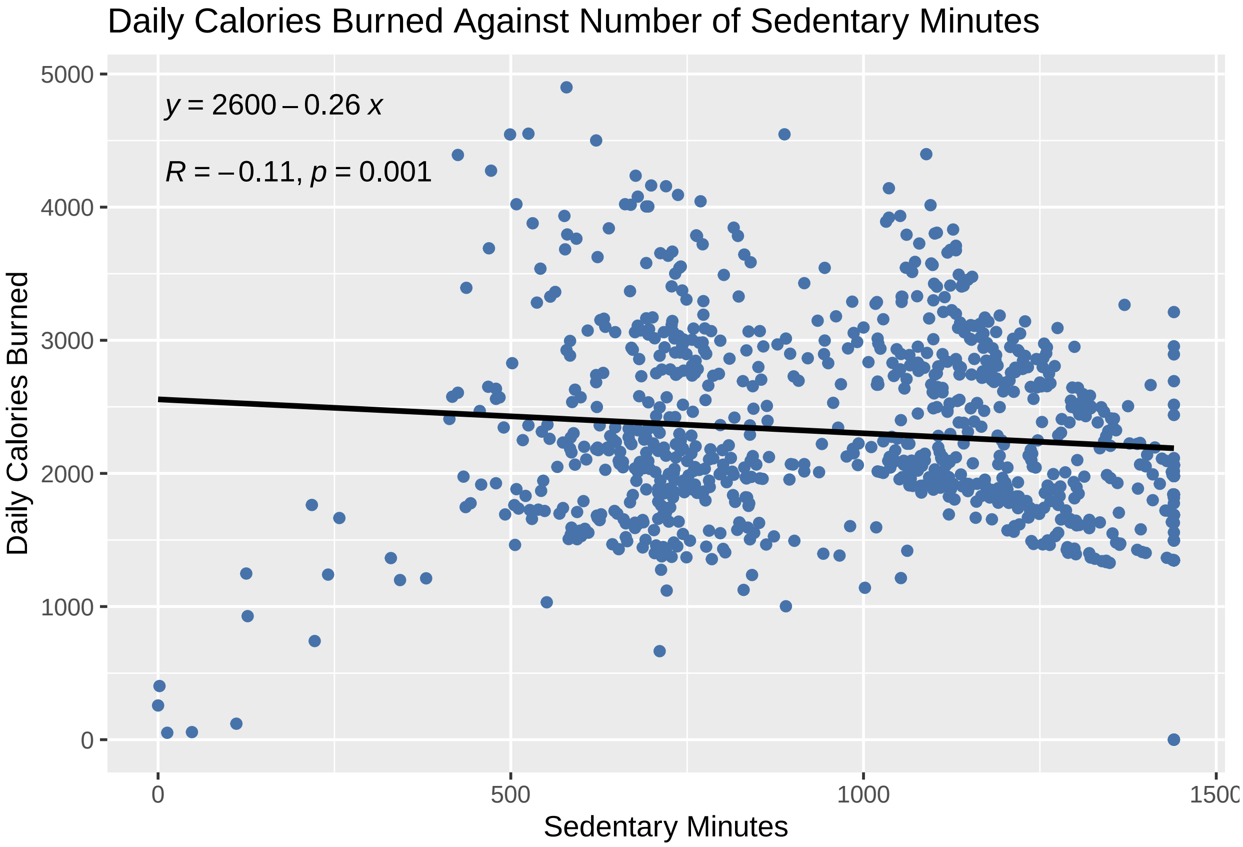
A look at activity

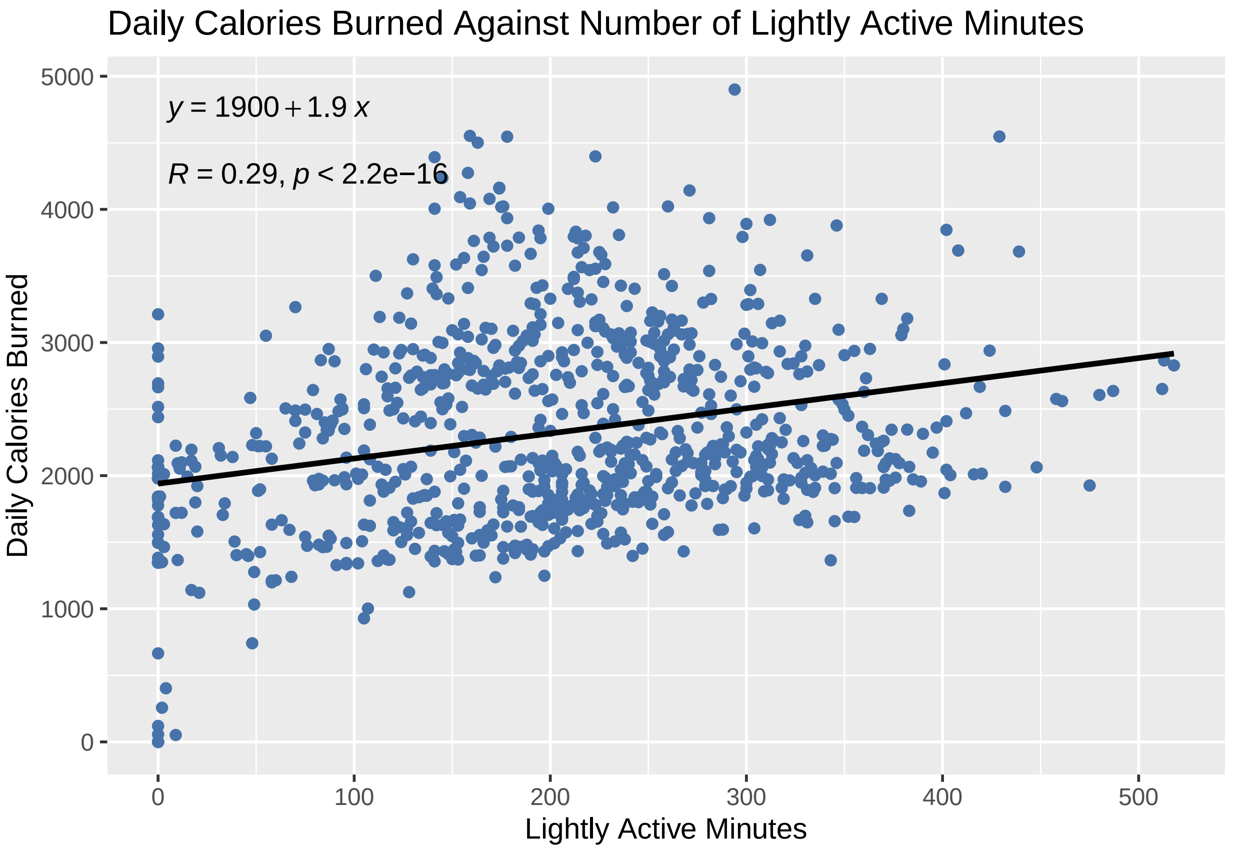
Summary stats for fields in daily_act_df data frame

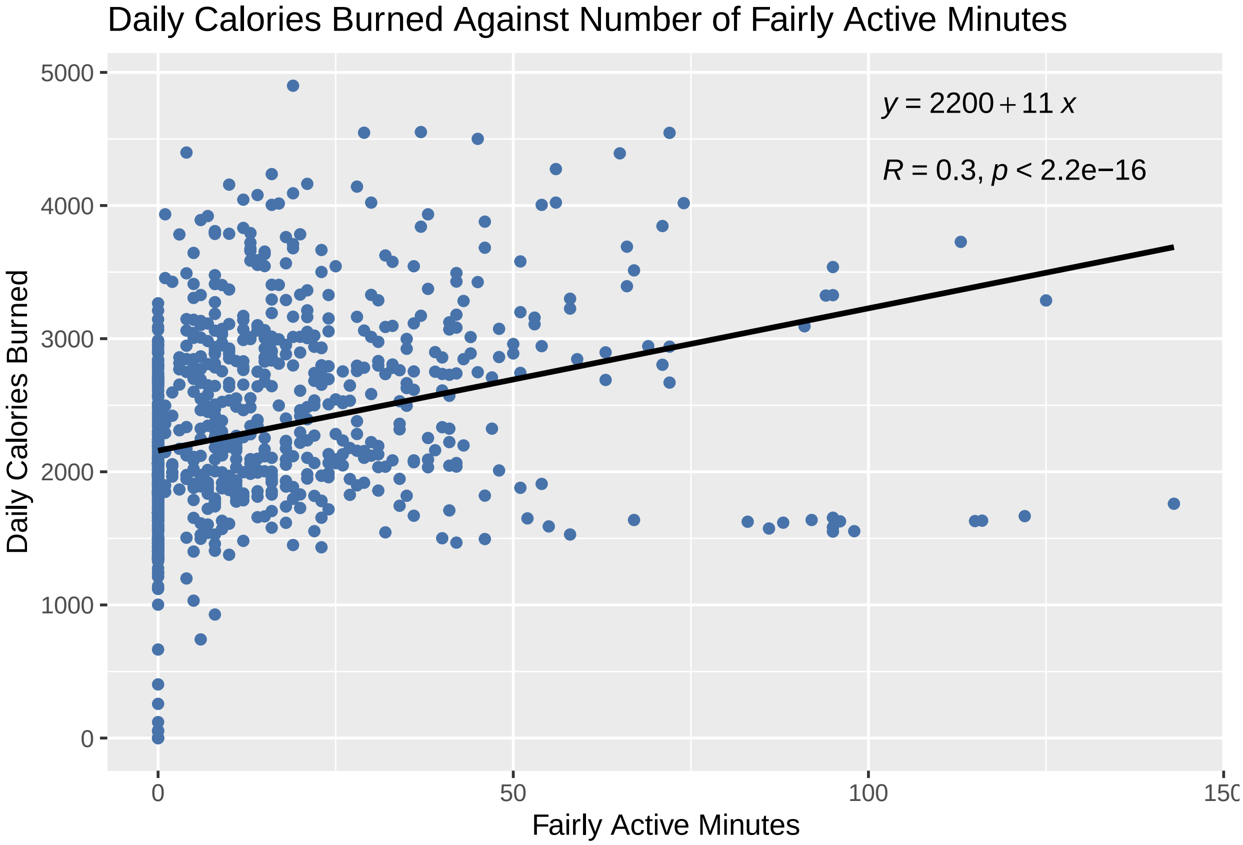
```
##   total_steps    total_distance      calories   
##  Min.   :    0   Min.   : 0.000   Min.   :   0  
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.:1828  
##  Median : 7406   Median : 5.245   Median :2134  
##  Mean   : 7638   Mean   : 5.490   Mean   :2304  
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:2793  
##  Max.   :36019   Max.   :28.030   Max.   :4900
```

