Bellabeat Case Study

Coursera Google Data Analytics Capstone Project

Author: Cherie Weren

Date: March 8, 2023

Table of Contents

---

1. Project Overview

Bellabeat is a high-tech manufacturer of health-focused smart devices designed exclusively for women. The suite of products includes the Bellabeat app, the Leaf tracker; the Time wellness watch; the Spring water bottle and the Bellabeat membership program.

2. Ask

Business Task

Bellabeat stakeholders have requested an analysis of data from a competing smart device to learn how consumers use non-Bellabeat devices. Apply the insights from the analysis to one Bellabeat device and form recommendations on how these trends could inform the Bellabeat marketing strategy.

Stakeholders

- Urška Sršen - Cofounder and Chief Creative Officer
- Sando Mur - Bofounder, Mathematician and key member of executive team
- Marketing analytics team - A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeats marketing strategy

3. Prepare

Data Used

The data set used is from Kaggle, FitBit Fitness Tracker Data supplied by Mobius. The data was collected using a survey with 30 people submitting their data.

Privacy & Data Information

Verifying the metadata of our dataset we can confirm it is open-source. The owner has dedicated the work to the public domain by waiving all of his or her rights to the work worldwide under copyright law, including all related and neighboring rights, to the extent allowed by law. You can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission.

There is a sampling bias, even though the minimum required is 30, it will be preferable to have more data available to do a more in depth analysis. The data does not indicate who the demographic is and was only taken over the span of 31 days, from 12 April 2016 to 12 May 2016.

Data Verification

I used the following three datasets provided as they contain data that is not currently collected by Bellabeat:

dailyActivity_merged - The file contains data from 33 users tracked over 31 days. It includes User ID, ActivityDays, total steps, total distance, calories, Tracker Distance, logged activities distance and very/moderately/light/sedentary Active distance & minutes. This data has everything required to do the current analysis. No duplicates or whitespaces were encountered.

dailyCalories_merged - The file contains data from 33 users over 31 days, each logged calories burned at 1 hour intervals. The data includes user ID, Day and hour of the entry and the amount of calories burned in the hour.

dailyIntensities_merged - The file contains data from 33 users over 31 days. Each logs the amount of daily activity. The data includes user id, date and number of minutes that the user was active divided between four categories: sedentary, lightly active, fairly active and very active.

weightLogInfo_merged - The file contains data from 8 users over 31 days and inlcudes BMI, Weight (pounds and kilograms) and body fat percentage.

4. Process

R was used for the analysis.

Install and Load Libraries

Install and load the required libraries.

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.4.0      ✓ purrr   1.0.1
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.5.0
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
```

```
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
library(lubridate)
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
library(janitor)
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
library(reshape)
##
## Attaching package: 'reshape'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
##
## The following object is masked from 'package:dplyr':
##
##     rename
```

```
##
## The following objects are masked from 'package:tidyr':
##
##     expand, smiths
```

```
library(ggplot2)
```

```
library(dplyr)
```

Load Datasets

```
activity <- read_csv('/kaggle/input/fitbit/Fitabase Data 4.12.16-
5.12.16/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.
```

```
calories <- read_csv('/kaggle/input/fitbit/Fitabase Data 4.12.16-
5.12.16/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.
```

```
intensity <- read_csv('/kaggle/input/fitbit/Fitabase Data 4.12.16-
5.12.16/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.
```

```
weight <- read_csv('/kaggle/input/fitbit/Fitabase Data 4.12.16-
5.12.16/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.
```

Preview Data

View first ten rows of imported data to become familiar with table structures, variable names, data types and discover what cleaning needs to be done.

***Activity* dataset**

The activity dataset contains total number of steps, distance traveled with level of intensity, minutes of training with level of intensity, total calories burned as well a user category with three user types.

```
head(activity,10)
```

```
## # A tibble: 10 × 15

##            Id Activity…¹ Total…² Total…³ Track…⁴ Logge…⁵ VeryA…⁶ Moder…⁷ Light…⁸

##         <dbl> <chr>        <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>

##  1 1503960366 4/12/2016    13162    8.5     8.5        0    1.88   0.550    6.06
```

```
##  2 1503960366 4/13/2016    10735    6.97    6.97       0    1.57   0.690    4.71
##  3 1503960366 4/14/2016    10460    6.74    6.74       0    2.44   0.400    3.91
##  4 1503960366 4/15/2016     9762    6.28    6.28       0    2.14   1.26     2.83
##  5 1503960366 4/16/2016    12669    8.16    8.16       0    2.71   0.410    5.04
##  6 1503960366 4/17/2016     9705    6.48    6.48       0    3.19   0.780    2.51
##  7 1503960366 4/18/2016    13019    8.59    8.59       0    3.25   0.640    4.71
##  8 1503960366 4/19/2016    15506    9.88    9.88       0    3.53   1.32     5.03
##  9 1503960366 4/20/2016    10544    6.68    6.68       0    1.96   0.480    4.24
## 10 1503960366 4/21/2016     9819    6.34    6.34       0    1.34   0.350    4.65
## # … with 6 more variables: SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>, and
## #   abbreviated variable names ¹ActivityDate, ²TotalSteps, ³TotalDistance,
## #   ⁴TrackerDistance, ⁵LoggedActivitiesDistance, ⁶VeryActiveDistance,
## #   ⁷ModeratelyActiveDistance, ⁸LightActiveDistance
str(activity)
## spc_tbl_ [940 × 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>
any(is.na.data.frame(activity))
## [1] FALSE
```

***Calories* dataset**

Calories dataset contains observations about calories burned by users each day.

