Bellabeat

2022-04-04

Casestudy in R for Bellabeat

by Irina Khafizova

Table of Contents

Table of Contents

1. Introduction

Bellabeat is a high-tech company that manufactures health-focused smart devices for women, founded in 2013 by Urška Sršen and Sando Mur. Their products collect data on nutrition, activity, sleep and stress empowering women with knowledge which help them better understand their habits and make healthier decisions.

Here is a list of their current products:

Bellabeat app: provides users with health data related to their activity, sleep, stress, menstrual cycle and mindfulness habits.

Leaf: a wellness tracker that can be worn as a bracelet,necklace or clip, connected to the Bellabeat app to track activity, sleep and stress.

Time: a wellness watcg that tracks activity,sleep and stress, connected to the Bellabeat app to get insights into the user's wellness.

Spring: a water bottle that tracks daily water intake, connected to the Bellabeat app to track hydration levels.

Bellabeat membership: a subscription-bases membership program for users, with 24/7 access to fully personalized guidance on nutrition, activity, sleep, health, beauty and mindfulness based on lifestyle and goals.

2. The Task

Business task This case study focuses on analyzing smart device usage data to gain insight into how consumers are using the Bellabeat device, which will help unlock new growth opportunities.

Stakeholders Urška Sršen - Bellabeat's co-founder and Chief Creative Officer Sando Mur - Bellabeat's co-founder and key member of Bellabeat executive team

3. Prepare

Datasets The data source used for this analysis is FitBit Fitness Tracker Data, which is a public dataset made available through Mobius.

Data privacy and accessibility When verifying the Metadata we can confirm that the dataset is an open source. The dataset has been 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

Data organization The data comes in 18 .csv files with several rows for each subject (ID). Part of the files have a long format, and others have a wide format.

Data limitations The data is original, comprehensive and cited. However, it includes a small sample, a small period of data collection (31 days) and it doesn't show any demographic information, which translates in highly possibly biased data. Another limitation is that the data is not current (2016).

4. Process Phase

4.1 Uploading the data frames For this project, we will be focusing on files with daily data.

```
daily_activity <- read_csv("/Users/irina/Desktop/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.
```

```
daily_intensities <- read_csv("/Users/irina/Desktop/Fitabase Data 4.12.16-5.12.16/dailyIntensities_merg
```

```
## 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.
```

```
daily_calories <- read_csv("/Users/irina/Desktop/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.
```

```
daily_steps <- read_csv("/Users/irina/Desktop/Fitabase Data 4.12.16-5.12.16/dailySteps_merged.csv")
```

```
## Rows: 940 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDay
## dbl (2): Id, StepTotal
##
## 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.
```

```
daily_sleep <- read_csv("/Users/irina/Desktop/Fitabase Data 4.12.16-5.12.16/sleepDay_merged.csv")
```

```
## Rows: 413 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): SleepDay
## dbl (4): Id, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed
##
## 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.
```

4.2. Data frames review and data cleaning Let's take a look at the information contained by each data frame

```
colnames(daily_activity)
```

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

```
colnames(daily_calories)
```

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

```
colnames(daily_intensities)
```

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

```
colnames(daily_steps)
```

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

```
colnames(daily_sleep)
```

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

The data set "daily_activities" contains the information from daily_steps, daily_intensities, daily_calories data sets, therefore, we can safely remove them and proceed.

```
rm(daily_steps, daily_calories, daily_intensities)
```

```
id_summary1 <- data.frame(ids_daily_activity = n_distinct(daily_activity$Id), ids_daily_sleep = n_distin
ids_summary2 <- gather(id_summary1, dataframe, IDs, factor_key=TRUE) %>%
  arrange(desc(IDs))
print(ids_summary2)
```

Then, let's check out the user participation.

```
##            dataframe IDs
## 1 ids_daily_activity  33
## 2    ids_daily_sleep  24
```