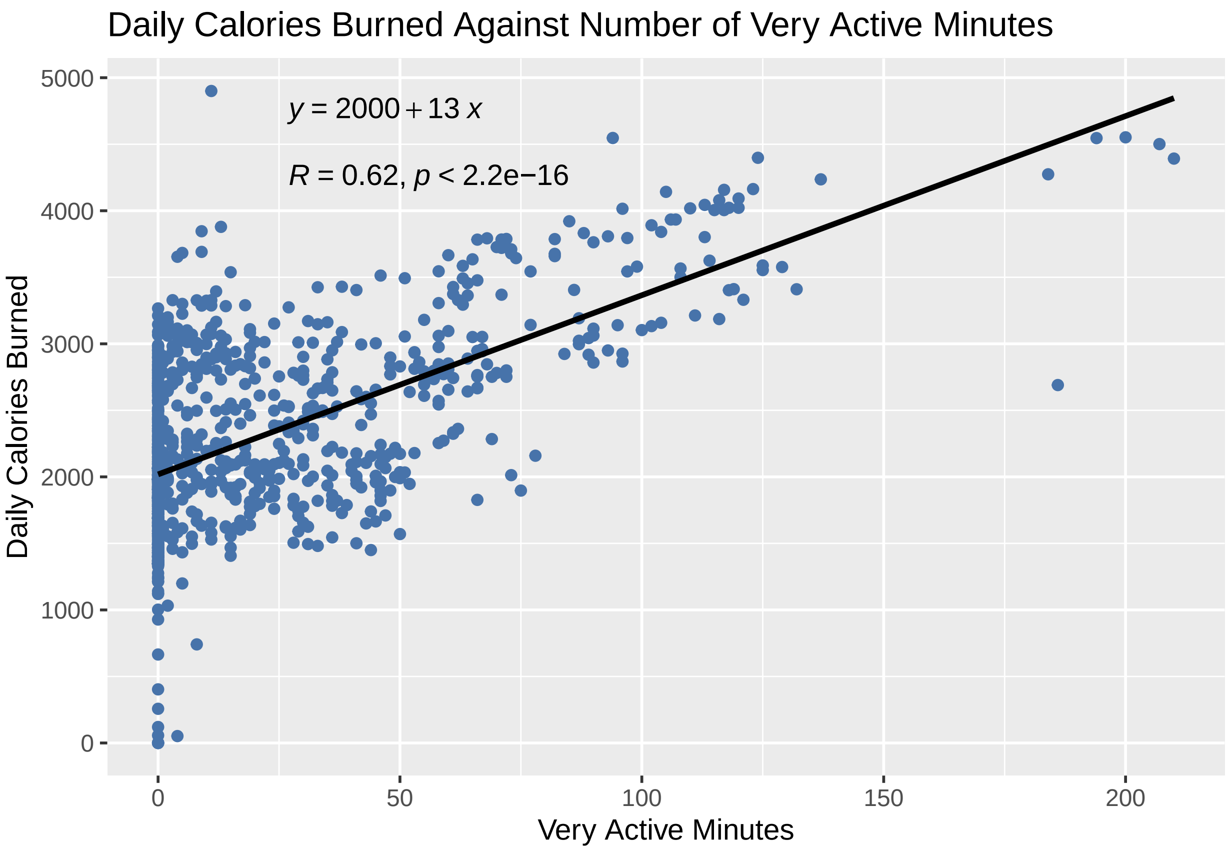
Create a new field in the daily_act_df to total all active minutes for each observation