```
head(calories, 10)
```

```
## # A tibble: 10 × 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
##  7 1503960366 4/18/2016       1921
##  8 1503960366 4/19/2016       2035
##  9 1503960366 4/20/2016       1786
## 10 1503960366 4/21/2016       1775
```

```
str(calories)
```

```
## spc_tbl_ [940 × 3] (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 ...
##  $ ActivityDay: chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ Calories   : num [1:940] 1985 1797 1776 1745 1863 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDay = col_character(),
##   ..   Calories = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
any(is.na(calories))
```

```
## [1] FALSE
```

***Intensity* dataset**

The intensity dataset contains observations about the intensity of training including length of time and distance of training.

```
head(intensity)
```

```
## # A tibble: 6 × 10
##       Id Activ…¹ Seden…² Light…³ Fairl…⁴ VeryA…⁵ Seden…⁶ Light…⁷ Moder…⁸ VeryA…⁹
##    <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1.50e9 4/12/2…     728     328      13      25       0    6.06   0.550    1.88
## 2 1.50e9 4/13/2…     776     217      19      21       0    4.71   0.690    1.57
## 3 1.50e9 4/14/2…    1218     181      11      30       0    3.91   0.400    2.44
## 4 1.50e9 4/15/2…     726     209      34      29       0    2.83   1.26     2.14
## 5 1.50e9 4/16/2…     773     221      10      36       0    5.04   0.410    2.71
## 6 1.50e9 4/17/2…     539     164      20      38       0    2.51   0.780    3.19
## # … with abbreviated variable names ¹ActivityDay, ²SedentaryMinutes,
## #   ³LightlyActiveMinutes, ⁴FairlyActiveMinutes, ⁵VeryActiveMinutes,
## #   ⁶SedentaryActiveDistance, ⁷LightActiveDistance, ⁸ModeratelyActiveDistance,
## #   ⁹VeryActiveDistance
```

```
str(intensity)
```

```
## spc_tbl_ [940 × 10] (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
...
##  $ ActivityDay              : chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016"
"4/15/2016" ...
##  $ SedentaryMinutes         : num [1:940] 728 776 1218 726 773 ...
##  $ LightlyActiveMinutes     : num [1:940] 328 217 181 209 221 164 233 264 205 211
...
##  $ FairlyActiveMinutes      : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
##  $ VeryActiveMinutes        : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
##  $ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ LightActiveDistance      : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
```

```
##  $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
##  $ VeryActiveDistance      : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDay = col_character(),
##   ..   SedentaryMinutes = col_double(),
##   ..   LightlyActiveMinutes = col_double(),
##   ..   FairlyActiveMinutes = col_double(),
##   ..   VeryActiveMinutes = col_double(),
##   ..   SedentaryActiveDistance = col_double(),
##   ..   LightActiveDistance = col_double(),
##   ..   ModeratelyActiveDistance = col_double(),
##   ..   VeryActiveDistance = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
any(is.na(intensity))
## [1] FALSE
```

***Weight* dataset**

This dataset stores information about weight, bmi and fat percentage of the user.

```
head(weight)
## # A tibble: 6 × 8
##           Id Date                  WeightKg Weight…¹   Fat   BMI IsMan…²   LogId
##        <dbl> <chr>                    <dbl>    <dbl> <dbl> <dbl> <lgl>     <dbl>
## 1 1503960366 5/2/2016 11:59:59 PM      52.6     116.    22  22.6 TRUE    1.46e12
## 2 1503960366 5/3/2016 11:59:59 PM      52.6     116.    NA  22.6 TRUE    1.46e12
## 3 1927972279 4/13/2016 1:08:52 AM     134.      294.    NA  47.5 FALSE   1.46e12
## 4 2873212765 4/21/2016 11:59:59 PM     56.7     125.    NA  21.5 TRUE    1.46e12
```

```
## 5 2873212765 5/12/2016 11:59:59 PM     57.3     126.    NA  21.7 TRUE    1.46e12

## 6 4319703577 4/17/2016 11:59:59 PM     72.4     160.    25  27.5 TRUE    1.46e12

## # … with abbreviated variable names ¹WeightPounds, ²IsManualReport

str(weight)

## spc_tbl_ [67 × 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)

##  $ Id            : num [1:67] 1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...

##  $ Date          : chr [1:67] "5/2/2016 11:59:59 PM" "5/3/2016 11:59:59 PM"
"4/13/2016 1:08:52 AM" "4/21/2016 11:59:59 PM" ...

