Bellabeat Case Study

Coco Choo

2025-04-02

Bellabeat, founded in 2013 by Urška Sršen and Sando Mur, is a tech-driven wellness company specializing in health-focused smart products designed to empower women. Tracking activity, sleep, stress, and reproductive health has enabled women to make informed decisions about their well-being.

By 2016, Bellabeat had expanded globally, launching multiple products available through online retailers and their e-commerce platform. The company combines traditional advertising with a strong focus on digital marketing, leveraging platforms like Google Search and various social media to engage consumers year-round.

To explore growth opportunities, Bellabeat's marketing analytics team has been tasked with analyzing smart device usage data. This analysis aims to uncover user trends and the insights I discorver will provide strategic recommendations to enhance Bellabeat's marketing efforts.

This study follows the six phases of the data life cycle which are:

1. Ask
2. Prepare
3. Process
4. Analyze
5. Share
6. Act

Ask

Sršen has tasked you with analyzing smart device usage data to uncover trends in how consumers use non-Bellabeat smart devices. These insights will be applied to a selected Bellabeat product to inform strategic recommendations.

The primary stakeholders are:

- UrškaSršen: Bellabeat's Co-founder and Chief Creative Officer
- SandoMur: Mathematician and Bellabeat's Co-founder
- Bellabeat marketing analytics team: A team of data analysts

Prepare

The dataset used in this study can be found below:

FitBit Fitness Tracker Data

The datasets are accessible through Kaggle, licensed under the CCO Public Domain. The datasets contain personal fitness tracker from thirty fitbit users; for instance, daily activity, steps, minute-level output for physical activity, heart rate, and sleep monitoring.

The datasets are in CSV format and are stored in long format with unique ID having multiple rows of data. For this study, I focused on activity dataset to discover trends in smart device usage.

Process

Below are the datasets I will be using for this analysis:

- dailyactivity_merged (4.12.16-5.12.16)

I performed data cleaning using Excel and R. The following steps were taken for each dataset:

1. I sorted and filtered the data by ActivityDate
2. I checked for NULLs
3. I transformed the data type of ActivityDate column to Date type instead of Character type

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.2
```

```
## Warning: package 'ggplot2' was built under R version 4.4.1
```

```
## Warning: package 'tibble' was built under R version 4.4.1
```

```
## Warning: package 'tidyr' was built under R version 4.4.1
```

```
## Warning: package 'readr' was built under R version 4.4.2
```

```
## Warning: package 'purrr' was built under R version 4.4.1
```

```
## Warning: package 'dplyr' was built under R version 4.4.1
```

```
## Warning: package 'forcats' was built under R version 4.4.2
```

```
## Warning: package 'lubridate' was built under R version 4.4.1
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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
```

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.4.1
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
# Import data
setwd("C:\\Coco\\TTU\\Google Data Analytics Certification\\Bellabeat\\Fitabit_data_AM\\")

# Dataset I will be using for this analysis
activity <- read.csv("dailyActivity_merged.csv")
```

```r
# Check for null values
sum(is.na(activity))
```