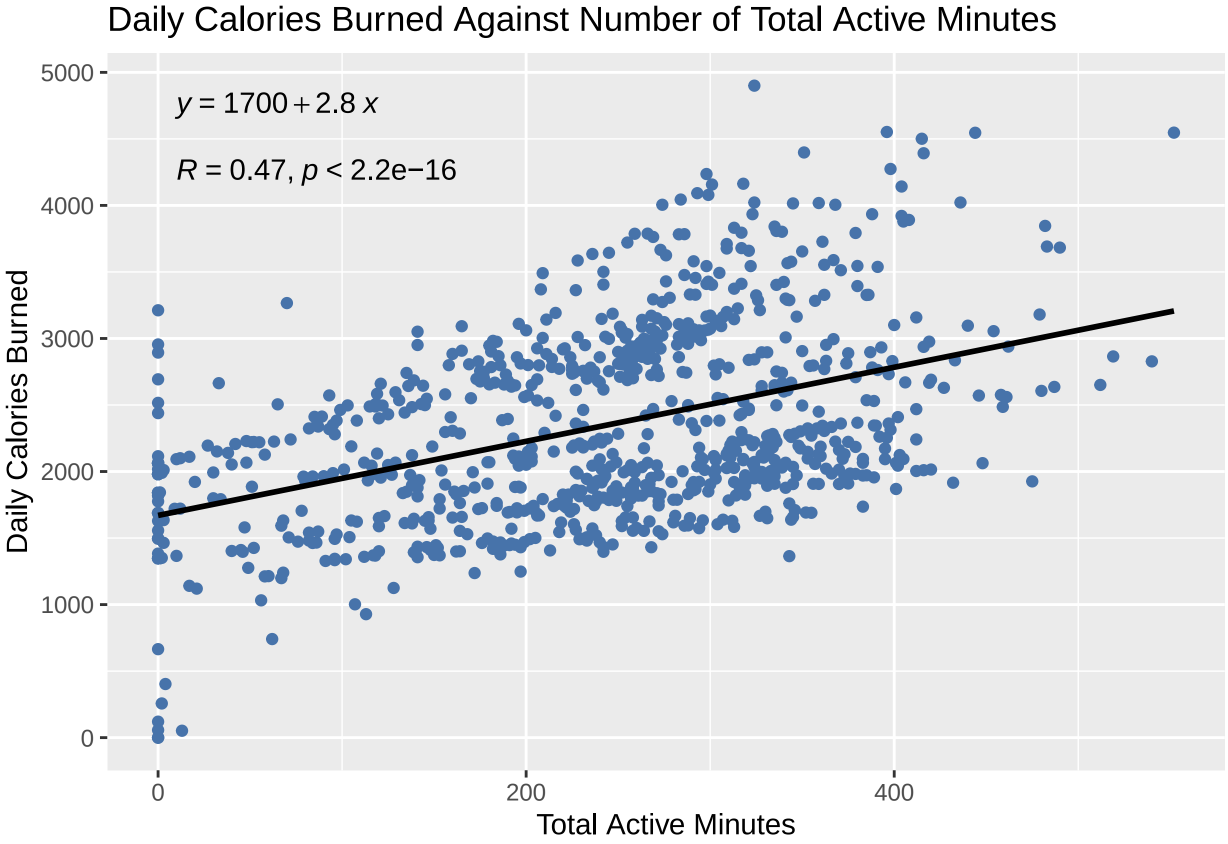
```
daily_act_df$total_active_minutes <- daily_act_df$lightly_active_minutes +
  daily_act_df$fairly_active_minutes + daily_act_df$very_active_minutes
```