##  $ WeightKg      : num [1:67] 52.6 52.6 133.5 56.7 57.3 ...

##  $ WeightPounds  : num [1:67] 116 116 294 125 126 ...

##  $ Fat           : num [1:67] 22 NA NA NA NA 25 NA NA NA NA ...

##  $ BMI           : num [1:67] 22.6 22.6 47.5 21.5 21.7 ...

##  $ IsManualReport: logi [1:67] TRUE TRUE FALSE TRUE TRUE TRUE ...

##  $ LogId         : num [1:67] 1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...

##  - attr(*, "spec")=

##   .. cols(

##   ..   Id = col_double(),

##   ..   Date = col_character(),

##   ..   WeightKg = col_double(),

##   ..   WeightPounds = col_double(),

##   ..   Fat = col_double(),

##   ..   BMI = col_double(),

##   ..   IsManualReport = col_logical(),

##   ..   LogId = col_double()

##   .. )

##  - attr(*, "problems")=<externalptr>

sum(is.na(weight))

## [1] 65
```

Data Cleaning and Formatting

Count unique users

```
n_distinct(activity$Id)
```

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

```
n_distinct(calories$Id)
```

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

```
n_distinct(intensity$Id)
```

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

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

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

There are 33 participants in all datasets except for weight. As only 8 users participated and there are 65 NA values, it is determined the weight dataset is too small and incomplete to be used in our analysis.

Search for duplicates

```
sum(duplicated(activity))
```

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

```
sum(duplicated(calories))
```

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

```
sum(duplicated(intensity))
```

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

Clean duplicates

```
calories <- calories %>%
```

```
  distinct
```

Verify

```
sum(duplicated(calories))
```

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

Formatting datatype in column *ActivityDate* (from *char* to *datetime*)

```
activity$ActivityDate <- as.POSIXct(activity$ActivityDate, format = '%m/%d/%Y',
tz=Sys.timezone())
```

Format datatype in *ActivityDay* from *char* to *datetime*

```
calories$ActivityDay <- as.POSIXct(calories$ActivityDay, format = '%m/%d/%Y',
tz=Sys.timezone())
```

Format *ActivityDay* type from *char* to *datetime*

```
intensity$ActivityDay <- as.POSIXct(intensity$ActivityDay, format = '%m/%d/%Y',
tz=Sys.timezone())
```

Clean column names for variable consistency; rename column in activity data

```
activity <- clean_names(activity)

calories <- clean_names(calories)

intensity <- clean_names(intensity)


activity <- activity %>%

    dplyr::rename(activity_day = activity_date)
```

Summary of each dataset

```
activity %>%

  select(id, total_steps, total_distance, sedentary_minutes) %>%

  summary()

##        id             total_steps    total_distance   sedentary_minutes

##  Min.   :1.504e+09   Min.   :    0   Min.   : 0.000   Min.   :   0.0

##  1st Qu.:2.320e+09   1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 729.8

##  Median :4.445e+09   Median : 7406   Median : 5.245   Median :1057.5

##  Mean   :4.855e+09   Mean   : 7638   Mean   : 5.490   Mean   : 991.2

##  3rd Qu.:6.962e+09   3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:1229.5

##  Max.   :8.878e+09   Max.   :36019   Max.   :28.030   Max.   :1440.0

calories %>%

  select(calories) %>%

  summary()

##     calories

##  Min.   :   0

##  1st Qu.:1828

##  Median :2134

##  Mean   :2304
```

```
##  3rd Qu.:2793

##  Max.   :4900

intensity %>%

  select(sedentary_minutes, lightly_active_minutes, fairly_active_minutes,
very_active_minutes) %>%

  summary()

##  sedentary_minutes lightly_active_minutes fairly_active_minutes

##  Min.   :   0.0    Min.   :  0.0          Min.   :  0.00

##  1st Qu.: 729.8    1st Qu.:127.0          1st Qu.:  0.00

##  Median :1057.5    Median :199.0          Median :  6.00

##  Mean   : 991.2    Mean   :192.8          Mean   : 13.56

##  3rd Qu.:1229.5    3rd Qu.:264.0          3rd Qu.: 19.00

##  Max.   :1440.0    Max.   :518.0          Max.   :143.00

##  very_active_minutes

##  Min.   :  0.00

##  1st Qu.:  0.00

##  Median :  4.00

##  Mean   : 21.16

##  3rd Qu.: 32.00

##  Max.   :210.00
```

Merge hourly_calories & hourly_steps to see if there are any correlations. The data will be merged with the id and date fields as their primary keys.

