Bellabeat Case Study

Laia Ferrer-Argemi

2022-10-19

Contents

1. Business Task

The Objective of this case study is to identify the trends of smart fitness device users and apply the findings to the marketing strategy for Bellabeat products, which include:

- The Bellabeat app: provides users with health data related to their activity, sleep, stress, menstrual cycle, and mindfulness habits.
- Leaf: Bellabeat's classic wellness tracker can be worn as a bracelet, necklace, or clip.
- Time: wellness watch that combines the timeless look of a classic timepiece with smart technology to track user activity, sleep, and stress.
- Spring: water bottle that tracks daily water intake using smart technology
- Bellabeat membership: a subscription-based membership program for users that provides 24/7 access to fully personalized guidance on nutrition, activity, sleep, health and beauty, and mindfulness based on their lifestyle and goals.

The key stakeholders are Urška Sršen, co-founder and Chief Creative Officer of Bellabeat, Sando Mur, Bellabeat's co-founder and key member of the Bellabeat executive team, and Bellabeat marketing analytics team.

2. Data Source

The data I am going to analyze is publicly available in Kaggle. This data set was generated by respondents to a distributed survey via Amazon Mechanical Turk between 03.12.2016 and 05.12.2016. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring.

The data is structured in 28 tables including

- Daily activity totals: including calories, steps, total distance and distance per intensity, and minutes per heart rate intensity
- Hourly and minute by minute data on heart rate, steps, and heart rate intensity
- Daily and minute by minute sleep zones
- Daily weight and related metrics

Is this data ROCCC?

- Reliable: LOW - Only 33 respondents
- Original: LOW - Third-party provider(Amazon Mechanical Turk)
- Comprehensive: MED - Match most of Bellabeat product's parameters
- Current: LOW - it is from 2016 so it might not be relevant anymore
- Cited: MED - we know the origin but not how the survey was performed nor the demographics of the respondents

This data is considered bad quality and it would not be used to drive business decisions. Another key drawback of this data set is that it only contains data for 2 moths, both in Spring. Thus, we will be missing information on seasonality and long-term trends.

3. Loading Packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(ggplot2)
library(tidyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

4. Loading and Checking Data

I checked the data, first, using excel but some of the tables were too large to load. I decided to use R instead to do the analysis, so first I need to load the data. I am going to start first with the average daily and hourly data, and check the minute by minute data of any interesting findings later on.

```
activity_daily <- read.csv("Data/Fitabase Data 4.12.16-5.12.16/dailyActivity_merged.csv")
calories_hourly <- read.csv("Data/Fitabase Data 4.12.16-5.12.16/hourlyCalories_merged.csv")
intensity_hourly <- read.csv("Data/Fitabase Data 4.12.16-5.12.16/hourlyIntensities_merged.csv")
sleep_daily <- read.csv("Data/Fitabase Data 4.12.16-5.12.16/sleepDay_merged.csv")
weight_daily <- read.csv("Data/Fitabase Data 4.12.16-5.12.16/weightLogInfo_merged.csv")
```

Then, I need to check the data has been upload correctly and that the formatting is appropiate

```
head(activity_daily)
```

```
##           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
```

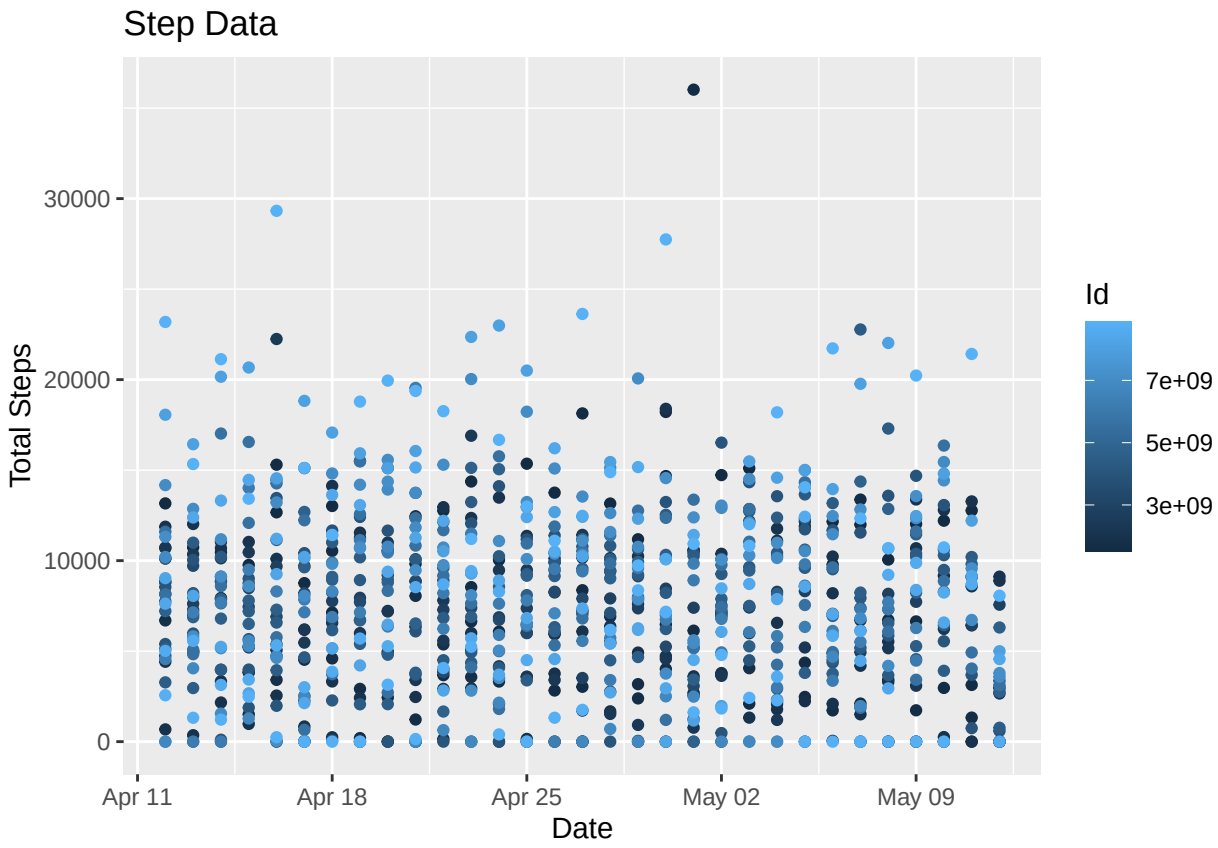
```
head(intensity_hourly)
```

```
##           Id          ActivityHour TotalIntensity AverageIntensity
```

```
## 1 1503960366 4/12/2016 12:00:00 AM             20         0.333333
## 2 1503960366  4/12/2016 1:00:00 AM              8         0.133333
## 3 1503960366  4/12/2016 2:00:00 AM              7         0.116667
## 4 1503960366  4/12/2016 3:00:00 AM              0         0.000000
## 5 1503960366  4/12/2016 4:00:00 AM              0         0.000000
## 6 1503960366  4/12/2016 5:00:00 AM              0         0.000000
```

All the data seems to be imported correctly, but I found that the date could not be used because it was in a non-standard unambiguous format. To fix this, I assigned the correct format and check that it is usable by doing a quite graph.

```
activity_daily$Date=as.POSIXct(activity_daily$ActivityDate, format="%m/%d/%Y", tz=Sys.timezone())

ggplot(data = activity_daily, aes(x = Date, y = TotalSteps, color = Id)) +
  geom_point() +
  labs(x = "Date",
    y = "Total Steps",
    title = "Step Data")
```



Then, I fixed the date and time format in the rest of the tables.

```
calories_hourly$Date<-as.POSIXct(calories_hourly$ActivityHour, format="%m/%d/%Y  %I:%M:%S %p", tz=Sys.t
intensity_hourly$Date<-as.POSIXct(intensity_hourly$ActivityHour, format="%m/%d/%Y  %I:%M:%S %p", tz=Sys
sleep_daily$Date<-as.POSIXct(sleep_daily$SleepDay, format="%m/%d/%Y   %I:%M:%S %p", tz=Sys.timezone())
weight_daily$Date<-as.POSIXct(weight_daily$Date, format="%m/%d/%Y   %I:%M:%S %p", tz=Sys.timezone())
intensity_hourly$Date<-as.POSIXct(intensity_hourly$ActivityHour, format="%m/%d/%Y   %I:%M:%S %p", tz=Sys
```

5. Exploring the Data

First, I want to check the minimum and maximum values for all the data, as well as the number of participants, to ensure there is no data entry errors.

```
n_distinct(activity_daily$Id)
```