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

```r
# Check the data types
str(activity)
```

```
## 'data.frame':    940 obs. of  15 variables:
##  $ Id                       : num  1.50e+09 1.62e+09 1.64e+09 1.84e+09 1.93e+09 ...
##  $ ActivityDate             : chr  "4/12/2016" "4/12/2016" "4/12/2016" "4/12/2016" ...
##  $ TotalSteps               : int  13162 8163 10694 6697 678 11875 4414 10725 10113 8796 ...
##  $ TotalDistance            : num  8.5 5.31 7.77 4.43 0.47 ...
##  $ TrackerDistance          : num  8.5 5.31 7.77 4.43 0.47 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance       : num  1.88 0 0.14 0 0 ...
##  $ ModeratelyActiveDistance: num  0.55 0 2.3 0 0 ...
##  $ LightActiveDistance      : num  6.06 5.31 5.33 4.43 0.47 ...
##  $ SedentaryActiveDistance  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes        : int  25 0 2 0 0 42 3 13 28 2 ...
##  $ FairlyActiveMinutes      : int  13 0 51 0 0 14 8 9 13 21 ...
##  $ LightlyActiveMinutes     : int  328 146 256 339 55 227 181 306 320 356 ...
##  $ SedentaryMinutes         : int  728 1294 1131 1101 734 1157 706 1112 964 1061 ...
##  $ Calories                 : int  1985 1432 3199 2030 2220 2390 1459 2124 2344 1982 ...
```

```r
# Transform the ActivityDate column into Date type
activity$ActivityDate <- as.Date(activity$ActivityDate, format="%m/%d/%Y")
```

```r
# Preview the datasets
head(activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366   2016-04-12      13162          8.50            8.50
## 2 1624580081   2016-04-12       8163          5.31            5.31
## 3 1644430081   2016-04-12      10694          7.77            7.77
## 4 1844505072   2016-04-12       6697          4.43            4.43
## 5 1927972279   2016-04-12        678          0.47            0.47
## 6 2022484408   2016-04-12      11875          8.34            8.34
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0              1.88                     0.55
## 2                        0              0.00                     0.00
## 3                        0              0.14                     2.30
## 4                        0              0.00                     0.00
## 5                        0              0.00                     0.00
## 6                        0              3.31                     0.77
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                5.31                       0                 0
## 3                5.33                       0                 2
## 4                4.43                       0                 0
```

```
## 5                0.47                        0                 0
## 6                4.26                        0                42
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  13                  328              728     1985
## 2                   0                  146             1294     1432
## 3                  51                  256             1131     3199
## 4                   0                  339             1101     2030
## 5                   0                   55              734     2220
## 6                  14                  227             1157     2390
```

```
colnames(activity)
```

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

```
# Unique participants in each dataset
n_distinct(activity$Id)
```

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

- Based on the result above, I observe that there are 33 unique participants in physical activity.

```
# Check number of observations in each dataset
nrow(activity)
```

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

- After conducting data cleaning and formatting on the dataset, I printed out some summary statistics with information I will focus on in this study for descriptive analysis purposes.

```
# Summary statistics
activity %>%
select(Calories,
       TotalSteps,
TotalDistance,
SedentaryMinutes) %>%
summary()
```

```
##     Calories      TotalSteps    TotalDistance    SedentaryMinutes
##  Min.   :   0   Min.   :    0   Min.   : 0.000   Min.   :   0.0
##  1st Qu.:1828   1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 729.8
##  Median :2134   Median : 7406   Median : 5.245   Median :1057.5
##  Mean   :2304   Mean   : 7638   Mean   : 5.490   Mean   : 991.2
##  3rd Qu.:2793   3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:1229.5
##  Max.   :4900   Max.   :36019   Max.   :28.030   Max.   :1440.0
```