```
calories_activity <- merge(calories, activity, by=c ("id", "activity_day"))

n_distinct(calories_activity$id)

## [1] 33

head(calories_activity)

##           id activity_day calories.x total_steps total_distance

## 1 1503960366   2016-04-12       1985       13162           8.50

## 2 1503960366   2016-04-13       1797       10735           6.97
```

```
## 3 1503960366   2016-04-14       1776       10460           6.74
## 4 1503960366   2016-04-15       1745        9762           6.28
## 5 1503960366   2016-04-16       1863       12669           8.16
## 6 1503960366   2016-04-17       1728        9705           6.48
##   tracker_distance logged_activities_distance very_active_distance
## 1             8.50                         0                 1.88
## 2             6.97                         0                 1.57
## 3             6.74                         0                 2.44
## 4             6.28                         0                 2.14
## 5             8.16                         0                 2.71
## 6             6.48                         0                 3.19
##   moderately_active_distance light_active_distance sedentary_active_distance
## 1                       0.55                  6.06                         0
## 2                       0.69                  4.71                         0
## 3                       0.40                  3.91                         0
## 4                       1.26                  2.83                         0
## 5                       0.41                  5.04                         0
## 6                       0.78                  2.51                         0
##   very_active_minutes fairly_active_minutes lightly_active_minutes
## 1                  25                    13                   328
## 2                  21                    19                   217
## 3                  30                    11                   181
## 4                  29                    34                   209
## 5                  36                    10                   221
## 6                  38                    20                   164
##   sedentary_minutes calories.y
## 1               728       1985
## 2               776       1797
```

```
## 3              1218       1776

## 4               726       1745

## 5               773       1863

## 6               539       1728
```

5. Analyze

To start the analysis, we will first look at activity to determine how the users are categorized.

Activity and Steps

The following resources from MedicineNet, 10000Steps and various other sources conclude the following:

- Sedentary is less than 5,000 steps per day
- Low active is 5,000 to 7,499 steps per day
- Somewhat active is 7,500 to 9,999 steps per day
- Active is more than 10,000 steps per day
- Highly active is more than 12,500 per day

We only have four activity categories in our data, so we will need to adjust our categories from the information we have. After which we will calculate the average steps per user and group each into their respective activity level.

```
average_daily_steps <- calories_activity %>%

  group_by(id) %>%

  summarise(average_daily_steps = mean(total_steps))



head(average_daily_steps)

## # A tibble: 6 × 2

##           id average_daily_steps

##        <dbl>               <dbl>

## 1 1503960366              12117.

## 2 1624580081               5744.

## 3 1644430081               7283.

## 4 1844505072               2580.

## 5 1927972279                916.

## 6 2022484408              11371.

user_activity <- average_daily_steps %>%
```

```
  mutate(user_activity = case_when(

    average_daily_steps < 5000 ~ "Sedentary - < 5,000",

    average_daily_steps >= 5000 & average_daily_steps < 7499 ~ "Lightly Active -
5,000-7,499",

    average_daily_steps >= 7500 & average_daily_steps < 9999 ~ "Fairly Active - 7,500
- 9,999",

    average_daily_steps >= 10000 ~ "Very Active - >10,000"

  ))


head(user_activity)

## # A tibble: 6 × 3

##           id average_daily_steps user_activity

##        <dbl>               <dbl> <chr>

## 1 1503960366              12117. Very Active - >10,000

## 2 1624580081               5744. Lightly Active - 5,000-7,499

## 3 1644430081               7283. Lightly Active - 5,000-7,499

## 4 1844505072               2580. Sedentary - < 5,000

## 5 1927972279                916. Sedentary - < 5,000

## 6 2022484408              11371. Very Active - >10,000
```

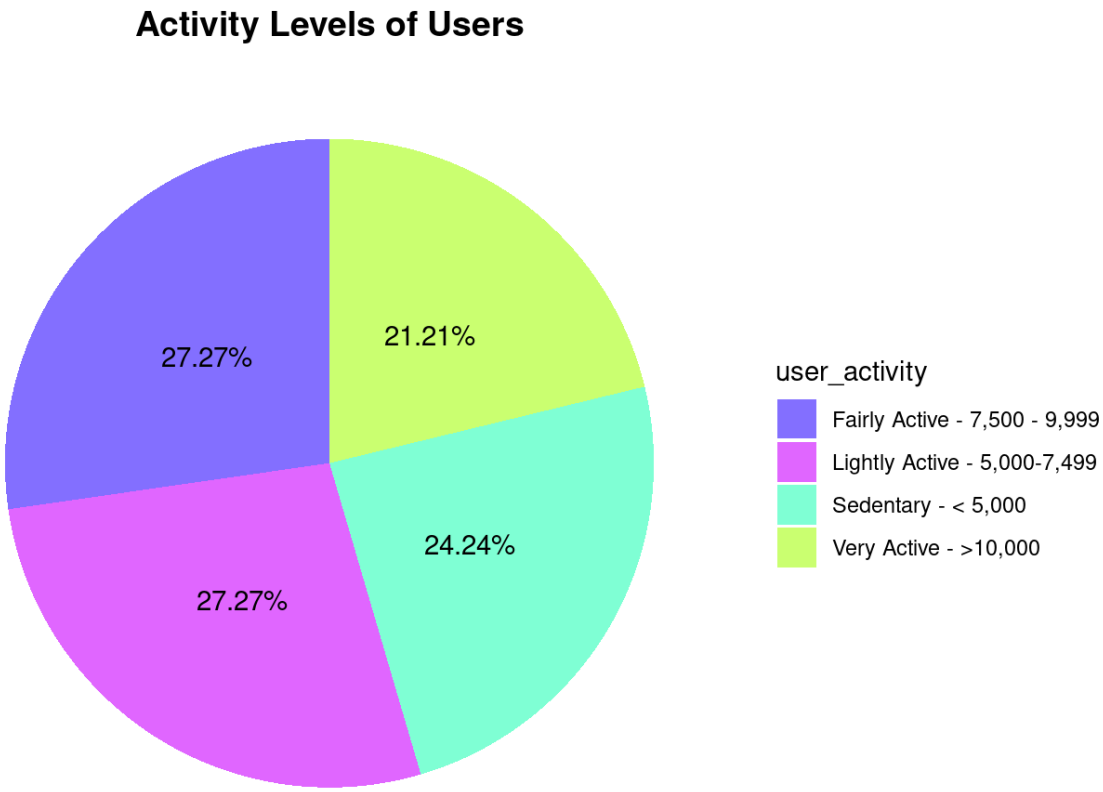
We will visualize the categorized users as percentages for ease of use in visualization.