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

```
n_distinct(calories_hourly$Id)
```

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

```
n_distinct(intensity_hourly$Id)
```

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

```
n_distinct(sleep_daily$Id)
```

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

```
n_distinct(weight_daily$Id)
```

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

I saw that most automatically gathered data is available for all 33 participants, but sleep data is only available for 24 participants, and weight data, which needs to be entered manually if the customer does not have a smart balance, is only available for 8 participants. This might be too small of a sample and it deems all the conclusions we might obtain pertaining to weight data meaningless.

```
# activity
activity_daily %>%
  select(TotalSteps,
         TotalDistance,
         SedentaryMinutes,
         Calories,
         VeryActiveMinutes,
         FairlyActiveMinutes,
         LightlyActiveMinutes) %>%
  summary()
```

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

```
# intensity
intensity_hourly %>%
  select(TotalIntensity,
```

```
        AverageIntensity) %>%
  summary()
```

```
##  TotalIntensity   AverageIntensity
##  Min.   :  0.00   Min.   :0.0000
##  1st Qu.:  0.00   1st Qu.:0.0000
##  Median :  3.00   Median :0.0500
##  Mean   : 12.04   Mean   :0.2006
##  3rd Qu.: 16.00   3rd Qu.:0.2667
##  Max.   :180.00   Max.   :3.0000
```

```
# calories
calories_hourly %>%
  select(Calories) %>%
  summary()
```

```
##     Calories
##  Min.   : 42.00
##  1st Qu.: 63.00
##  Median : 83.00
##  Mean   : 97.39
##  3rd Qu.:108.00
##  Max.   :948.00
```

```
# sleep
sleep_daily %>%
  select(TotalSleepRecords,
         TotalMinutesAsleep,
         TotalTimeInBed) %>%
  summary()
```

```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.000     Min.   : 58.0      Min.   : 61.0
##  1st Qu.:1.000     1st Qu.:361.0      1st Qu.:403.0
##  Median :1.000     Median :433.0      Median :463.0
##  Mean   :1.119     Mean   :419.5      Mean   :458.6
##  3rd Qu.:1.000     3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :3.000     Max.   :796.0      Max.   :961.0
```

```
# weight
weight_daily %>%
  select(WeightKg,
         BMI) %>%
  summary()
```

```
##     WeightKg           BMI
##  Min.   : 52.60   Min.   :21.45
##  1st Qu.: 61.40   1st Qu.:23.96
##  Median : 62.50   Median :24.39
##  Mean   : 72.04   Mean   :25.19
##  3rd Qu.: 85.05   3rd Qu.:25.56
##  Max.   :133.50   Max.   :47.54
```

All data ranges seem plausible, although there are some extreme sleep times up to 13 h that could be an erroneous entry but it could also indicate sickness. I also see some entries with 0 steps or 0 calories burned, which could indicate that the smart device was not being used that day.

6. Cleaning Data

To verify, let's plot the daily activity table ordered by ascending number of steps.

```
head(activity_daily[order(activity_daily$TotalSteps,decreasing=FALSE), ], 10)
```

```
##            Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 31  1503960366    5/12/2016          0             0               0
## 105 1844505072    4/24/2016          0             0               0
## 106 1844505072    4/25/2016          0             0               0
## 107 1844505072    4/26/2016          0             0               0
## 113 1844505072     5/2/2016          0             0               0
## 118 1844505072     5/7/2016          0             0               0
## 119 1844505072     5/8/2016          0             0               0
## 120 1844505072     5/9/2016          0             0               0
## 121 1844505072    5/10/2016          0             0               0
## 122 1844505072    5/11/2016          0             0               0
##     LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 31                         0                 0                        0
## 105                        0                 0                        0
## 106                        0                 0                        0
## 107                        0                 0                        0
## 113                        0                 0                        0
## 118                        0                 0                        0
## 119                        0                 0                        0
## 120                        0                 0                        0
## 121                        0                 0                        0
## 122                        0                 0                        0
##     LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 31                    0                       0                 0
## 105                   0                       0                 0
## 106                   0                       0                 0
## 107                   0                       0                 0
## 113                   0                       0                 0
## 118                   0                       0                 0
## 119                   0                       0                 0
## 120                   0                       0                 0
## 121                   0                       0                 0
## 122                   0                       0                 0
##     FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 31                    0                    0             1440        0
## 105                   0                    0             1440     1347
## 106                   0                    0             1440     1347
## 107                   0                    0             1440     1347
## 113                   0                    0             1440     1348
## 118                   0                    0             1440     1347
## 119                   0                    0             1440     1347
## 120                   0                    0             1440     1347
## 121                   0                    0             1440     1347
## 122                   0                    0             1440     1347
##           Date
## 31  2016-05-12
## 105 2016-04-24
## 106 2016-04-25
## 107 2016-04-26
```

```
## 113 2016-05-02
## 118 2016-05-07
## 119 2016-05-08
## 120 2016-05-09
## 121 2016-05-10
## 122 2016-05-11
```

It is clear that all entries are 0 when the number of steps is 0, except for the base calories, thus indicating that the device was not being used that day (since even going to the bathroom would count a couple steps). After verifying that all the entries with 0 steps have indeed 0 in all other activity data, I applied a filter to remove these entries so the results are not skewed towards false inactivity.

```
activity_daily_filtered <- filter(activity_daily,TotalSteps>0)
activity_daily_filtered %>%
  select(TotalSteps,
         TotalDistance,
         SedentaryMinutes,
         Calories,
         VeryActiveMinutes,
         FairlyActiveMinutes,
         LightlyActiveMinutes) %>%
  summary()
```

```
##    TotalSteps    TotalDistance   SedentaryMinutes    Calories
##  Min.   :    4   Min.   : 0.00   Min.   :   0.0   Min.   :  52
##  1st Qu.: 4923   1st Qu.: 3.37   1st Qu.: 721.5   1st Qu.:1856
##  Median : 8053   Median : 5.59   Median :1021.0   Median :2220
##  Mean   : 8319   Mean   : 5.98   Mean   : 955.8   Mean   :2361
##  3rd Qu.:11092   3rd Qu.: 7.90   3rd Qu.:1189.0   3rd Qu.:2832
##  Max.   :36019   Max.   :28.03   Max.   :1440.0   Max.   :4900
##  VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :  0.0
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:146.5
##  Median :  7.00    Median :  8.00      Median :208.0
##  Mean   : 23.02    Mean   : 14.78      Mean   :210.0
##  3rd Qu.: 35.00    3rd Qu.: 21.00      3rd Qu.:272.0
##  Max.   :210.00    Max.   :143.00      Max.   :518.0
```

Looks like there are still some erroneous entries with impossible calorie counts, so I deleted as well all entries with a calorie count below 1000 (since even a child burns that much energy a day).

```
activity_daily_filtered <- filter(activity_daily_filtered,Calories>1000)
activity_daily_filtered %>%
  select(TotalSteps,
         TotalDistance,
         SedentaryMinutes,
         Calories,
         VeryActiveMinutes,
         FairlyActiveMinutes,
         LightlyActiveMinutes) %>%
  summary()
```

```
##    TotalSteps    TotalDistance    SedentaryMinutes    Calories
##  Min.   :    4   Min.   : 0.000   Min.   : 125.0   Min.   :1002
##  1st Qu.: 4936   1st Qu.: 3.390   1st Qu.: 724.0   1st Qu.:1862
##  Median : 8062   Median : 5.605   Median :1024.0   Median :2222
```

```
##  Mean   : 8359   Mean   : 6.009   Mean   : 960.9   Mean   :2372
##  3rd Qu.:11101   3rd Qu.: 7.918   3rd Qu.:1189.8   3rd Qu.:2835
##  Max.   :36019   Max.   :28.030   Max.   :1440.0   Max.   :4900
##  VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :  0
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:148
##  Median :  7.00    Median :  8.00      Median :209
##  Mean   : 23.14    Mean   : 14.84      Mean   :211
##  3rd Qu.: 35.75    3rd Qu.: 21.00      3rd Qu.:272
##  Max.   :210.00    Max.   :143.00      Max.   :518
```

```
length(unique(activity_daily_filtered$Id))
```

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

Since we do not want to delete very sedentary days, I will stop the data cleaning of the daily activity data here. There is a significant increase on the average number of steps from 7638 to 8359 and other activity parameters compared to the unfiltered data.