Structure of the data frames: Let's take a look at how the data is organized

```
glimpse(daily_activity)
```

```
## Rows: 940
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/~
## $ TotalSteps               <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 13019~
## $ TotalDistance            <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
## $ TrackerDistance          <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance       <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5~
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3~
## $ LightActiveDistance      <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes        <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~
## $ FairlyActiveMinutes      <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~
## $ LightlyActiveMinutes     <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~
## $ SedentaryMinutes         <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~
## $ Calories                 <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203~
```

```
glimpse(daily_sleep)
```

```
## Rows: 413
## Columns: 5
## $ Id                 <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150~
## $ SleepDay           <chr> "4/12/2016 12:00:00 AM", "4/13/2016 12:00:00 AM", "~
## $ TotalSleepRecords  <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ TotalMinutesAsleep <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, 430, 2~
## $ TotalTimeInBed     <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, 449, 3~
```

We notice that the column names aren't clear enough

```
daily_activity <- clean_names(daily_activity)
daily_sleep <- clean_names(daily_sleep)
```

Moreover, it appears the dates are not in the right format

```
daily_sleep <- daily_sleep %>%
  mutate(sleep_day = as_date(sleep_day, format = "%m/%d/%Y"))
daily_activity <- daily_activity %>%
  mutate(activity_date = as_date(activity_date, format = "%m/%d/%Y"))
```

Now, let's run the glimpse function again to ensure the dates are in proper format

```
glimpse(daily_activity)
```

```
## Rows: 940
## Columns: 15
## $ id                       <dbl> 1503960366, 1503960366, 1503960366, 1503960~
## $ activity_date            <date> 2016-04-12, 2016-04-13, 2016-04-14, 2016-0~
```

```
## $ total_steps                <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 130~
## $ total_distance             <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9~
## $ tracker_distance           <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9~
## $ logged_activities_distance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ very_active_distance       <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3~
## $ moderately_active_distance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1~
## $ light_active_distance      <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5~
## $ sedentary_active_distance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ very_active_minutes        <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66,~
## $ fairly_active_minutes      <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, ~
## $ lightly_active_minutes     <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205~
## $ sedentary_minutes          <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 8~
## $ calories                   <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 2~
```

```
glimpse(daily_sleep)
```

```
## Rows: 413
## Columns: 5
## $ id                   <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 1~
## $ sleep_day            <date> 2016-04-12, 2016-04-13, 2016-04-15, 2016-04-16, ~
## $ total_sleep_records  <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ total_minutes_asleep <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, 430,~
## $ total_time_in_bed    <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, 449,~
```

```
sum(duplicated(daily_activity))
```

Checking for duplicates:

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

```
sum(duplicated(daily_sleep))
```

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

There are 3 duplicates in the daily_sleep data frame. Let's remove them

```
daily_sleep <- daily_sleep %>%
  distinct()
```

Let's check for duplicates again

```
sum(duplicated(daily_sleep))
```

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

Next step is removing empty fields

```
daily_activity <- daily_activity %>%
  drop_na()
daily_sleep <- daily_sleep %>%
  drop_na()
```

Merging data frames In order to merge the data frames, we will use their common values: id and date. First, we will need to rename the date columns

```
daily_activity <- daily_activity %>%
  rename(date = activity_date)
daily_sleep <- daily_sleep %>%
  rename(date = sleep_day)
```

Then, we can merge the data frames

```
daily_activity_sleep <- merge(daily_activity, daily_sleep, by=c("id","date"))
glimpse(daily_activity_sleep)
```

```
## Rows: 410
## Columns: 18
## $ id                          <dbl> 1503960366, 1503960366, 1503960366, 1503960~
## $ date                        <date> 2016-04-12, 2016-04-13, 2016-04-15, 2016-0~
## $ total_steps                 <dbl> 13162, 10735, 9762, 12669, 9705, 15506, 105~
## $ total_distance              <dbl> 8.50, 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6~
## $ tracker_distance            <dbl> 8.50, 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6~
## $ logged_activities_distance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ very_active_distance        <dbl> 1.88, 1.57, 2.14, 2.71, 3.19, 3.53, 1.96, 1~
## $ moderately_active_distance <dbl> 0.55, 0.69, 1.26, 0.41, 0.78, 1.32, 0.48, 0~
## $ light_active_distance       <dbl> 6.06, 4.71, 2.83, 5.04, 2.51, 5.03, 4.24, 4~
## $ sedentary_active_distance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ very_active_minutes         <dbl> 25, 21, 29, 36, 38, 50, 28, 19, 41, 39, 73,~
## $ fairly_active_minutes       <dbl> 13, 19, 34, 10, 20, 31, 12, 8, 21, 5, 14, 2~
## $ lightly_active_minutes      <dbl> 328, 217, 209, 221, 164, 264, 205, 211, 262~
## $ sedentary_minutes           <dbl> 728, 776, 726, 773, 539, 775, 818, 838, 732~
## $ calories                    <dbl> 1985, 1797, 1745, 1863, 1728, 2035, 1786, 1~
## $ total_sleep_records         <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ total_minutes_asleep        <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361~
## $ total_time_in_bed           <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384~
```