```
user_activity_percent <- user_activity %>%

  group_by(user_activity) %>%

  summarise(cnt = n()) %>%

  mutate(percent_value = formattable::percent(cnt / sum(cnt)))


head(user_activity_percent)

## # A tibble: 4 × 3

##   user_activity                  cnt percent_value

##   <chr>                        <int> <formttbl>
```

```
## 1 Fairly Active - 7,500 - 9,999     9 27.27%

## 2 Lightly Active - 5,000-7,499      9 27.27%

## 3 Sedentary - < 5,000               8 24.24%

## 4 Very Active - >10,000             7 21.21%
```

Create visualization to show user activity.

```
user_activity_percent %>%

    ggplot(aes(x="",y=percent_value, fill=user_activity)) +

    geom_bar(stat = "identity", width = 1) +

    coord_polar("y", start=0)+

    theme_minimal()+

    theme(axis.title.x= element_blank(),

          axis.title.y = element_blank(),

          panel.border = element_blank(),

          panel.grid = element_blank(),

          axis.ticks = element_blank(),

          axis.text.x = element_blank(),

          plot.title = element_text(hjust = 0.5, size=14, face = "bold"))+

    scale_fill_manual(values = c("slateblue1", "mediumorchid1", "aquamarine1",
"darkolivegreen1")) +