7. Merging and Grouping Data

In order to have insights on the relationship or lack thereof between daily activity and sleep, I am going to merge the daily activity and sleep data with an inner join (as to only keep the filtered data that also has sleep data associated with it)

```
merge_daily <- merge(activity_daily_filtered, sleep_daily, by = c("Id","Date"), all.x = FALSE, all.y = F
```

The 33 users that reported daily activity can be separated into different groups depending on their level of activity. There are several criteria that could be used to create these groups. I will use the average intensity minutes per zone for all users (calculated in the previous section for the cleaned data) and compare it to the average minutes per user to classify them between 4 groups: sedentary, lightly active, moderately active, and very active. The decision tree, which was created to ensure all users had a type assigned to them, is as follows:

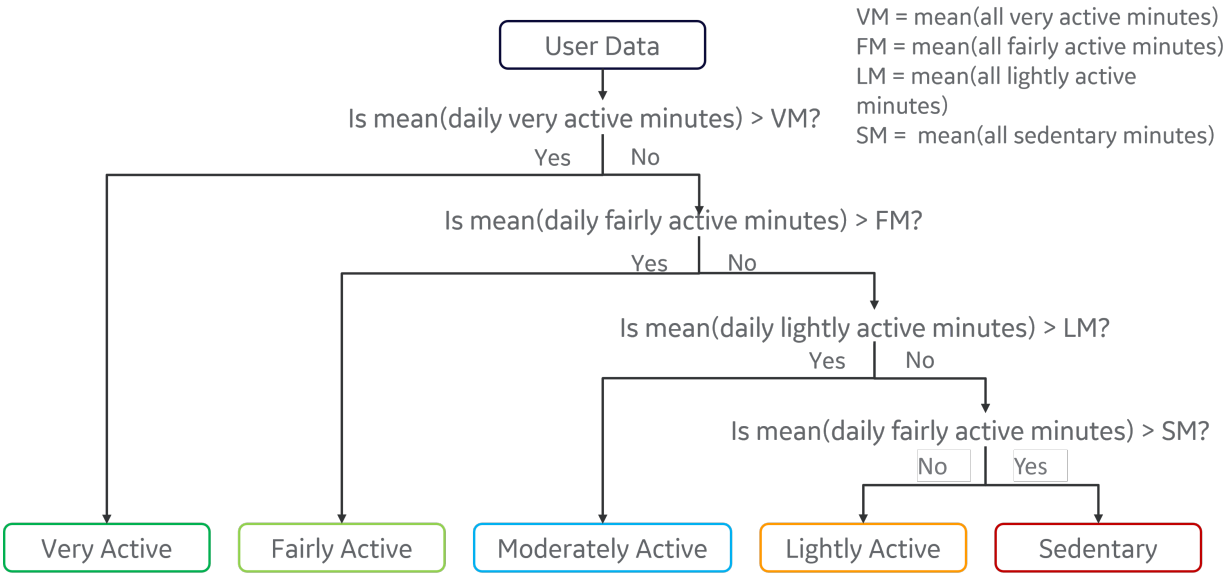


Figure 1: Decision Tree to group the users into activity levels

This decision tree puts more emphasis on workouts than moving around all day; for example, someone might sit for 22 h a day and do one intense workout a day and that will place them in the very active category. A more in depth analysis with a larger dataset could compare the difference between these two types of users. I will apply this criteria per user and also for individual days to see if the overall activity level or the activity of a particular day affect the sleep more. I will also add another division to see if walking, which is the only activity that is reported by the number of steps, has a particular impact on sleep without taking into account other activity. Recent meta-analyses (Paluch, The Lancet, 2022) has found that walking incrementally decreases the mortality risk until it levels of at around 8000 steps/day; therefore, I will use the 8000 threshold to divide the users between walkers and non-walkers.

```
#creating a new column to assign an activity level to each daily and hourly report
activity_daily_filtered <- activity_daily_filtered %>%
  mutate(activity_level = case_when(
    VeryActiveMinutes > mean(VeryActiveMinutes) ~ "Very Active",
    VeryActiveMinutes < mean(VeryActiveMinutes) & FairlyActiveMinutes > mean(FairlyActiveMinutes) ~ "Fa
    VeryActiveMinutes < mean(VeryActiveMinutes) & FairlyActiveMinutes < mean(FairlyActiveMinutes) & Ligh
    VeryActiveMinutes < mean(VeryActiveMinutes) & FairlyActiveMinutes < mean(FairlyActiveMinutes) & Ligh
    VeryActiveMinutes < mean(VeryActiveMinutes) & FairlyActiveMinutes < mean(FairlyActiveMinutes) & Ligh
activity_daily_filtered$activity_level <- factor(activity_daily_filtered$activity_level,levels = c("Sede

#doing the same assignment for the data that includes both activity and sleep
merge_daily <- merge_daily %>%
  mutate(activity_level = case_when(
    VeryActiveMinutes > mean(VeryActiveMinutes) ~ "Very Active",
    VeryActiveMinutes < mean(VeryActiveMinutes) & FairlyActiveMinutes > mean(FairlyActiveMinutes) ~ "Fa
    VeryActiveMinutes < mean(VeryActiveMinutes) & FairlyActiveMinutes < mean(FairlyActiveMinutes) & Ligh
    VeryActiveMinutes < mean(VeryActiveMinutes) & FairlyActiveMinutes < mean(FairlyActiveMinutes) & Ligh
    VeryActiveMinutes < mean(VeryActiveMinutes) & FairlyActiveMinutes < mean(FairlyActiveMinutes) & Ligh
merge_daily$activity_level <- factor(merge_daily$activity_level,levels = c("Sedentary", "Lightly Active

#creating a new table with only the average the user type by activity level and the average calories th
data_by_user <- activity_daily_filtered %>%
  select(Id,
         TotalSteps,
         SedentaryMinutes,
         VeryActiveMinutes,
         FairlyActiveMinutes,
         LightlyActiveMinutes,
         Calories) %>%
  group_by(Id) %>%
  summarise_at(vars(TotalSteps,
                    SedentaryMinutes,
                    VeryActiveMinutes,
                    FairlyActiveMinutes,
                    LightlyActiveMinutes,
                    Calories),
               list(mean))
data_by_user <- data_by_user %>%
  mutate(user_type = case_when(
    VeryActiveMinutes > mean(VeryActiveMinutes) ~ "Very Active",
    VeryActiveMinutes < mean(VeryActiveMinutes) & FairlyActiveMinutes > mean(FairlyActiveMinutes) ~ "Fa
    VeryActiveMinutes < mean(VeryActiveMinutes) & FairlyActiveMinutes < mean(FairlyActiveMinutes) & Ligh
    VeryActiveMinutes < mean(VeryActiveMinutes) & FairlyActiveMinutes < mean(FairlyActiveMinutes) & Ligh
    VeryActiveMinutes < mean(VeryActiveMinutes) & FairlyActiveMinutes < mean(FairlyActiveMinutes) & Ligh
```