5. Analyze and Share

```
daily_average <- daily_activity_sleep %>%
  group_by(id) %>%
  summarise(avg_steps = mean(total_steps), avg_cals = mean(calories),
            avg_sleep = mean(total_minutes_asleep),
            avg_in_bed = mean(total_time_in_bed))
glimpse(daily_average)
```

5.1 Summarizing values

```
## Rows: 24
## Columns: 5
## $ id         <dbl> 1503960366, 1644430081, 1844505072, 1927972279, 2026352035,~
## $ avg_steps  <dbl> 12405.680, 7967.750, 3477.000, 1490.000, 5618.679, 5079.000~
## $ avg_cals   <dbl> 1872.280, 2977.750, 1676.333, 2316.200, 1540.786, 1804.000,~
## $ avg_sleep  <dbl> 360.2800, 294.0000, 652.0000, 417.0000, 506.1786, 61.0000, ~
## $ avg_in_bed <dbl> 383.2000, 346.0000, 961.0000, 437.8000, 537.6429, 69.0000, ~
```

5.3 Activity Level We will be classifying the levels according to this article, "How Many Steps a Day is Considered Active

```
## # A tibble: 4 x 2
##   steps_day   level
##   <chr>       <chr>
## 1 <5000       Sedentary
## 2 5000 - 7499 Low active
## 3 7500 - 9999 Somewhat active
## 4 >10000      Active
```

Let's assign this classification to our data

```
average_levels <- daily_average %>%
  mutate(level = case_when(
  avg_steps < 5000 ~ 'Sedentary',
  avg_steps >= 5000 & avg_steps <= 7499 ~ 'Low Active',
  avg_steps >= 7500 & avg_steps <= 9999 ~ 'Somewhat Active',
  avg_steps >= 10000 ~ 'Active'
  ))

average_levels
```