  geom_text(aes(label= percent_value),

            position = position_stack(vjust = 0.5))+

  labs(title = "Activity Levels of Users")
```

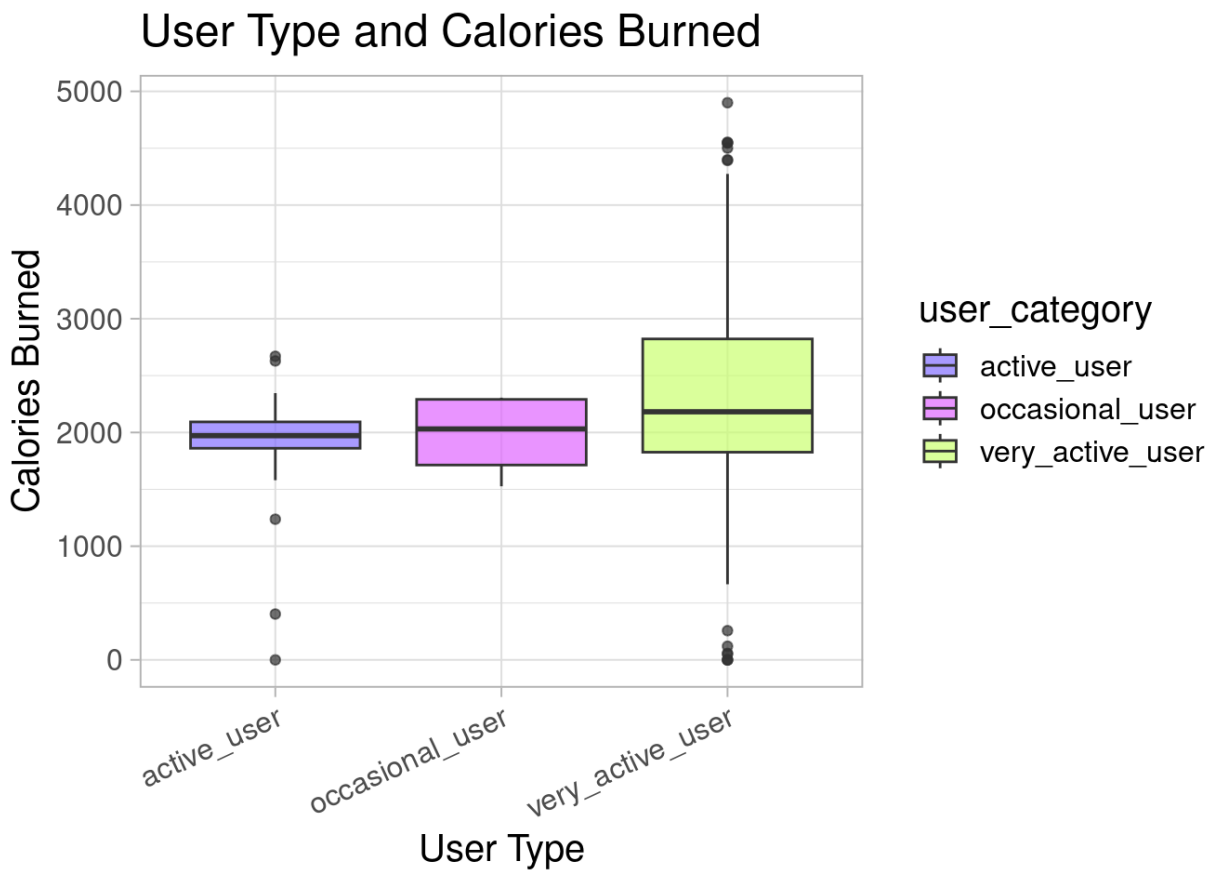


This chart shows that the use of smart devices is evenly distributed between users with all types of activity levels with very active being slightly lower.

Activity Levels and Calories Burned

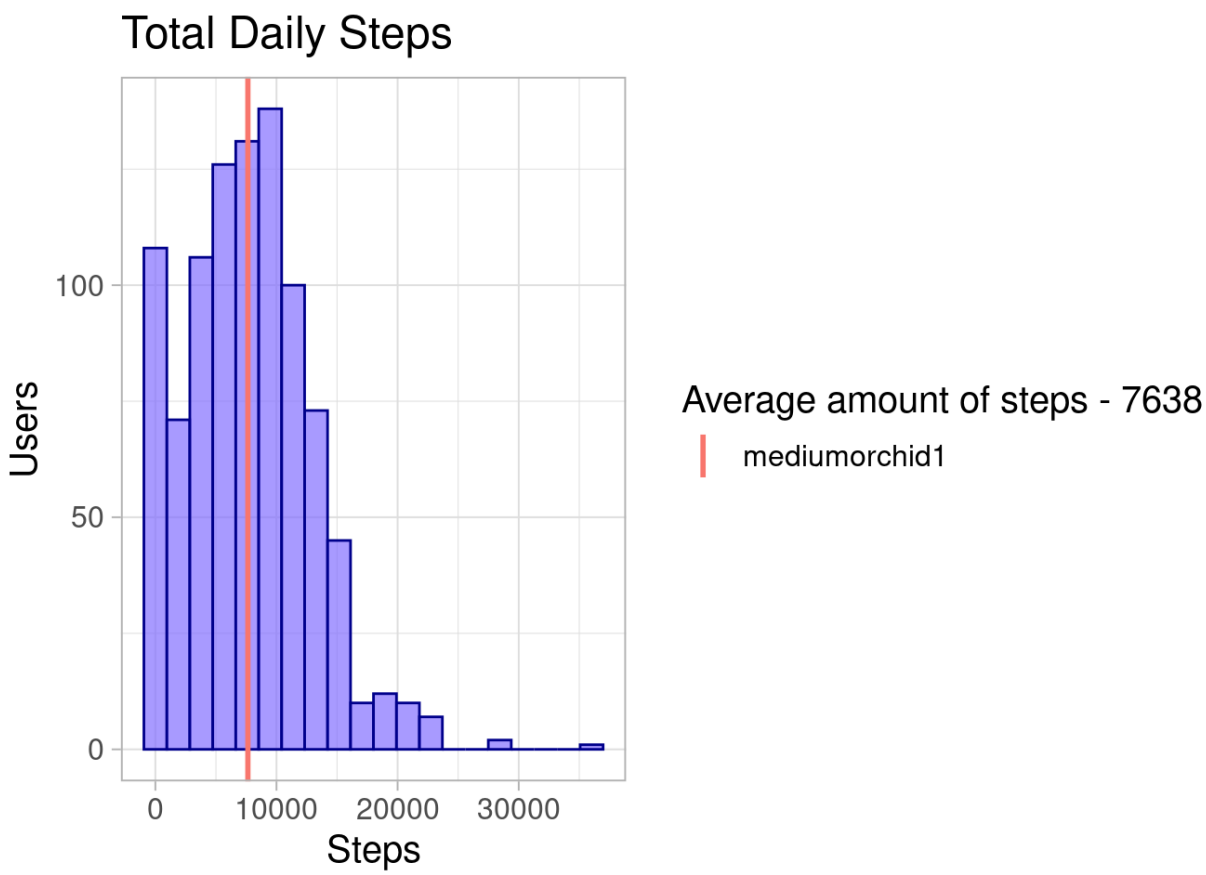
```
activity_by_id <- activity %>%