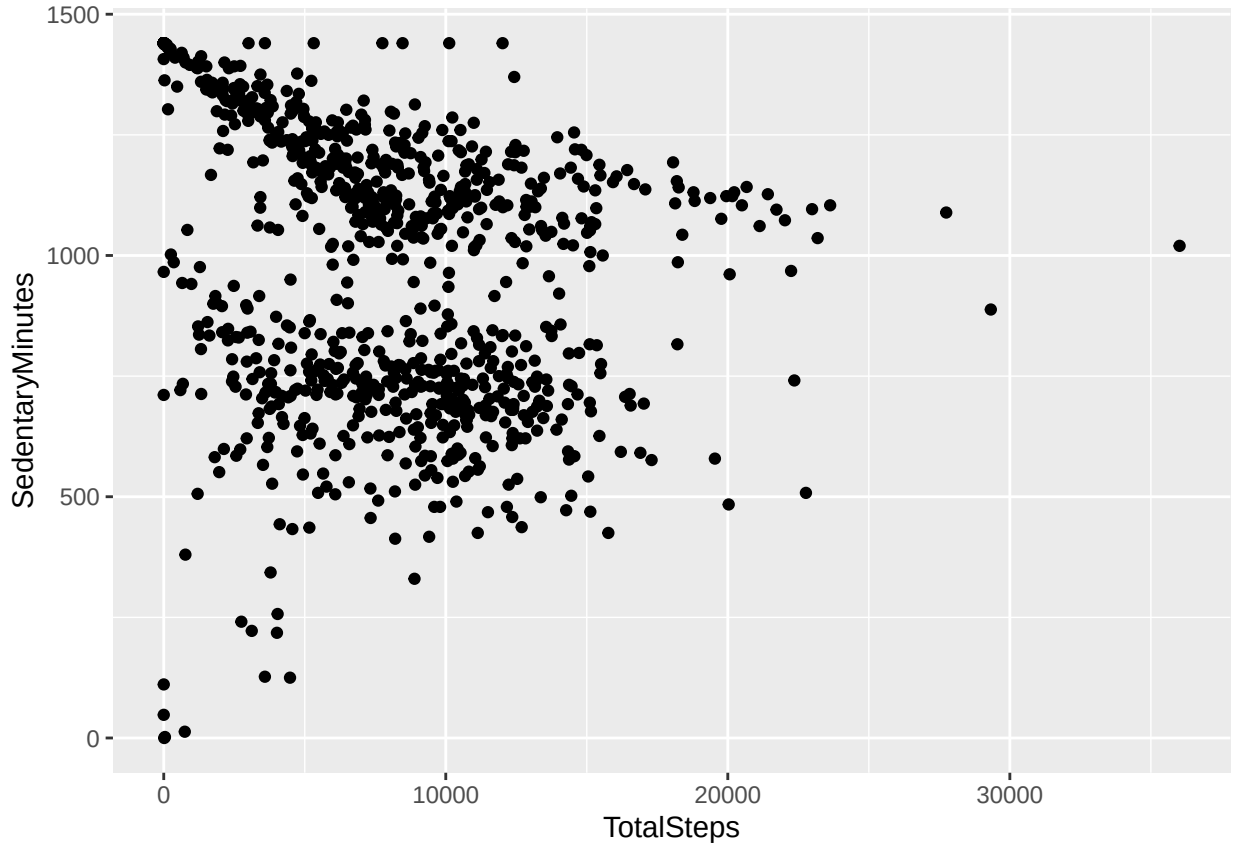
```
# Summary statistics
activity %>%
select(VeryActiveMinutes,
FairlyActiveMinutes,
LightlyActiveMinutes) %>%
summary()
```

```
##  VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :  0.0
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:127.0
##  Median :  4.00    Median :  6.00      Median :199.0
##  Mean   : 21.16    Mean   : 13.56      Mean   :192.8
##  3rd Qu.: 32.00    3rd Qu.: 19.00      3rd Qu.:264.0
##  Max.   :210.00    Max.   :143.00      Max.   :518.0
```

Analyze and Share

Although summary statistics provide valuable insights about the data, it does not provide a clear picture of how each variable is related at one sight. Thus, I created some plots to visualize the findings, connecting the dots.

```
# Relationship between steps and sedentary minutes
ggplot(data=activity, aes(x=TotalSteps, y=SedentaryMinutes)) + geom_point()
```



Based on the scatterplot displaying the relationship between Total Steps and Sedentary Minutes, I can observe some outliers with total steps above 25,000 but the overall, there appears to be a weak negative association between Total Steps and Sedentary Minutes. As Total Steps increases, Sedentary Minutes tend to decrease slightly, though the relationship is not very strong. One thing I notice is that the data points are densely clustered at lower Total Steps, around 0 to 10,000 and higher Sedentary Minutes, around 500 to 15,000, suggesting that most of the participants are relatively sedentary and do not have high step counts.