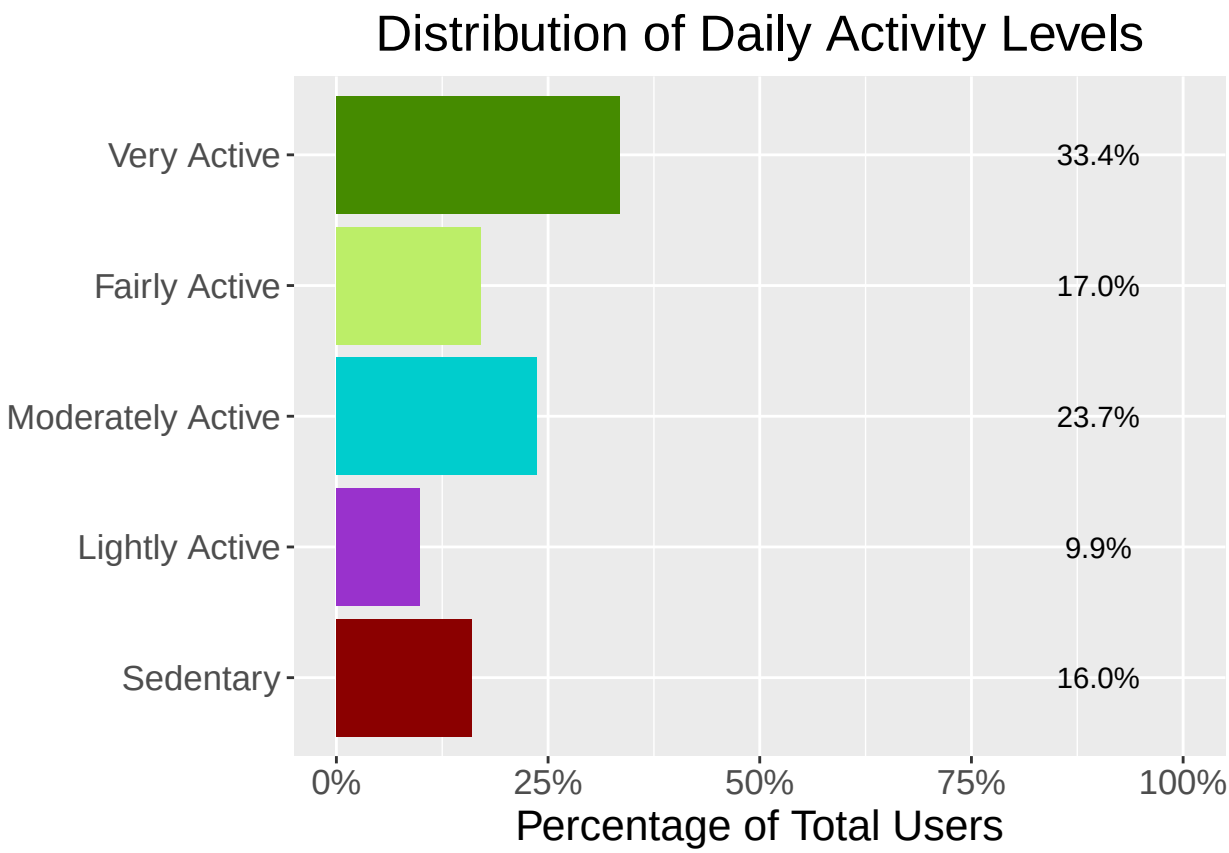
```
data_by_user$user_type <- factor(data_by_user$user_type,levels = c("Sedentary", "Lightly Active", "Mode

#creating a new column to assign if a user is a walker or not
data_by_user <- data_by_user %>%
  mutate(walker = ifelse(
    TotalSteps > 8000, TRUE, FALSE))

#adding the user type to the daily and hourly activity data frames
activity_daily_filtered <- merge(activity_daily_filtered,
                                 data_by_user[, c("Id", "user_type","walker")],
                                 by = "Id", all.x = TRUE)
merge_daily <- merge(merge_daily,
                     data_by_user[, c("Id", "user_type","walker")],
                     by = "Id", all.x = TRUE)
intensity_hourly <- merge(intensity_hourly,
                                 data_by_user[, c("Id", "user_type","walker")],
                                 by = "Id", all.x = TRUE)
```

8. Analysis

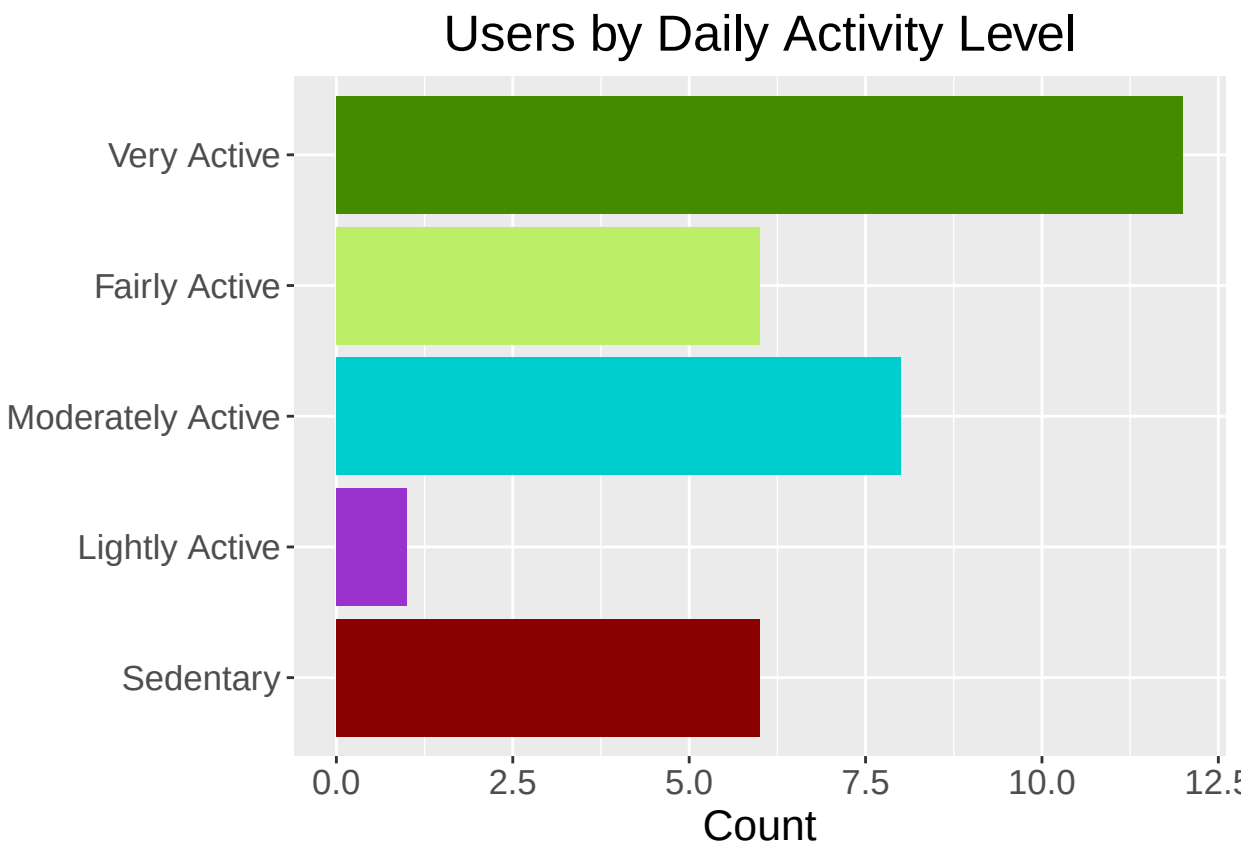
8.1 Activty Levels of the Users

I will start by visualizing how the daily and average user types are distributed

```
#looking at how active are daily entries
activity_daily_filtered %>%
  ggplot(aes(x=activity_level, fill=activity_level)) +
    geom_bar(aes(y = (..count..)/sum(..count..)))  +
    geom_text(stat="count",
              aes(label=scales::percent(prop.table(stat(count)))),
              position = position_fill(vjust = 0.9, reverse = FALSE)) +
    scale_y_continuous(labels = scales::percent)+
    labs(title="Distribution of Daily Activity Levels", x=NULL, y="Percentage of Total Users")+
    theme(legend.position="none", text = element_text(size = 16),plot.title =  element_text(hjust = 0.5)
    coord_flip() +
    scale_fill_manual(values = c("Sedentary"="darkred",
                                 "Lightly Active"="darkorchid",
                                 "Moderately Active"="cyan3",
                                 "Fairly Active" ="darkolivegreen2",
                                 "Very Active" ="chartreuse4"))
```




```
#looking at how many users we have per user type
ggplot(data_by_user, aes(x = user_type, fill = user_type)) +
    geom_bar() +
    coord_flip() +
    labs(title="Users by Daily Activity Level", x=NULL, y="Count") +
    theme(legend.position="none", text = element_text(size = 16),plot.title = element_text(hjust = 0.5))
  scale_fill_manual(values = c("Sedentary"="darkred",
                               "Lightly Active"="darkorchid",
                               "Moderately Active"="cyan3",
                               "Fairly Active" ="darkolivegreen2",
                               "Very Active" ="chartreuse4"))
```