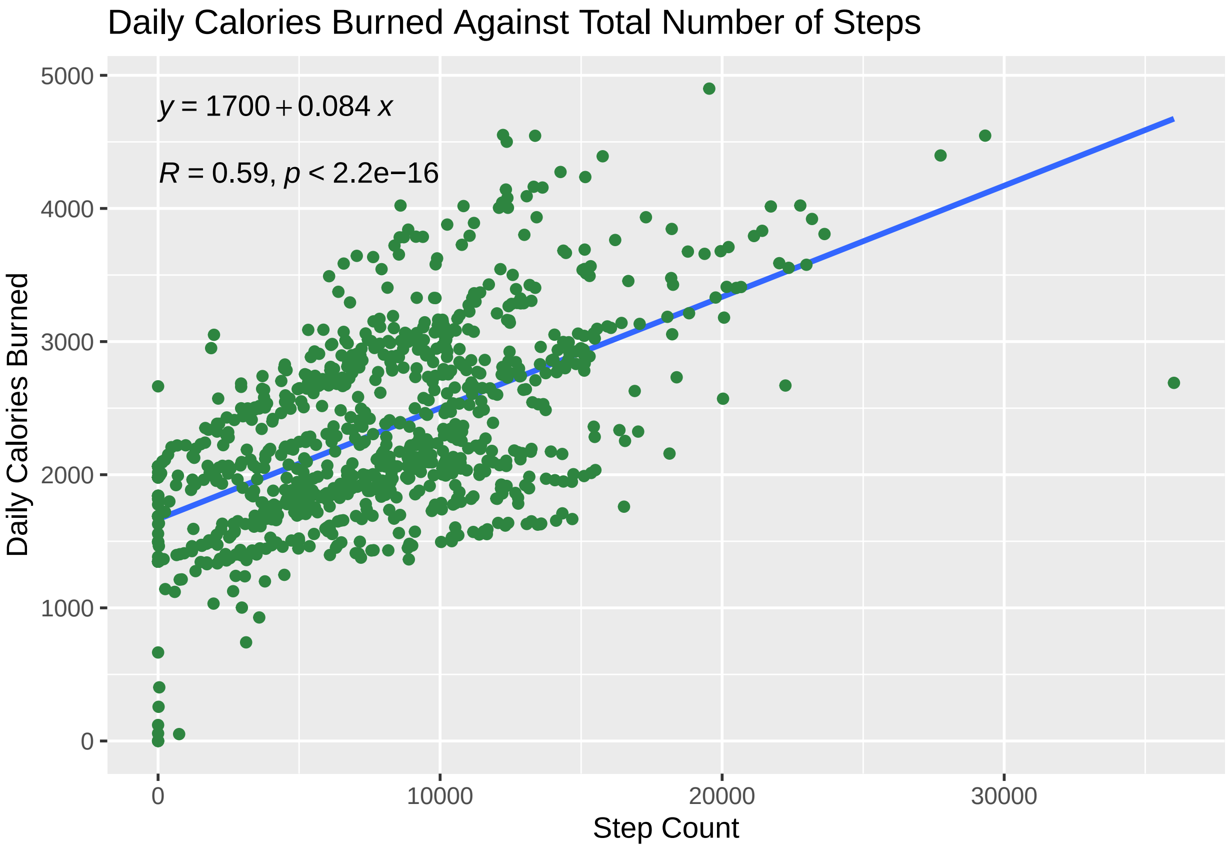

Daily Calories Burned Against Number of Lightly Active Minutes
$y = 1900 + 1.9\,x$
$R = 0.29,\ p < 2.2\text{e}{-16}$
Daily Calories Burned
0
1000
2000
3000
4000
5000
0
100
200
300
400
500
Lightly Active Minutes


Daily Calories Burned Against Number of Fairly Active Minutes
$y = 2200 + 11\,x$
$R = 0.3,\ p < 2.2\text{e}{-16}$
Daily Calories Burned
0
1000
2000
3000
4000
5000
0
50
100
150
Fairly Active Minutes


Daily Calories Burned Against Number of Very Active Minutes


Daily Calories Burned Against Number of Total Active Minutes

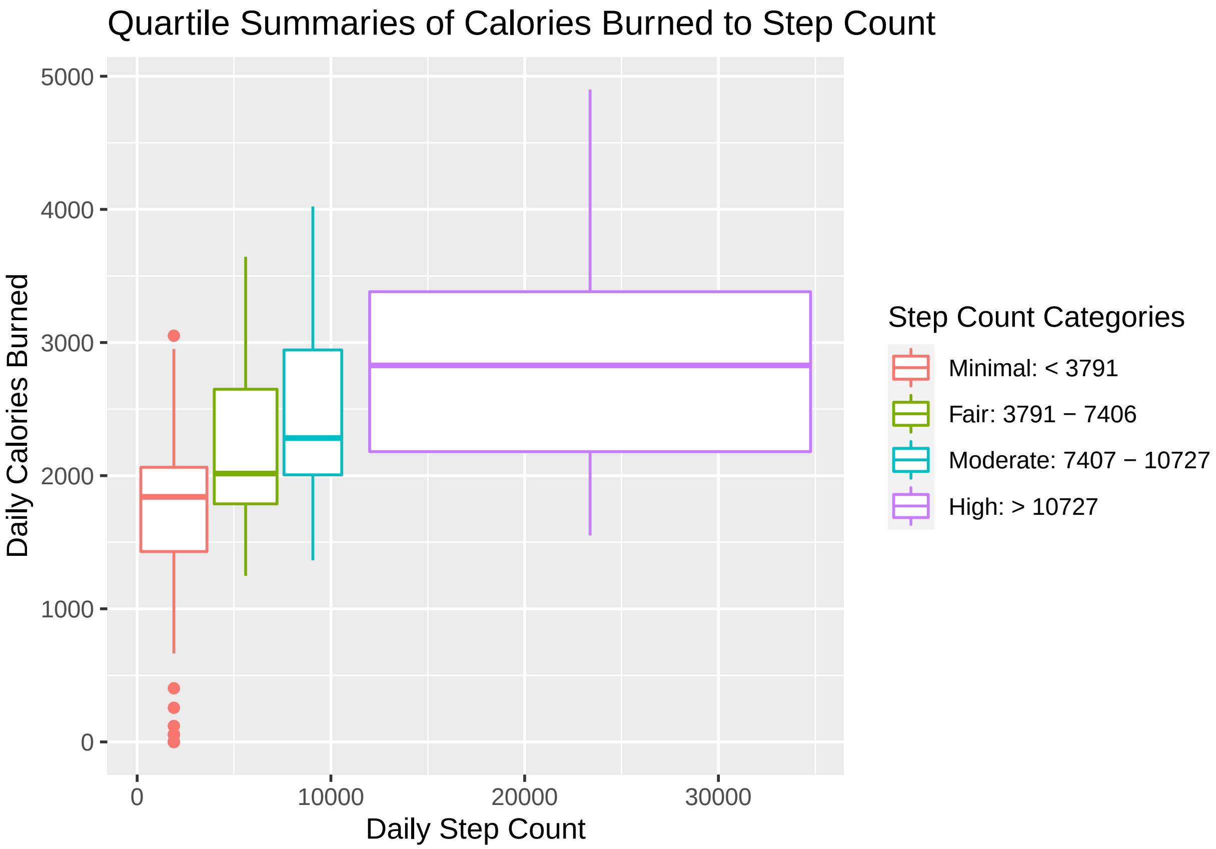
More specifically: A look at step activity