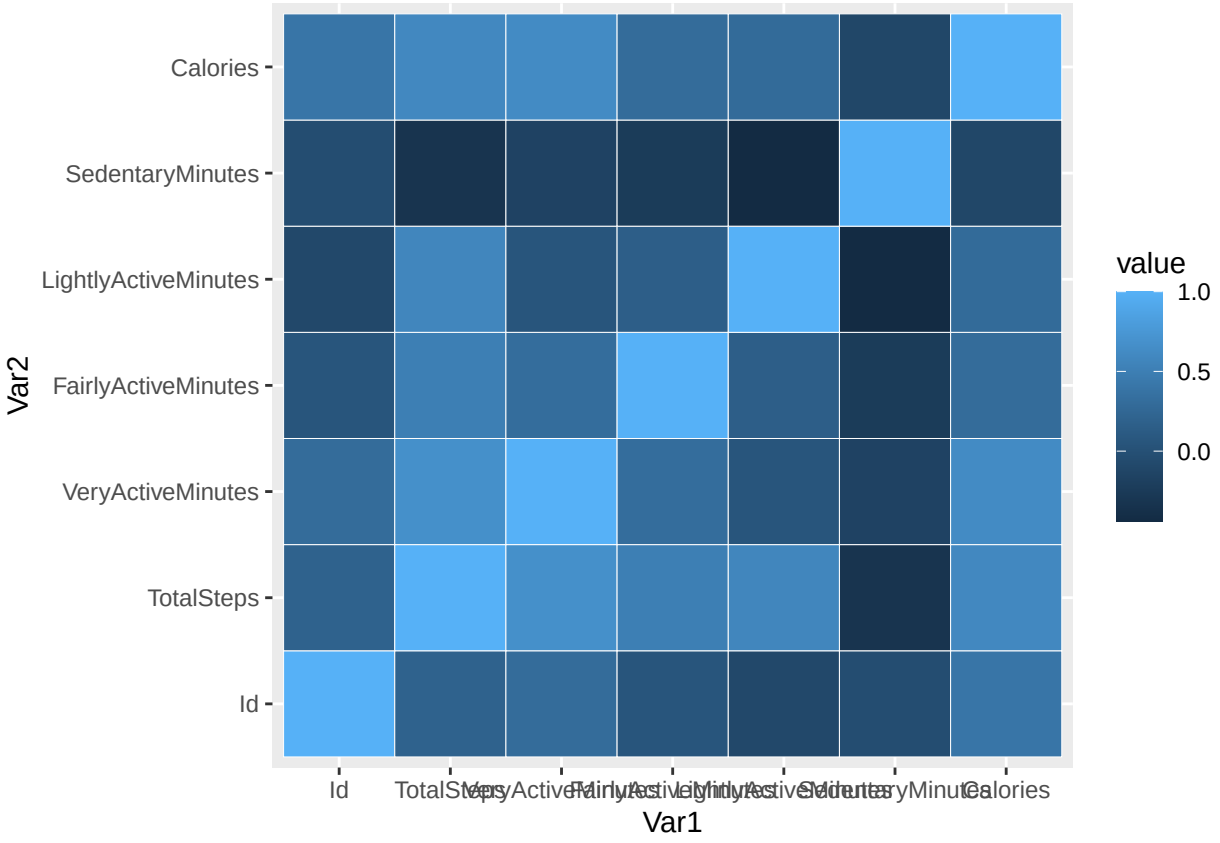
Then, I created a correlation matrix on numerical columns in activity dataset to investigate the correlation among the columns.

```r
# Focus on total steps, minutes and calories burnt
activity_num <- activity[c("Id", "ActivityDate", "TotalSteps", "VeryActiveMinutes", "FairlyActiveMinutes
# Create a correlation matrix of the numerical data
corr <- round(cor(activity_num[sapply(activity_num, is.numeric)]), 2)
head(corr)
```

```
##                           Id TotalSteps VeryActiveMinutes FairlyActiveMinutes
## Id                      1.00       0.19              0.30                0.05
## TotalSteps              0.19       1.00              0.67                0.50
## VeryActiveMinutes       0.30       0.67              1.00                0.31
## FairlyActiveMinutes     0.05       0.50              0.31                1.00
## LightlyActiveMinutes   -0.10       0.57              0.05                0.15
## SedentaryMinutes       -0.04      -0.33             -0.16               -0.24
##                         LightlyActiveMinutes SedentaryMinutes Calories
## Id                                     -0.10            -0.04     0.40
## TotalSteps                              0.57            -0.33     0.59
## VeryActiveMinutes                       0.05            -0.16     0.62
## FairlyActiveMinutes                     0.15            -0.24     0.30
## LightlyActiveMinutes                    1.00            -0.44     0.29
## SedentaryMinutes                       -0.44             1.00    -0.11
```

```r
# Melt the correlation matrix
melt_corr <- melt(corr)
```

```r
# Visualize correlation matrix on a heatmap
ggplot(data = melt_corr, aes(x=Var1, y=Var2, fill=value)) + geom_tile(color = "white")
```



Now that I looked into relationship between several variables, I then proceeded to analyze further with Very Active Minutes and Calories to segment participants into three categories: Highly active, moderately active, and sedentary.