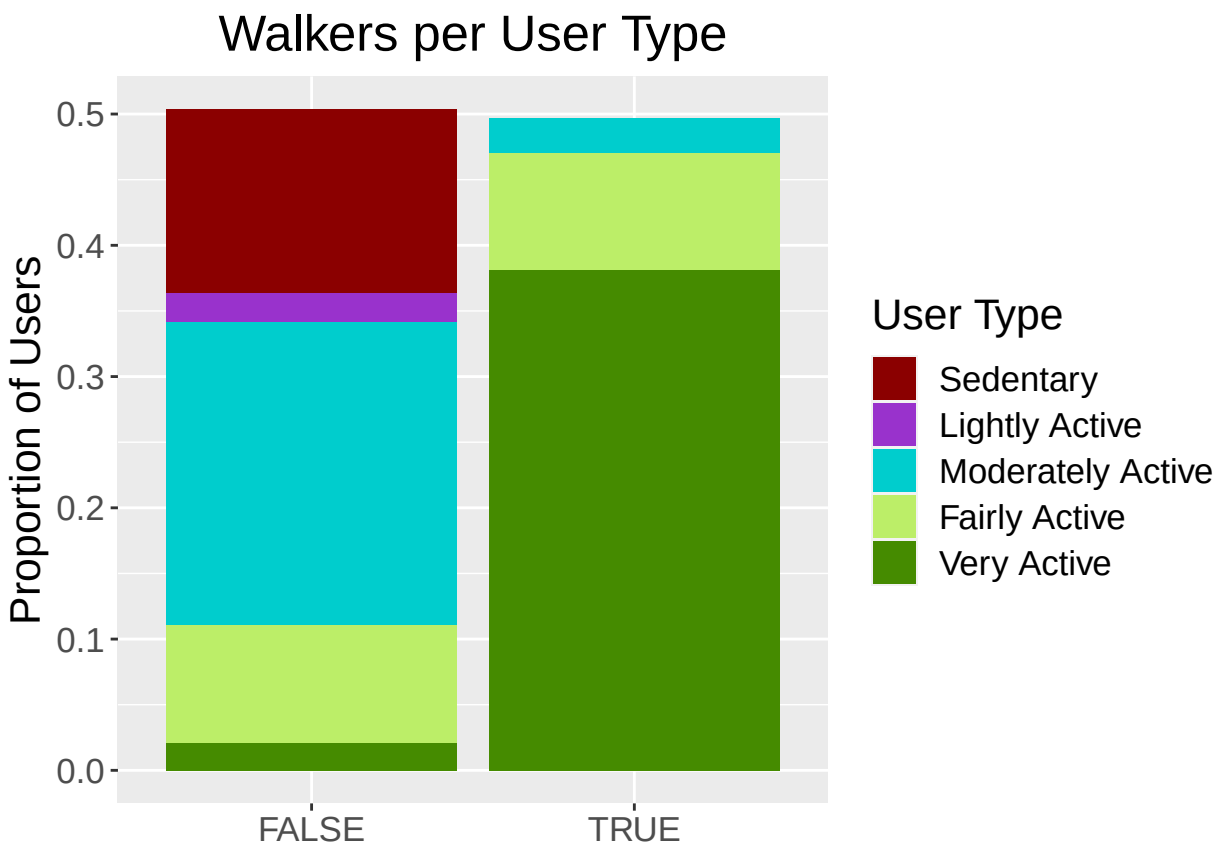


The results indicate that

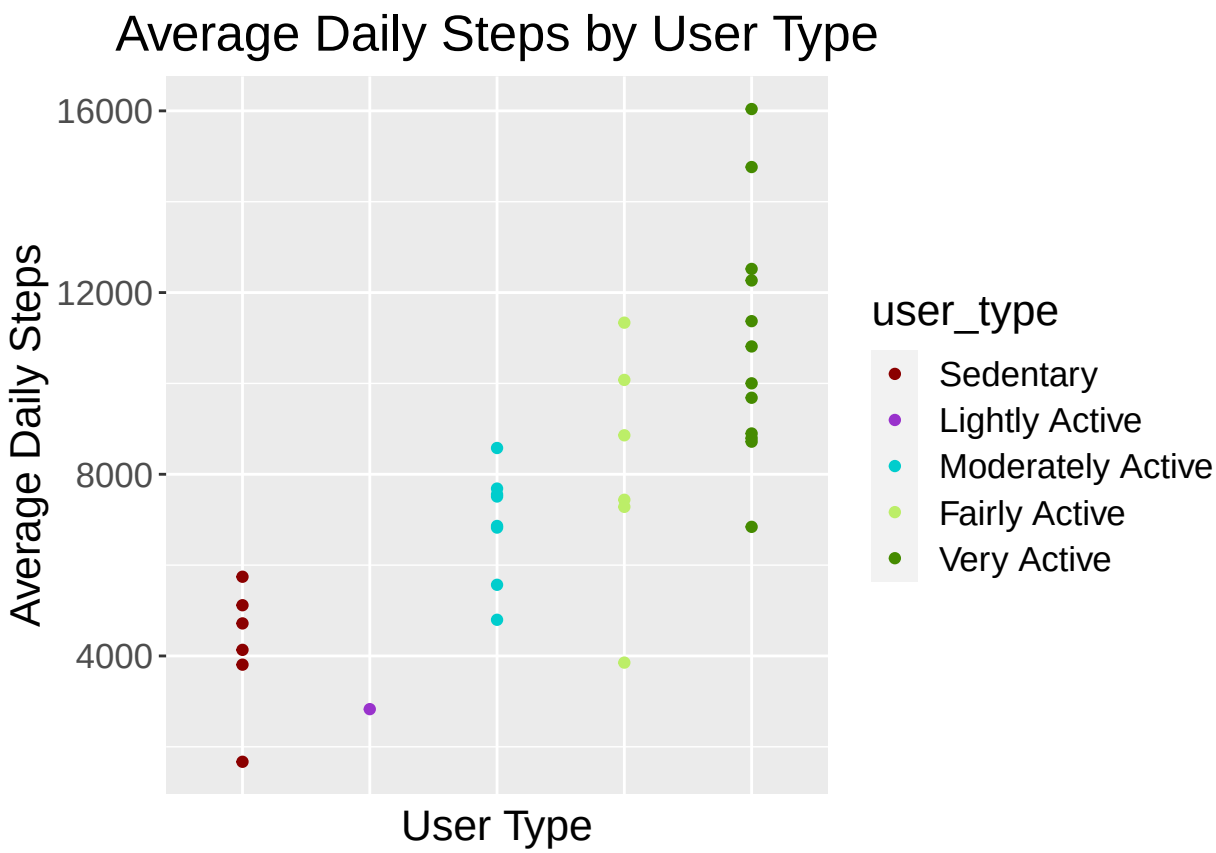
- in a third of the daily entries the users were very active (high heart rate for more than 23 minutes),
- in 74% of the days, the users performed at least a moderate activity (more than 211 minutes of light or more intense activity). - only 16% of the daily entries were sedentary, meaning the user did not perform any kind of activity for more than a few minutes.
- Overall, most users in this study are at least moderately active, with only 6 being primarily sedentary and 1 being lightly active.

Let's now check if the user average activity level correlates with them also walking more

```
#looking if user type correlates with walker category
ggplot(activity_daily_filtered, aes(walker, fill=user_type)) +
    geom_bar(aes(y = (..count..)/sum(..count..))) +
    labs(title="Walkers per User Type", x=NULL, y="Proportion of Users", fill="User Type") +
    theme(text = element_text(size = 16),plot.title = element_text(hjust = 0.5)) +
  scale_fill_manual(values = c("Sedentary"="darkred",
                               "Lightly Active"="darkorchid",
                               "Moderately Active"="cyan3",
                               "Fairly Active" ="darkolivegreen2",
                               "Very Active" ="chartreuse4"))
```




```
#plotting the average number of daily steps
ggplot(data_by_user, aes(user_type,TotalSteps)) +
  geom_point(aes(colour = user_type))  +
  labs(title="Average Daily Steps by User Type", x="User Type", y="Average Daily Steps")+
    theme(text = element_text(size = 16),plot.title = element_text(hjust = 0.5),axis.text.x=element_bla
  scale_colour_manual(values = c("Sedentary"="darkred",
                                 "Lightly Active"="darkorchid",
                                 "Moderately Active"="cyan3",
                                 "Fairly Active" ="darkolivegreen2",
                                 "Very Active" ="chartreuse4"))
```