    group_by(id) %>%

    count(id)

activity_by_id <- activity_by_id %>% mutate(user_category = case_when(

        n >= 25 ~ 'very_active_user',

        n >= 10 ~ 'active_user',

        n < 10 ~ 'occasional_user'))

activity <- merge(x=activity, y=activity_by_id[ , c("id", "n", "user_category")],
by='id')

options(repr.plot.width=10, repr.plot.height=10)

ggplot(data=activity)+

    geom_boxplot(aes(x=user_category, y=calories, fill=user_category), alpha=0.7)+

        theme_light(base_size=15)+
```

```
        guides(x =  guide_axis(angle = 25)) +

        labs(x="User Type", y="Calories Burned", title="User Type and Calories
Burned")+


scale_fill_manual(values=c("very_active_user"="darkolivegreen1","active_user"="slateb
lue1","occasional_user"="mediumorchid1"))
```



User activty chart shows that most active users burn more calories on average than other users.

```
options(repr.plot.width=10, repr.plot.height=10)

ggplot(data=activity, aes(x=total_steps))+

    geom_histogram(bins = 20, color='darkblue', fill='slateblue1', alpha=0.7)+

    geom_vline(data=activity, aes(xintercept=mean(total_steps),
color='mediumorchid1'), size=1)+

    theme_light(base_size=15) +

    labs(color = "Average amount of steps - 7638", x= "Steps", y= "Users",
title="Total Daily Steps")

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.

## ℹ Please use `linewidth` instead.
```



Total steps per day chart is right skewed with most 10,000 or less with average right at 7500.

Are users that are taking more steps burning more calories?

Calculate correlation of calories to steps:

```
round(cor(activity$calories, activity$total_steps, method="pearson"),2)
```

```
## [1] 0.59
```

Chart calories/steps:

```
options(repr.plot.widht=10, repr.plot.height=10)
```

```
ggplot(data=activity, aes(x=calories, y=total_steps, color=calories))+
    geom_point()+
    geom_smooth(method = 'loess', formula = y ~ x, color = 'aquamarine1')+
    theme_light(base_size=15)+
    scale_color_gradient(low='darkolivegreen1', high='mediumorchid1')+
    labs(x='Calories Burned', y='Steps', title='Daily Steps and Calories Burned')
```

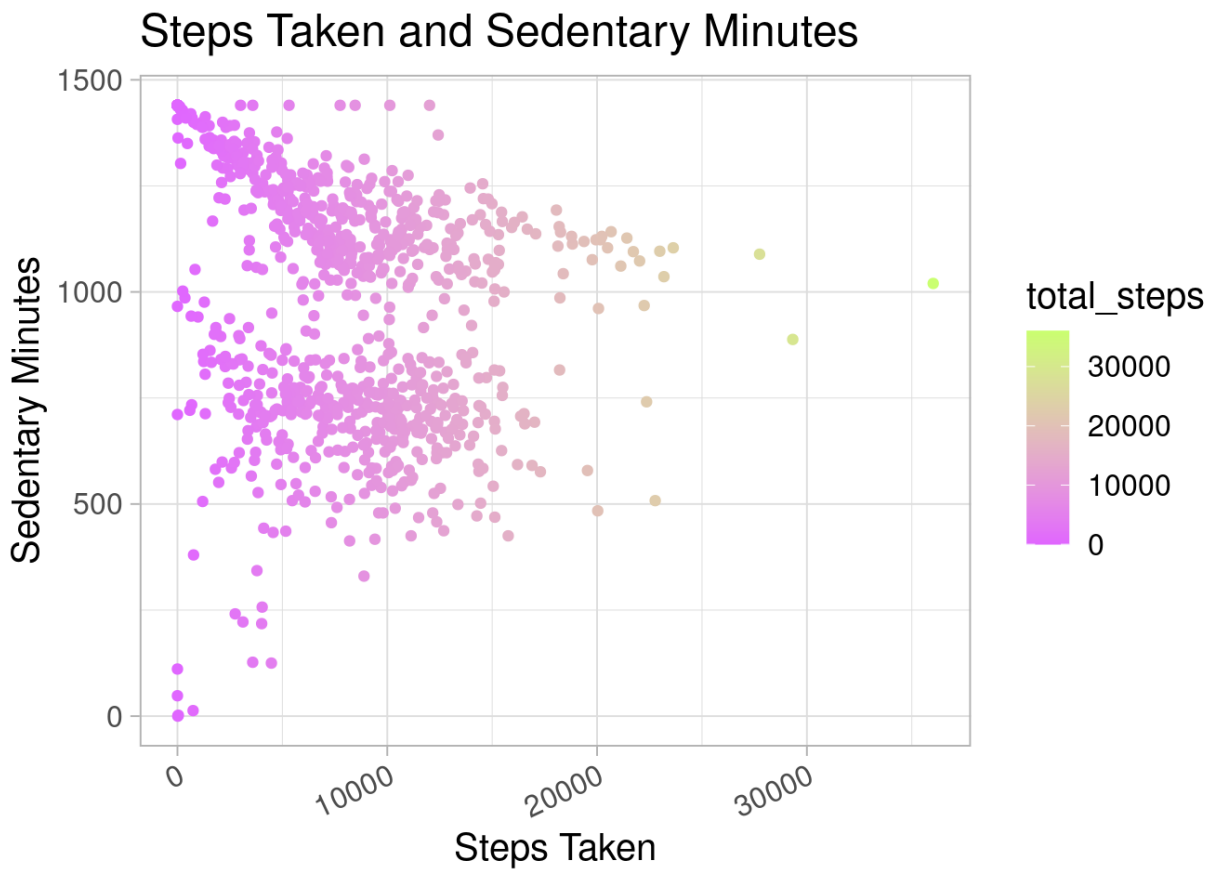
The correlation between calories burned and recorded steps of .59 which indicates there is a moderate linear dependency between these two variables. We can conclude that users burn calories during other types of activities as well.

Steps Taken and Sedentary Minutes