The segmentation will help provide personalized Bellabeat marketing strategy based on the trends in the smart device usage. The process involves comparing the activity metrics I have selected to the mean values and then visualize them using a bar plot and a pie chart to get a clearer insight.

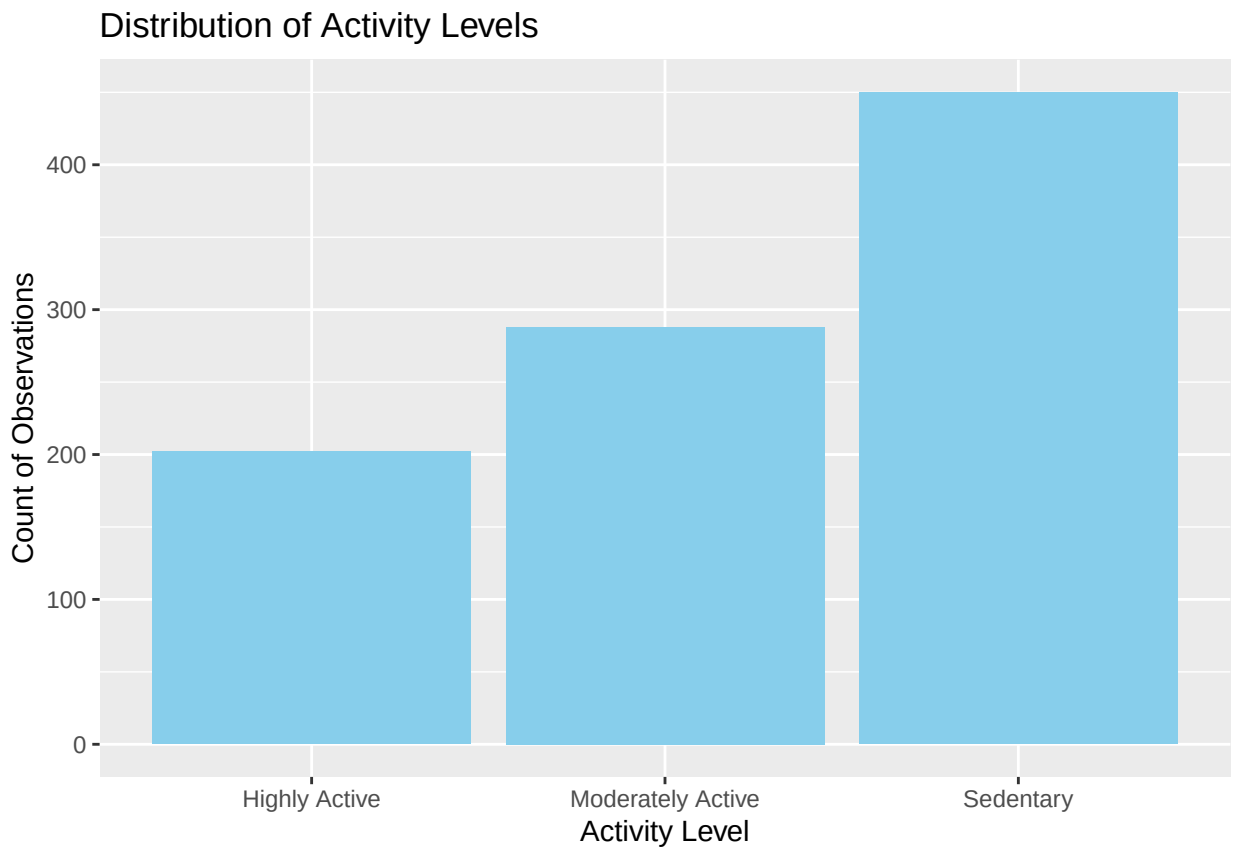
```
# Find the average of very active minutes and calories
mean_very_active_minutes <- mean(activity$VeryActiveMinutes)
mean_calories <- mean(activity$Calories)
```

```
# Segment users based on selected metrics relative to mean
activity$activity_level <-
  ifelse(
      activity$VeryActiveMinutes >= mean_very_active_minutes &
      activity$Calories >= mean_calories,
      "Highly Active",
  ifelse(
      activity$VeryActiveMinutes >= mean_very_active_minutes |
      activity$Calories >= mean_calories,
      "Moderately Active",
      "Sedentary"))

table(activity$activity_level)
```