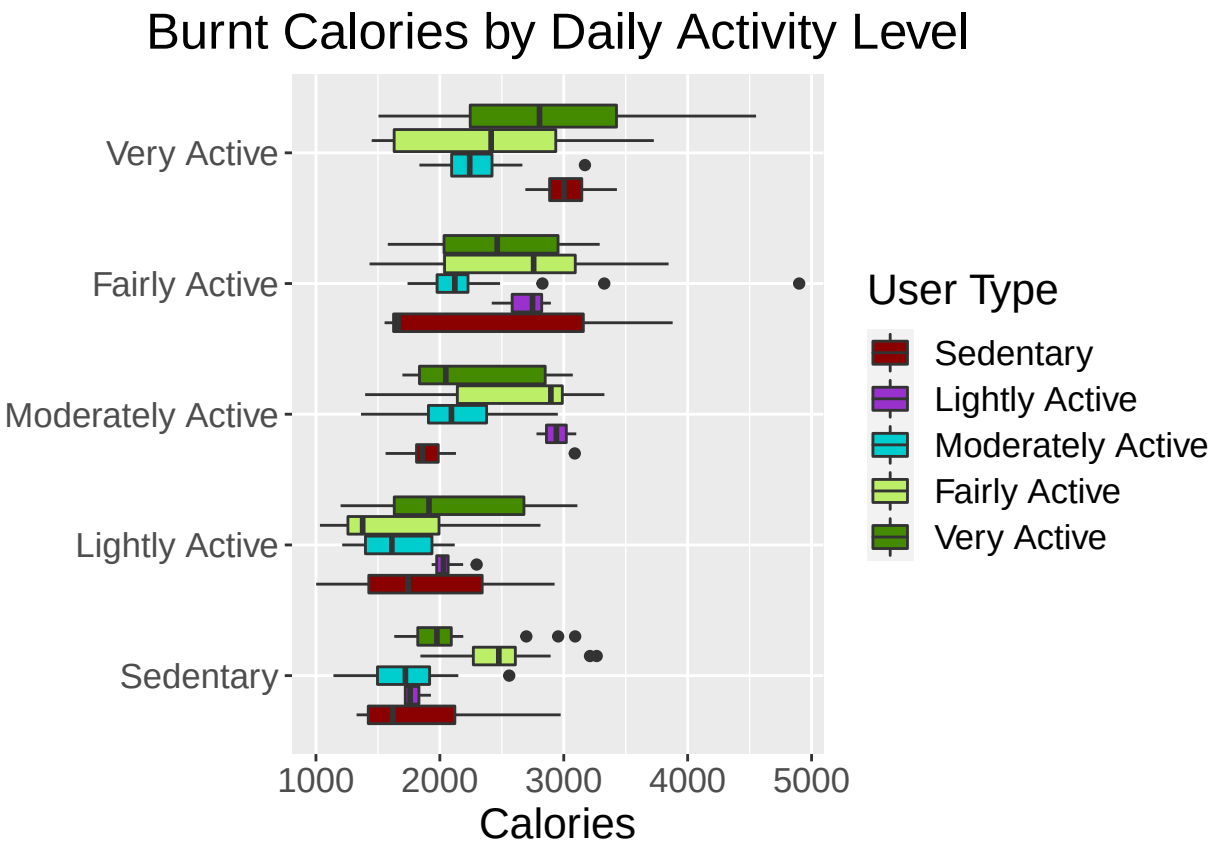


All walkers, users with an average step count of 8000 or more, are also at least moderately active and higher activity users tend to have higher average step counts. The activity level translates into hihger calory counts on average, but I wanted to look at if average activity levels influence burned calories in different activity days.

8.2 Relationship Between Activity Level and Burnt Calories

So let's dive now into the relationship between activity levels and burnt daily calories

```
ggplot(activity_daily_filtered, aes(activity_level, Calories, fill=user_type)) +
    geom_boxplot() +
    labs(title="Burnt Calories by Daily Activity Level", x=NULL, fill="User Type") +
    theme(text = element_text(size = 16),plot.title = element_text(hjust = 0.5)) +
  coord_flip()+
  scale_fill_manual(values = c("Sedentary"="darkred",
                               "Lightly Active"="darkorchid",
                               "Moderately Active"="cyan3",
                               "Fairly Active" ="darkolivegreen2",
                               "Very Active" ="chartreuse4"))
```

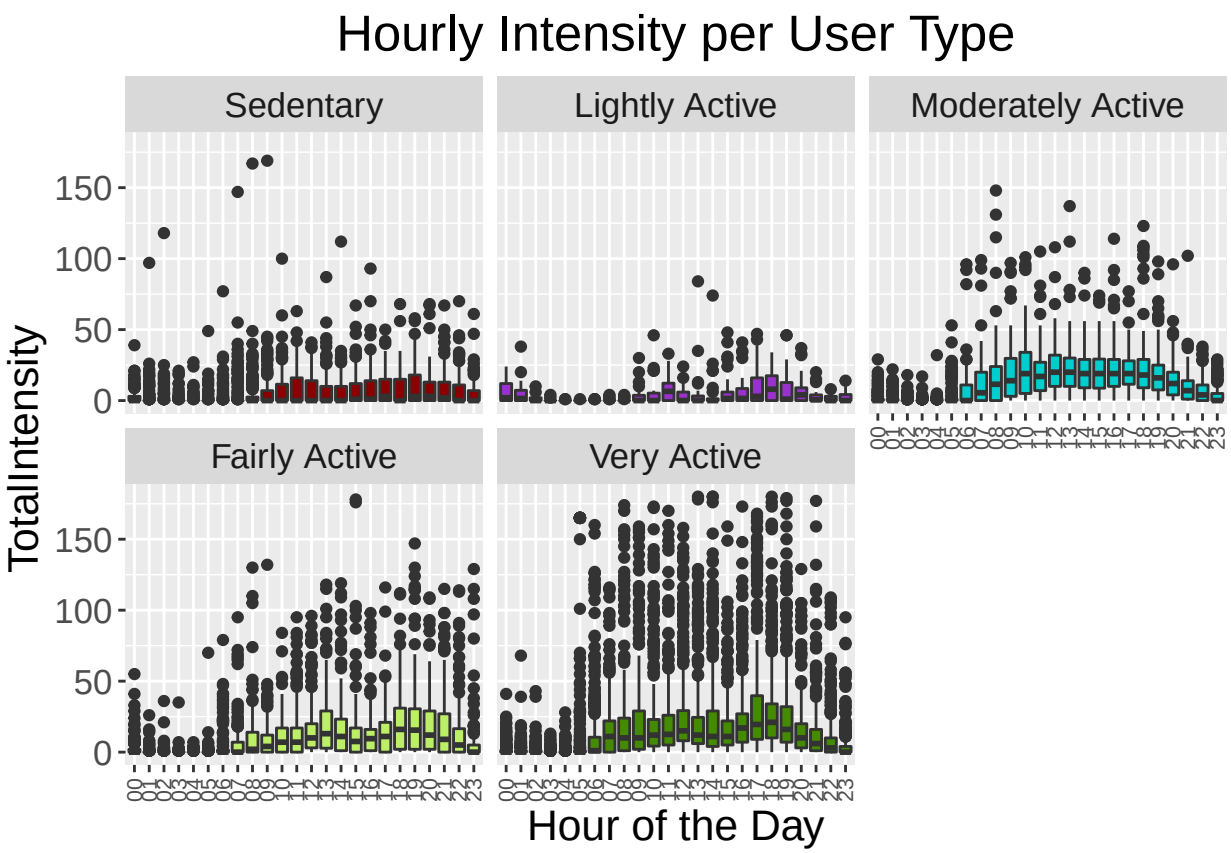


Very and fairly active users also tend to burn a higher amount of calories during sedentary days, but sedentary users can burn more calories during very active days. More insight could be drawn from weight and body fat percentage if they were available from all users, but only 8 users reported any kind of weight data and all but 2 only provided 1 to 4 entries.

8.3 Hourly Activity

Different types of users might behave differently during the day, which could be used to personalize daily reminders.

```
intensity_hourly$hour <- format(as.POSIXct(intensity_hourly$Date), format = "%H")
ggplot(intensity_hourly, aes(hour, TotalIntensity,fill=user_type)) +
  geom_boxplot() +
  facet_wrap(~user_type) +
  labs(title="Hourly Intensity per User Type", x="Hour of the Day") +
  theme(text = element_text(size = 16),plot.title = element_text(hjust = 0.5),axis.text.x = element_tex
  scale_fill_manual(values = c("Sedentary"="darkred",
                          "Lightly Active"="darkorchid",
                          "Moderately Active"="cyan3",
                          "Fairly Active" ="darkolivegreen2",
                          "Very Active" ="chartreuse4"))
```