```
activity %>%

  select(total_steps,

         total_distance,

         sedentary_minutes) %>%

  summary()
```

```
##    total_steps    total_distance   sedentary_minutes

##  Min.   :    0   Min.   : 0.000   Min.   :   0.0

##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 729.8

##  Median : 7406   Median : 5.245   Median :1057.5

##  Mean   : 7638   Mean   : 5.490   Mean   : 991.2

##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:1229.5

##  Max.   :36019   Max.   :28.030   Max.   :1440.0
```

```
ggplot(data=activity, aes(x=total_steps, y=sedentary_minutes, colour = total_steps))+
  geom_point()+
  guides(x = guide_axis(angle=25))+
  theme_light(base_size=15)+
  scale_color_gradient(low='mediumorchid1', high='darkolivegreen1')+
    labs(x="Steps Taken", y = "Sedentary Minutes", title = "Steps Taken and Sedentary Minutes")
```



A moderate negative correlation exists between steps and sedentary minutes with two distinct clusters at 750 sedentary minutes and above 1000 sedentary minutes.

6. Recommendations for Bellabeat App

Note that the data used was from a small sample size without any demographic information, therefore, these recommendations are high level and could be further substantiated with additional analysis using more comprehensive datasets.

Data used for the analysis was from FitBit, a wearable fitness tracking device with many different model types and features. No description of the type of FitBit device was provided. Like Bellabeat, FitBit also offers an app.

For this analysis, I will assume that users utilize both the wearable device and the app for both FitBit and Bellabeat and base my recommendations accordingly.

Activity Reminders and Rewards

Analysis showed that the users are split fairly evenly from sedentary to very active. We can derive that no matter what the activity level of the user is, they are all interested in tracking their activity. Approximately 79% of the users fall below the very active category and could potentially improve their daily activity level. With this, periodic alerts could be implemented to encourage users to increase their activity to hit a goal which could be either manually or automatically calculated. A reward system could be included to applaud users for reaching their goals daily, weekly and monthly.

Activity Tracking

The correlation between steps taken and sedentary minutes showed a moderate negative correlation with two distinct clusters at 7,500 and above 10,000 steps. This could be attributed to activities having different paces or number of steps taken per minute. For instance, running would count more far more steps per minute than slowly walking. There would be more sedentary minutes for someone who ran their 10000 steps than for someone who walked 10000 steps. This could also account the range of calories burned per steps taken shown on the Daily Steps and Calories Burned visualization. Most users with 10,000 steps have calories burned ranging from 1,500 to 3,500.

Bellabeat could implement a feature on the app that allows the user to distinguish when they engage in a fitness activity so it could more accurately calculate calories burned.

Combining Functionality

Bellabeat may have data on the timeline of the menstrual cycle and how it correlates with activity and sleep. If users to log their symptoms throughout the menstrual cycle along with their sleep and activity levels, we could possibly associate how levels of activity and sleep affect their overall symptoms. If greater activity and sleep levels are shown to improve symptoms during the menstrual cycle, notifications to increase these could be beneficial to users.

Marketing Strategy Recommendations

The analysis shows that people with all levels of activity utilize fitness tracking devices and most likely have an interest in their overall health and wellness. Bellabeat's women focused products allows them to have a more defined target audience and allows for unique features to be implemented. Bellabeat's features must also be competitive with other fitness tracking devices and apps as they do compete for some of the same market share. I recommend adding an enhanced activity tracking module on the app with both goal setting and notifications. This module should also interface with the menstrual cycle and symptom tracker. With these improvements/additions, Bellabeat could then create a marketing strategy to attract women who want tools at their fingertips to feel better everyday of the month.

Many thanks to the Kaggle community members for publishing their notebooks. These were instrumental in providing inspiration and information on how to complete my first R Markdown analysis. Comments welcome as I continue my Data Analysis journey.