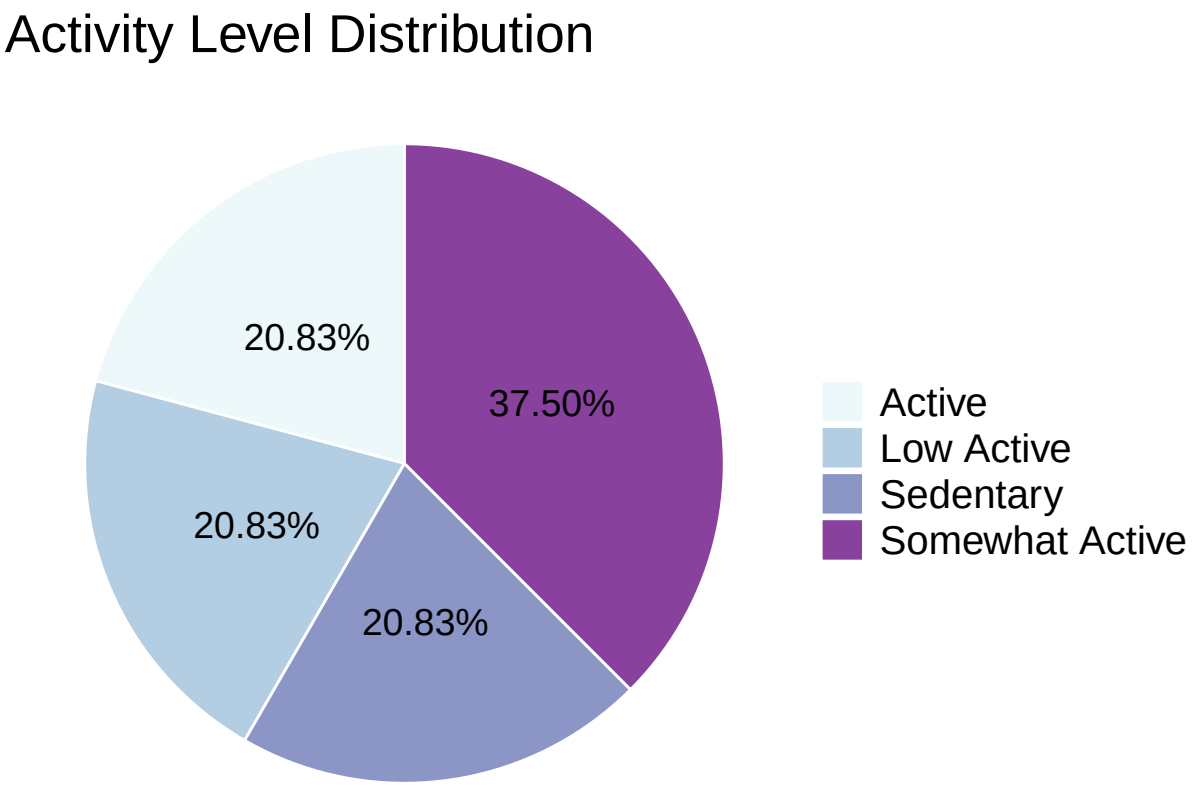
```
## # A tibble: 24 x 6
##            id avg_steps avg_cals avg_sleep avg_in_bed level
##         <dbl>     <dbl>    <dbl>     <dbl>      <dbl> <chr>
##  1 1503960366    12406.    1872.      360.       383. Active
##  2 1644430081     7968.    2978.      294        346  Somewhat Active
##  3 1844505072     3477     1676.      652        961  Sedentary
##  4 1927972279     1490     2316.      417        438. Sedentary
##  5 2026352035     5619.    1541.      506.       538. Low Active
##  6 2320127002     5079     1804        61         69  Low Active
##  7 2347167796     8533.    1971.      447.       491. Somewhat Active
##  8 3977333714    11218     1560.      294.       461. Active
##  9 4020332650     6597.    3195       349.       380. Low Active
## 10 4319703577     7125.    2025.      477.       502. Low Active
## # ... with 14 more rows
```

We need to get percentage of each activity level

```
activity_levels_percentage <- average_levels %>%
  group_by(level) %>%
  summarise(total_level=n()) %>%
  mutate(percentage = (total_level / sum(total_level))) %>%
```

```
  mutate(percentage = formattable::percent(percentage)) %>%
  arrange((level))
```

```
ggplot(activity_levels_percentage, aes(x="", y=percentage, fill=level))+
  geom_bar(width=1, stat="identity", color="white")+
  coord_polar("y", start = 0)+
  geom_text(aes(label=percentage), position = position_stack(vjust = 0.5), size = 5)+
  labs(title = "Activity Level Distribution")+
  scale_fill_brewer(palette ="BuPu")+
  guides(fill = guide_legend(title=NULL))+
  theme_void()+
  theme(plot.title = element_text(size=20), legend.text = element_text(size=15))
```



Insights Looking at our diagram, 37.50% are somewhat active, with an average of 7500 and 9999 steps, meanwhile Low Active, Sedentary, and Active users share equal parts of 20.83%.

5.3 Sleep levels It is recommended that adults gets between 7 to 9 hours of sleep each night. Therefore, we will establish the following rules