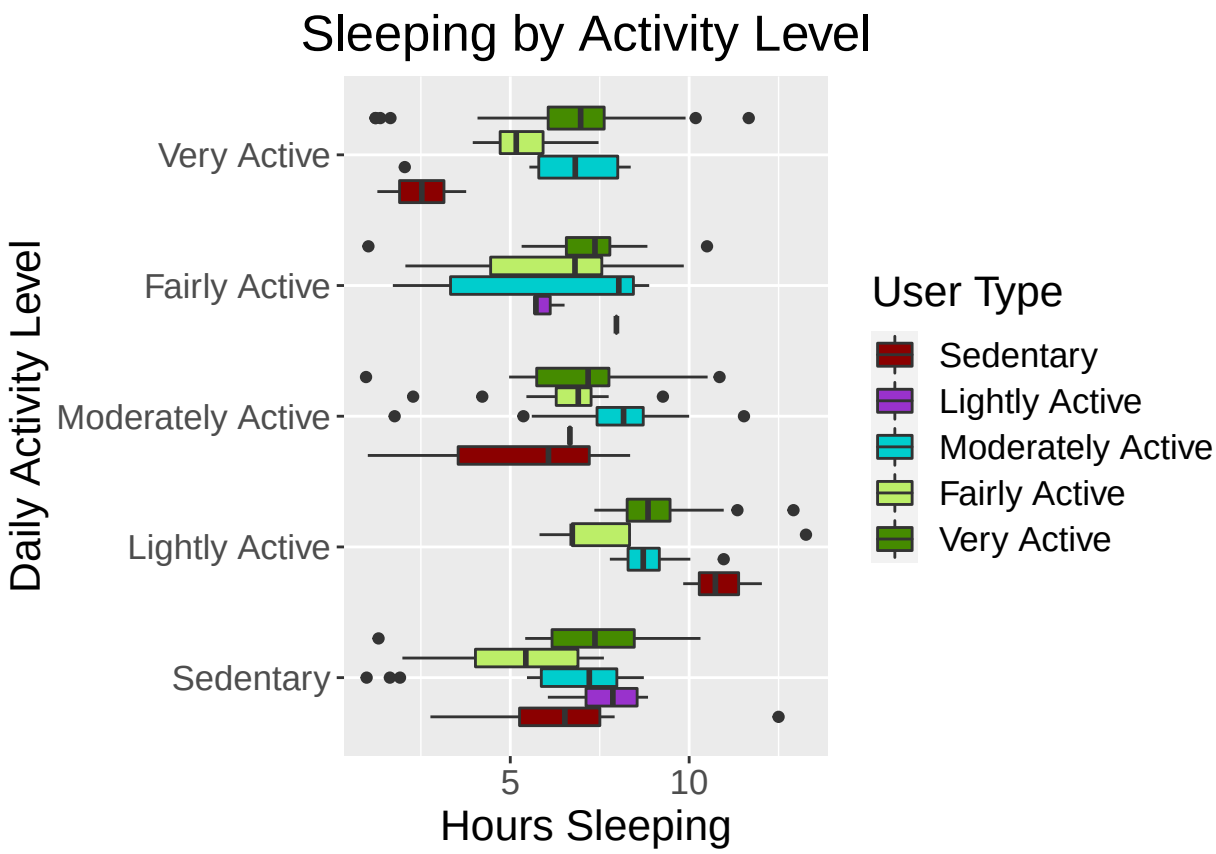


Since there is only 1 lightly active user in this data set, that plot is indicative of a personal average activity. This user does not usually have significant activity until 9 to 10 am, and their most active hours are after work from 5 yo 7 pm. Late start of daily activity can be also observed for the sedentary users, who also present significant activity past midnight. The most active users start the day at 6 am and mantain a high activity level during the day.

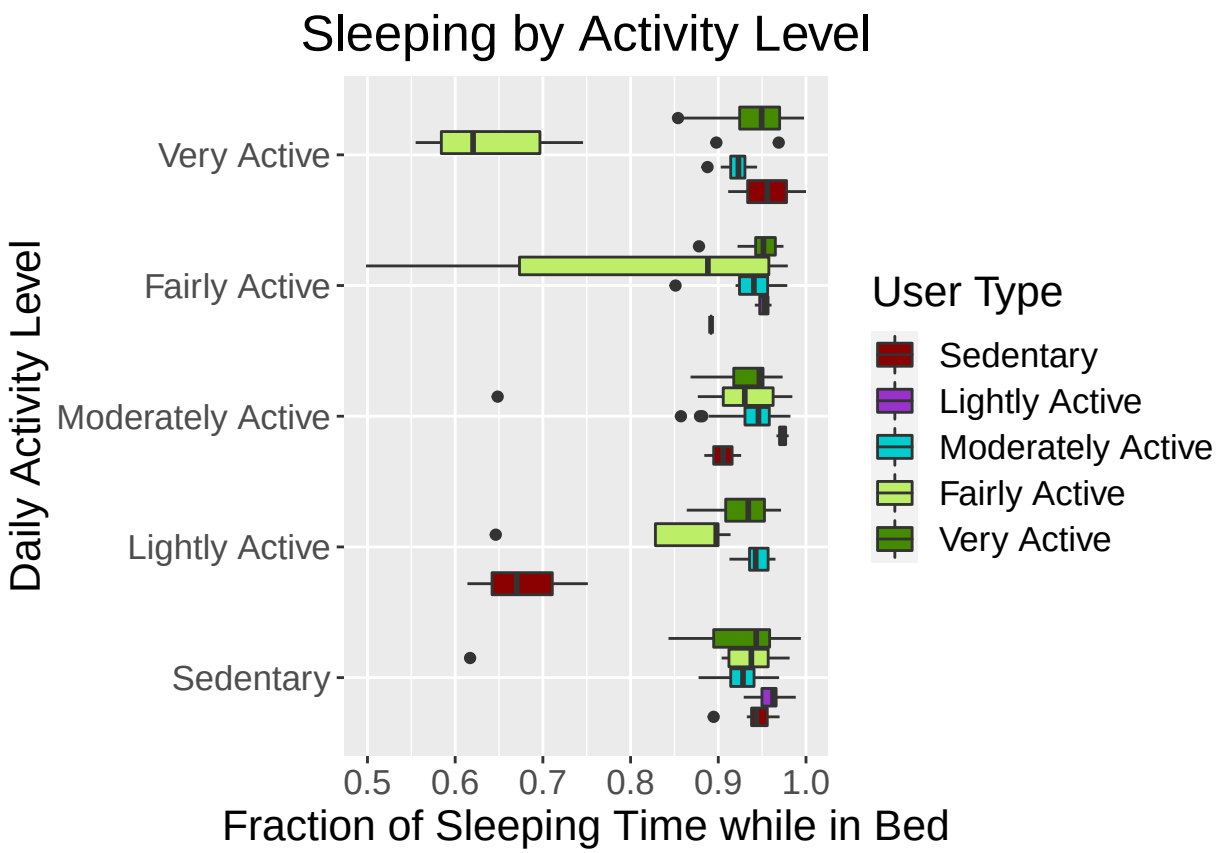
8.3 Relationship Between Activity and Sleep

Let's look now at the effect of the activity level on sleep patterns.

```
#Looking at how many hours do each group sleep per daily activity
ggplot(merge_daily, aes(activity_level, TotalMinutesAsleep/60, fill=user_type)) +
    geom_boxplot() +
    labs(title="Sleeping by Activity Level", x="Daily Activity Level",y="Hours Sleeping", fill="User Ty
    theme(text = element_text(size = 16),plot.title = element_text(hjust = 0.5)) +
  coord_flip()+
  scale_fill_manual(values = c("Sedentary"="darkred",
                               "Lightly Active"="darkorchid",
                               "Moderately Active"="cyan3",
                               "Fairly Active" ="darkolivegreen2",
                               "Very Active" ="chartreuse4"))
```

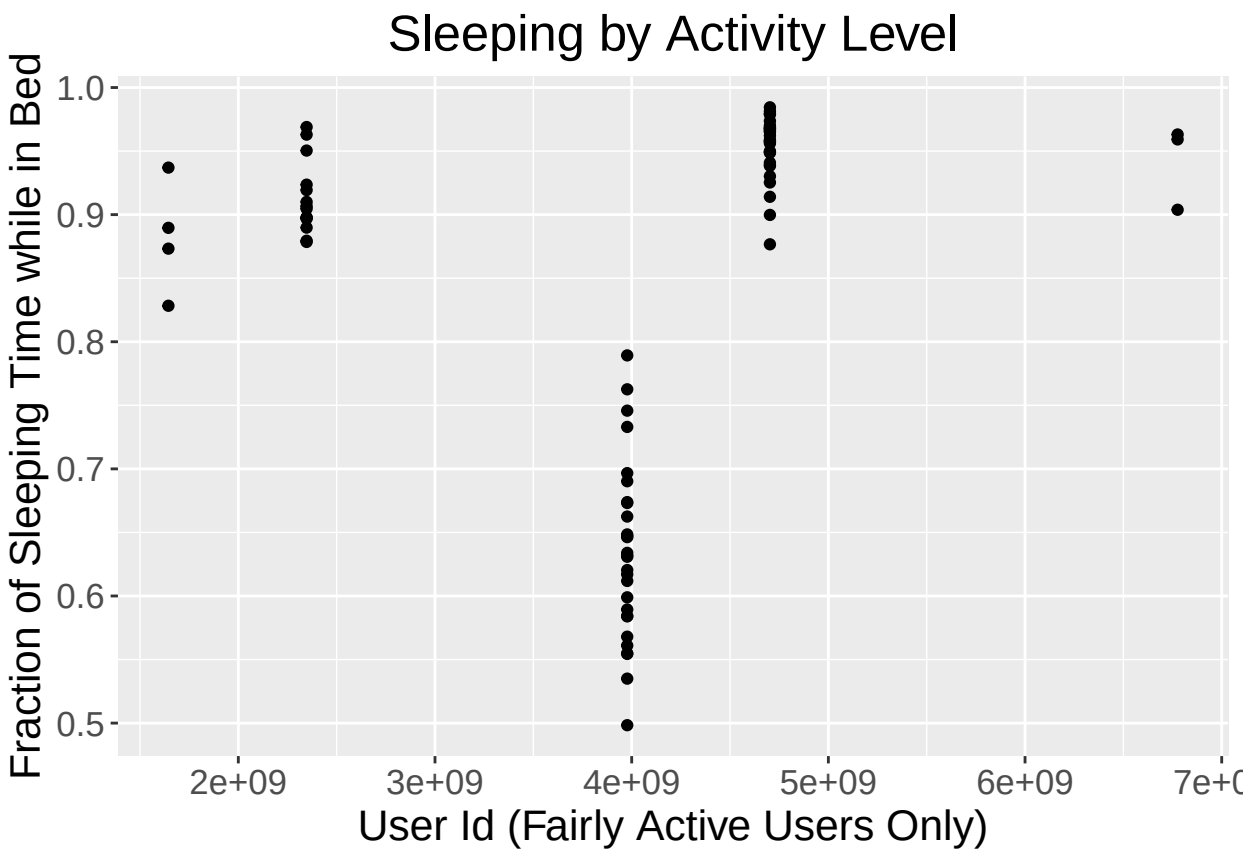



```
#Looking at how effective they are at falling asleep
ggplot(merge_daily, aes(activity_level, TotalMinutesAsleep/TotalTimeInBed, fill=user_type)) +
    geom_boxplot() +
    labs(title="Sleeping by Activity Level", x="Daily Activity Level",y="Fraction of Sleeping Time while
    theme(text = element_text(size = 16),plot.title = element_text(hjust = 0.5)) +
  coord_flip()+
  scale_fill_manual(values = c("Sedentary"="darkred",
                               "Lightly Active"="darkorchid",
                               "Moderately Active"="cyan3",
                               "Fairly Active" ="darkolivegreen2",
                               "Very Active" ="chartreuse4"))
```