Daily Calories Burned Against Total Number of Steps

Create quartile categories based on number of daily steps

```
daily_act_df <- daily_act_df %>%
  mutate(
    step_act = case_when(
      total_steps <= 3790  ~ "q1_steps",
      total_steps > 3790 & total_steps <= 7406 ~ "q2_steps",
      total_steps > 7406 & total_steps <= 10727 ~ "q3_steps",
      total_steps > 10727  ~ "q4_steps"
  ))

daily_act_df$step_act <- factor(daily_act_df$step_act,
                                levels = c("q1_steps","q2_steps", "q3_steps", "q4_steps"))
```



A look at sleep

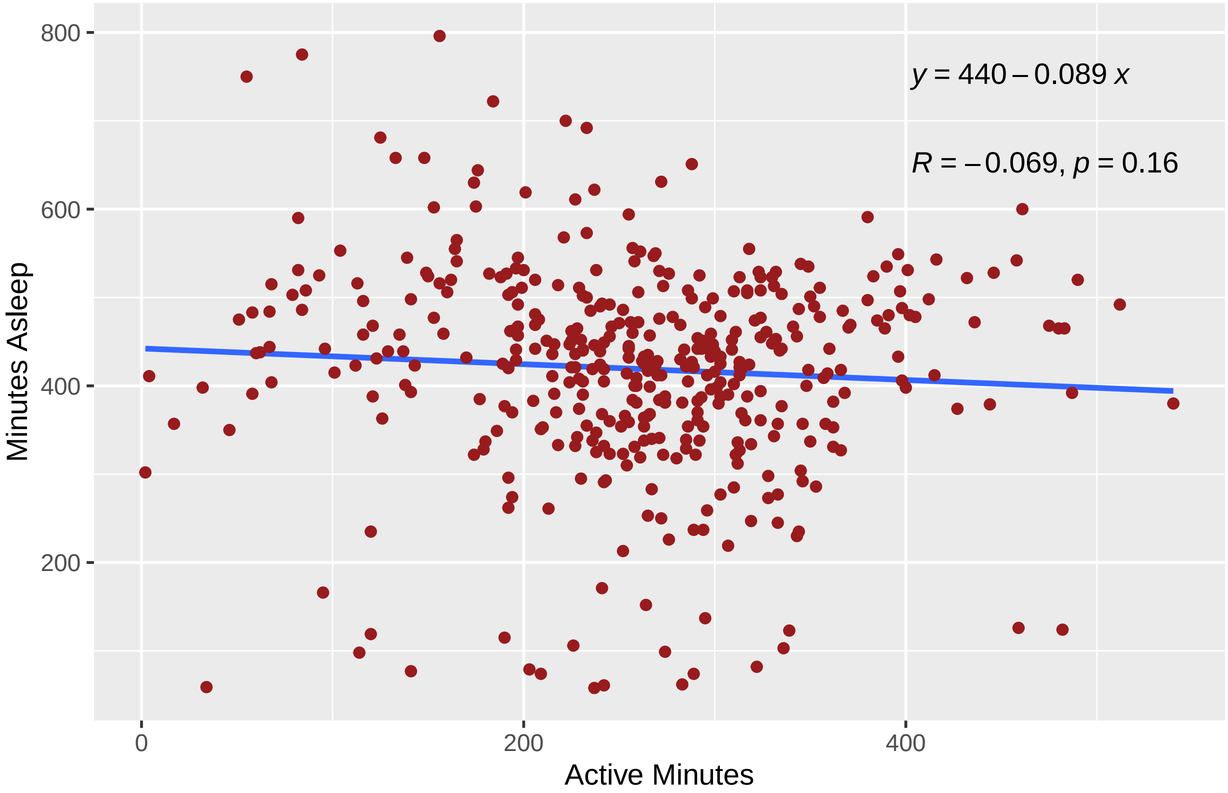
Summary stats for fields in sleep_day_df data frame

```
##  total_minutes_asleep total_time_in_bed
##  Min.   : 58.0        Min.   : 61.0    
##  1st Qu.:361.0        1st Qu.:403.8    
##  Median :432.5        Median :463.0    
##  Mean   :419.2        Mean   :458.5    
##  3rd Qu.:490.0        3rd Qu.:526.0    
##  Max.   :796.0        Max.   :961.0    
```