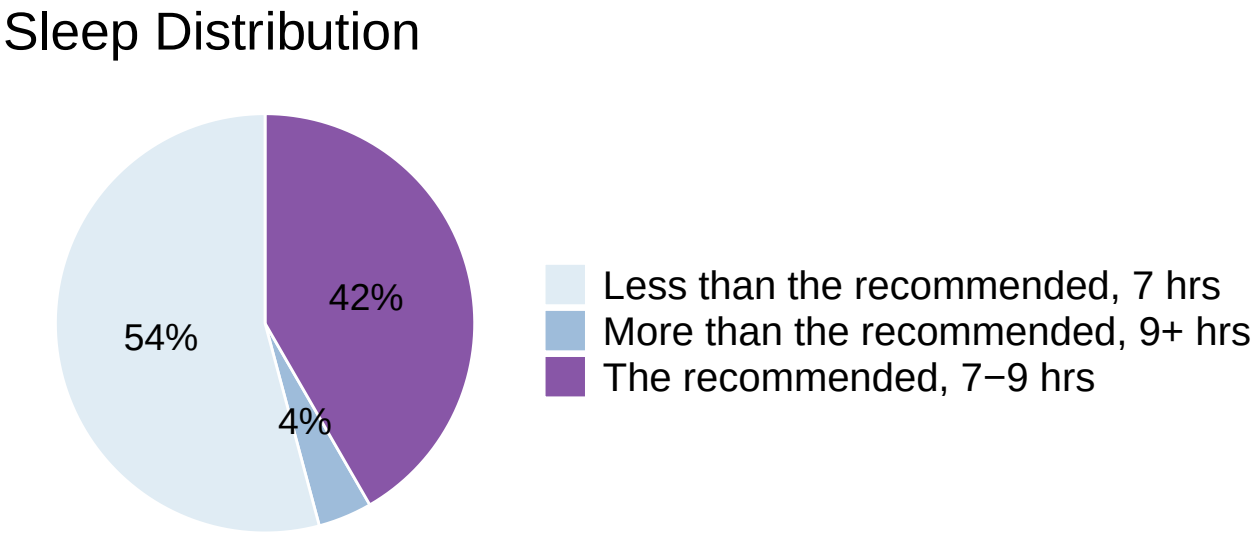
- Less, than the recommended, 7 hrs
- The recommended, 7-9 hrs
- More, than the recommended, 9+ hrs

```r
average_levels <- daily_average %>%
  mutate(avg_sleep = case_when(
    avg_sleep < 420 ~ 'Less than the recommended, 7 hrs',
    avg_sleep >= 420 & avg_sleep < 540 ~ 'The recommended, 7-9 hrs',
    avg_sleep >= 540 ~ 'More than the recommended, 9+ hrs'
  ))
```

Now, let's assign percentage by level of sleep

```r
sleep_average_levels <- average_levels %>%
  group_by(avg_sleep) %>%
  summarise(n_users=n()) %>%
  mutate(users_percentage = (n_users / sum(n_users))) %>%
  mutate(users_percentage = formattable::percent(users_percentage, 0)) %>%
  arrange(users_percentage)
```

```r
ggplot(sleep_average_levels, aes(x="", y=users_percentage, fill=avg_sleep))+
  geom_bar(width=1, stat="identity", color="white")+
  coord_polar("y", start = 0)+
  geom_text(aes(label=users_percentage), position = position_stack(vjust = 0.5), size = 5)+
  labs(title = "Sleep Distribution")+
  scale_fill_brewer(palette ="BuPu")+
  guides(fill = guide_legend(title=NULL))+
  theme_void()+
  theme(plot.title = element_text(size=20), legend.text = element_text(size=15))
```