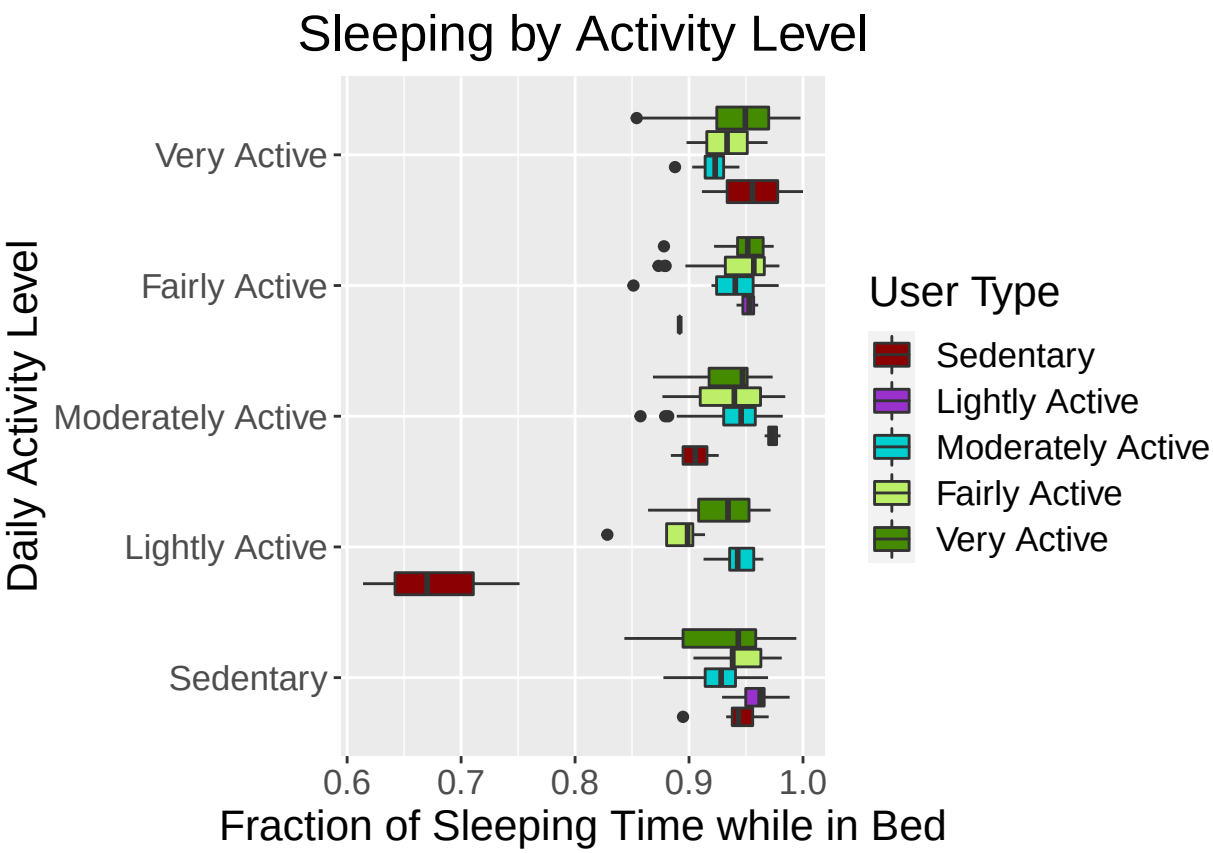


Interestingly, all user types tend to sleep more hours the days they are sightly active than any other day. While very active users tend to maintain a constant 7.5 hours of sleep, fairly active users seem to sleep worse the days they are sedentary, and sedentary users seem to not sleep much at all the days they are very active. Combined with a high ratio of minutes sleeping over minutes laying down, this might indicate that overall sedentary users try to fit all activity into few days, thus not having enough hours for a good rest afterwards. Interestingly, sedentary users also show a low ratio of sleeping time while in bed for lightly active days, which might indicate that they rest and then try to do a little activity. The large dispersion in the data for fairly active users does not make much sense, so I want to take a deeper look into this user cathegory:

```
#Checking what is going on with the fairly active group
ggplot(data=subset(merge_daily, user_type %in% c("Fairly Active")),aes(y=TotalMinutesAsleep/TotalTimeInB
    geom_point() +
    labs(title="Sleeping by Activity Level", x="User Id (Fairly Active Users Only)",y="Fraction of Sleep
    theme(text = element_text(size = 16),plot.title = element_text(hjust = 0.5))
```




```
#plotting again without the outlier user
`%notin%` <- Negate(`%in%`)
ggplot(data=subset(merge_daily, Id %notin% 3977333714), aes(activity_level, TotalMinutesAsleep/TotalTime
    geom_boxplot() +
    labs(title="Sleeping by Activity Level", x="Daily Activity Level",y="Fraction of Sleeping Time while
    theme(text = element_text(size = 16),plot.title = element_text(hjust = 0.5)) +
  coord_flip()+
  scale_fill_manual(values = c("Sedentary"="darkred",
                               "Lightly Active"="darkorchid",
                               "Moderately Active"="cyan3",
                               "Fairly Active" ="darkolivegreen2",
                               "Very Active" ="chartreuse4"))
```



It is clear that one of the fairly active users might have an insomnia problem, since some days he can only sleep half of the time he is in bed despite also having a low total sleeping time. Removing this outlier from the data provides a clearer picture on how daily activity affects sleep:

- Very active users tend to have more consistent sleeping behaviors
- Sedentary and low activity days generally result in a worse sleep
- Large increases on activity from the baseline can have bad repercussions on sleep

9. Conclusions and Recommendations

Given the limitations of the data, I can only offer recommendations on the app and membership Bellabeat products.

1. Doing at least 8000 steps a day correlates with higher overall activity, so daily encouragement to do more steps could snowball into higher activity levels for all users
2. An early start of the day (between 6 and 7 am) tends to result in higher overall activity, thus encouraging waking up early and starting the day with a brisk walk or light exercise can be a great opportunity to improve overall health
3. A reminder after work to work out can also be effective in increasing overall activity levels
4. Consistent activity results in better daily sleep, but large sudden increases on activity from the baseline can have bad repercussions on sleep. This could be tracked and added as a feature of the premium membership as to provide the best guidance while increasing daily activity and improving overall health.
5. Users forget to use smart fitness devices when they charge them. A notification indicating the device has been charged could lead to better data and improved user retention.

Further analysis should be performed with more recent data, larger amount of users, and more reports on daily sleep and weight. Fat percentage and sex of the user should also be collected since it can have a high impact on the calories burnt and activity recommended which could provide more personalized recommendations,

especially for woman during the period week or pregnancy.