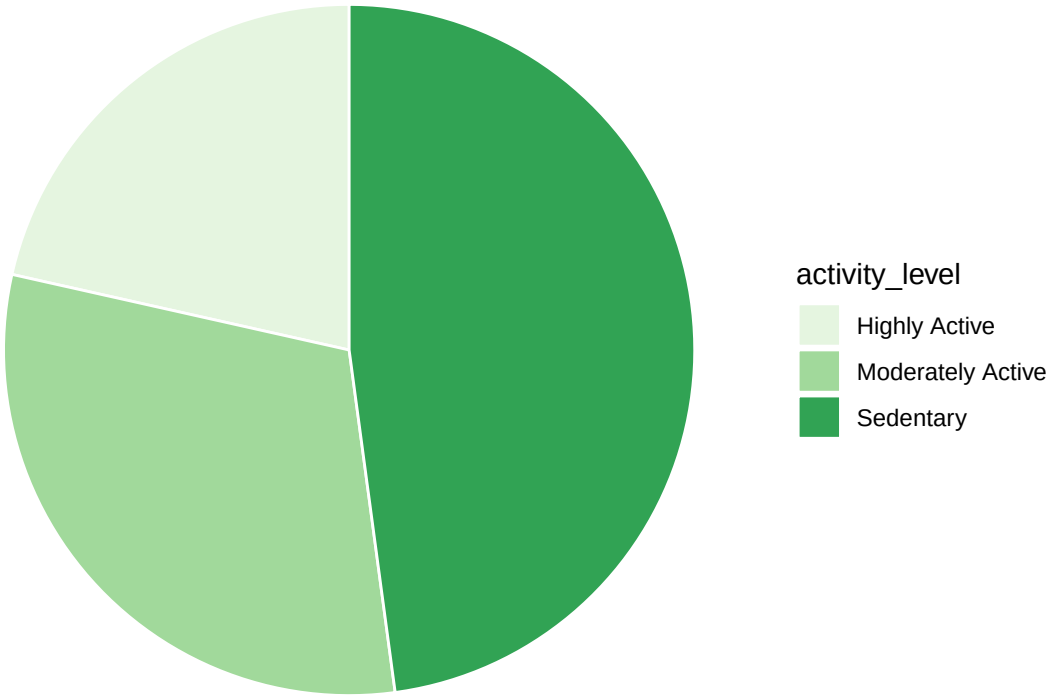
```
##
##     Highly Active Moderately Active         Sedentary
##               202               288               450
```

```
# Create a bar plot to visualize each user segment distribution
ggplot(activity, aes(x = activity_level)) +
  geom_bar(fill = "skyblue") +
  labs(title = "Distribution of Activity Levels", x = "Activity Level", y = "Count of Observations")
```




```
# Pie chart of the user segment distribution
ggplot(activity, aes(x="", fill=activity_level))+
  geom_bar(width = 1, color = "white")+
  coord_polar("y") + theme_void()+         scale_fill_brewer(palette = "Set4")
```

```
## Warning: Unknown palette: "Set4"
```



By segmenting participants based on the activity metrics, I have created categories with different levels of user engagement with Bellabeat product. The above plots suggest that most of the users tend to spend most of their time not doing much activities or only doing normal daily activities like walking.

I further analyzed when people are most active during the week. The process involves getting the day of week from the ActivityDate and create a bar plot to visualize the frequency by the day of week.