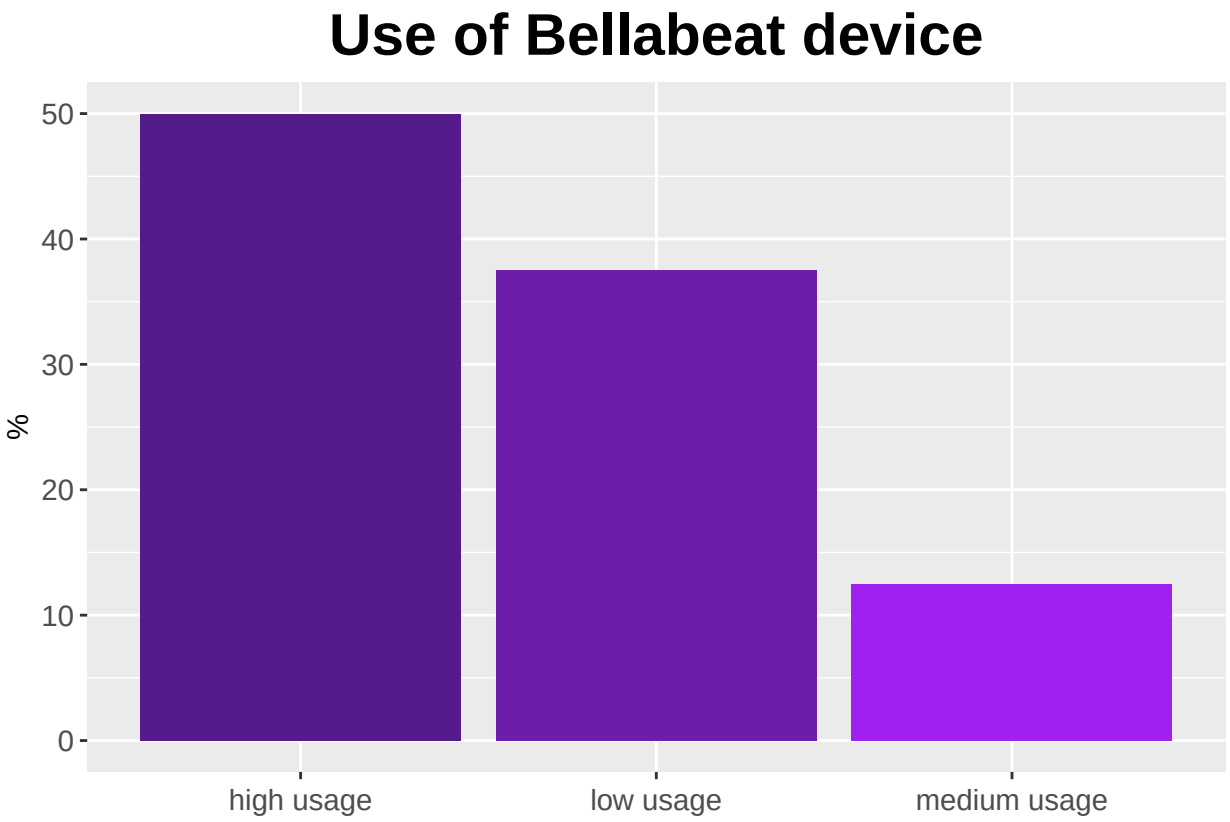
Insight We see that more than 50% of our users sleep less than the recommended amount of hours, while around 4% sleep more than the recommended, leaving 42% with the recommended amount of sleep.

5.4 Use of Bellabeat Devices We want to see how frequent each participant uses the devices. We will categorize them by the following

- 1 to 10 days: low usage
- 10 to 20 days: medium usage
- 20+ days: high usage

```
devices_usage <- daily_activity_sleep %>%
  group_by(id) %>%
  summarize(date=sum(n())) %>%
  rename(days_used = date) %>%
  mutate(usage = case_when(
    days_used >= 1 & days_used <= 10 ~ 'low usage',
    days_used >= 11 & days_used <= 20 ~ 'medium usage',
    days_used > 20 ~ 'high usage'
  )) %>%
  group_by(usage) %>%
  summarise(id=n()) %>%
  mutate(usage_percentage = (id / sum(id))) %>%
  mutate(usage_percentage = formattable::percent(usage_percentage)) %>%
  arrange(usage_percentage)
```

```
ggplot(data=devices_usage)+
  geom_col(mapping = aes(x=usage, y=(usage_percentage*100), fill=(usage_percentage*100)))+
  ylab("%") +
  xlab("")+
  scale_fill_gradient(low = "purple", high = "purple4")+
  theme(plot.title = element_text(hjust = 0.5,vjust= 1, size = 22, face = "bold"),
        axis.text.x = element_text(angle = 0, size=11, vjust= 0.4),
        axis.text.y = element_text(size=11),
        legend.title = element_blank(),
        legend.position = "none")+
  labs(title = "Use of Bellabeat device")
```



Insight Half of the users on average use the Bellabeat device more than 20 days, while 37.50% are within the low usage category

5.5 Correlation We will only be focusing on device usage vs steps correlation.

```
device_usage_day <- daily_activity_sleep %>%
  group_by(id) %>%
  summarize(date=sum(n())) %>%
  rename(days_used = date) %>%
  mutate(usage = case_when(
    days_used >= 1 & days_used <= 10 ~ 'low usage',
    days_used >= 11 & days_used <= 20 ~ 'medium usage',
    days_used > 20 ~ 'high usage'
  ))
```

Merging usage and steps data frames

```
usage_vs_steps <- merge(daily_activity, device_usage_day, by=c("id"))

glimpse(usage_vs_steps)
```

```
## Rows: 713
## Columns: 17
```