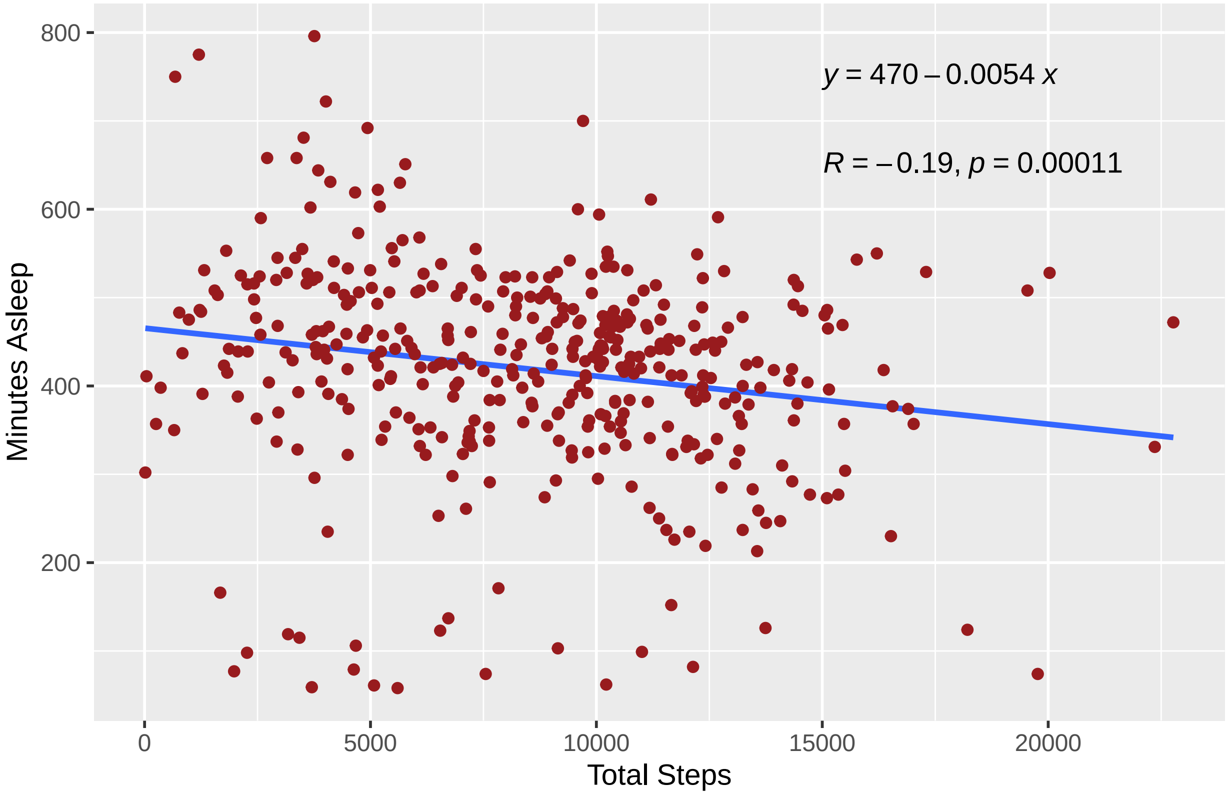
Merge daily_act_df and sleep_day_df

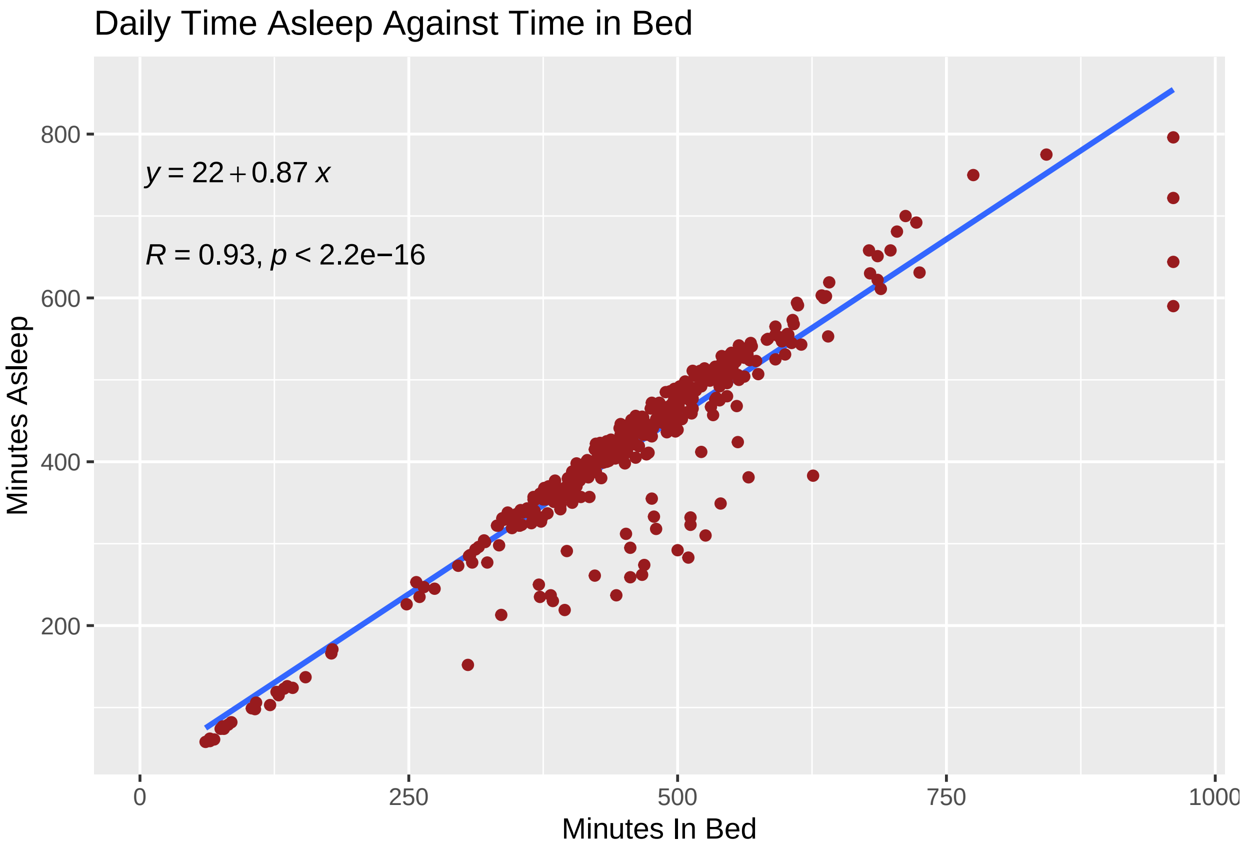
```
comb_daily_act_sleep_df <- merge(daily_act_df, sleep_day_df,
                                 by.x=c("id", "date"), by.y=c("id", "date"))
```

Minutes Asleep Against Daily Active Minutes



Minutes Asleep Against Total Steps





Share and Act

Reminder: The business task was to determine how fitness smart device trends can help influence Bellabeat's marketing strategy for their customers. This task was to be performed with stakeholders Urška Sršen and Sando Mur in mind, who are both co-founders an executives at Bellabeat.

Marketing Strategy Recommendations

Bellabeat Leaf - You won't leave home without it.

1. Results speak! Staying active during the day, including a high step count, increases calorie burn. The longer an individual is sedentary, the lower their calorie burn. The more active they are, the higher their calorie burn. Busy women need a fitness tracker that is out of mind, but getting the job done. Bellabeat Leaf is one less thing to think about and your tracking information is always there when you need it. As the day goes on, find out if you have met your goals or if this is a night to go to the gym or take a trip to the park. And you are not reliant on your phone to be with you and/or charged for Leaf to track!

2. Fitness meets fashion. Bellabeat Leaf will keep track of all of your activity, including step count, so you can stay on track and motivated to reach your goals. Tracking leads to action and it is happening every moment without you having to think about it. The Bellabeat Leaf is fashionable, hypo-allergenic, waterproof, and has a battery that will last 6 months! You will never have to leave it behind because you literally wear it 24 hours a day and it can be worn on a necklace, bracelet, or can be clipped to your clothing.

Suggestions

1. Revisit with more recent data of more users over a longer time period, preferably first party data from Bellabeat users.
2. If not already implemented, be sure the Bellabeat Leaf app allows users to set goals and that the app sends alerts about progress towards completing them.
3. If not already implemented, be sure the Bellabeat Leaf app alerts users when their sleep time is less than optimal, if their sleep is interrupted often during the night, or if there sleep quality if otherwise poor. Women want to feel energized when they wake up, so that they can meet their fitness and health goals.