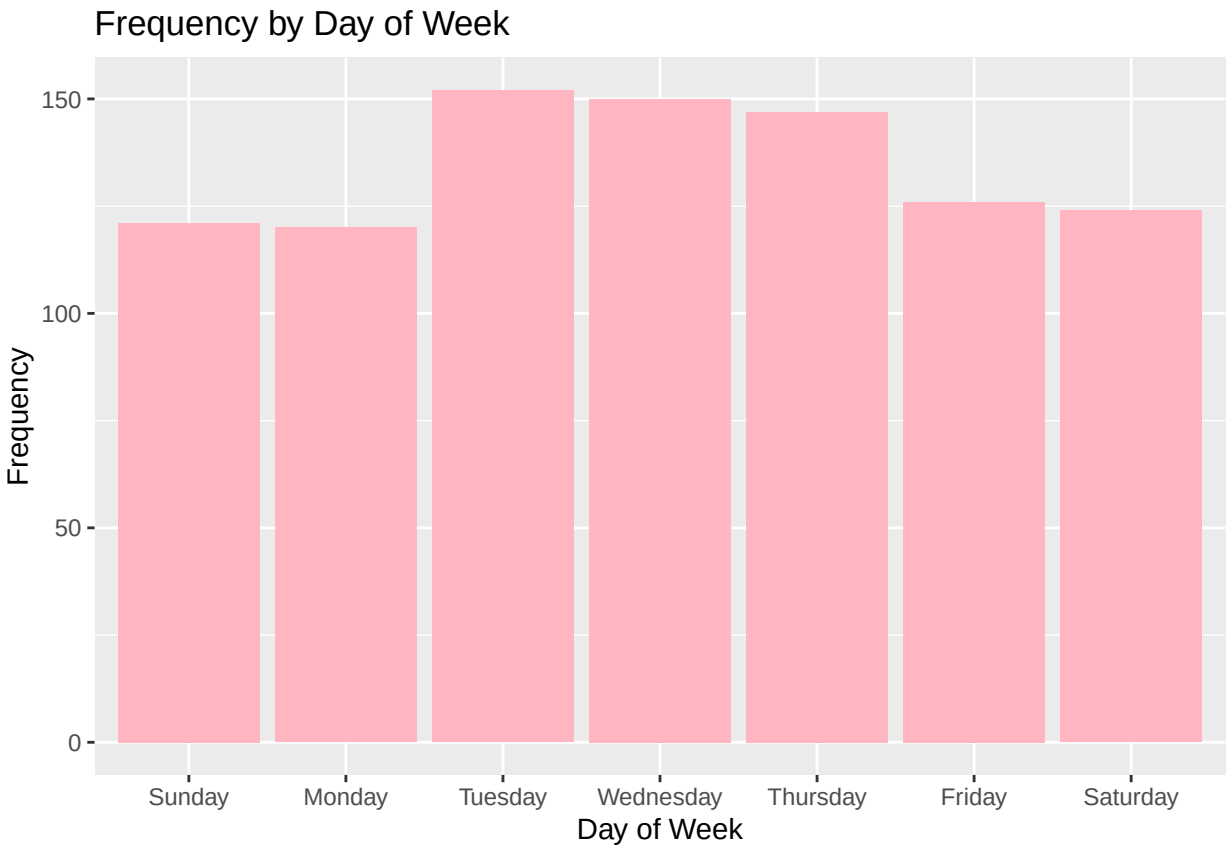
```
Sys.setlocale("LC_TIME", "C")
```

```
## [1] "C"
```

```
# Get day of week from the ActivityDate
activity$day_of_week <- weekdays(as.Date(activity$ActivityDate))

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

```
# Create a bar plot to visualize frequency by day of week
ggplot(activity, aes(x = day_of_week)) +
  geom_bar(fill = "lightpink") +
  labs(title = "Frequency by Day of Week", x = "Day of Week", y = "Frequency")
```



The above bar plot suggests that users are most active on Tuesdays, Wednesdays and Thursdays respectively compared to other days of the week.

Act

Based on the analysis and the insights I deduced from the analysis, here are my top three recommendations:

1. Since most of the Bellabeat product users are using the product on a daily basis with sedentary activity, only involving light activity like lying down, sitting with very low energy expenditure, the marketing team should focus on making users understand the importance of exercising for their health and general well-being.
2. The product can highlight features like tracking users' progress by setting activity goals per day and reminders to stay active. It can have a reminder feature that can help users complete their activity each day.
3. The product can also send motivational notifications to encourage users to stay active and improve their fitness habits specifically on week days other than Tuesdays, Wednesdays and Thursdays.