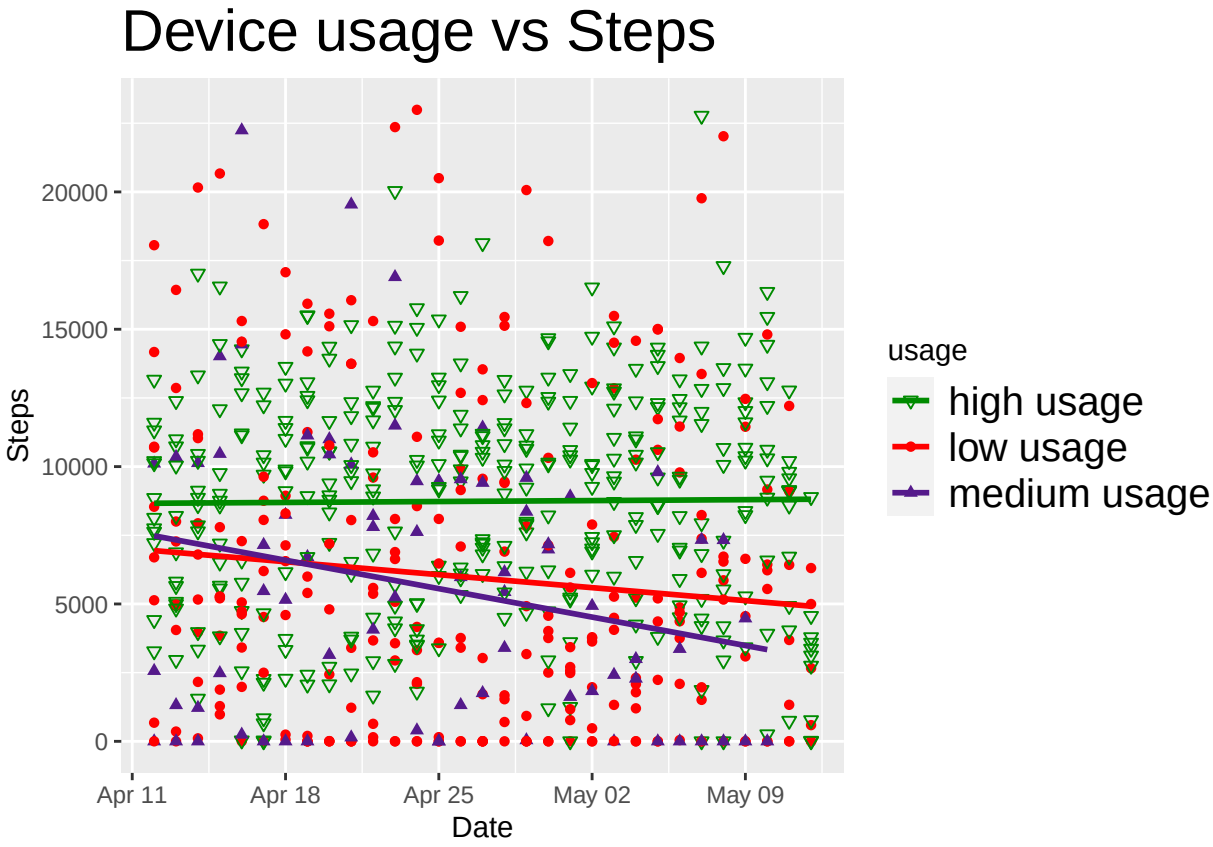
```
## $ id                          <dbl> 1503960366, 1503960366, 1503960366, 1503960~
## $ date                        <date> 2016-05-07, 2016-05-06, 2016-05-01, 2016-0~
## $ total_steps                 <dbl> 11992, 12159, 10602, 14673, 13162, 10735, 1~
## $ total_distance              <dbl> 7.71, 8.03, 6.81, 9.25, 8.50, 6.97, 9.80, 8~
## $ tracker_distance            <dbl> 7.71, 8.03, 6.81, 9.25, 8.50, 6.97, 9.80, 8~
## $ logged_activities_distance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ very_active_distance        <dbl> 2.46, 1.97, 2.29, 3.56, 1.88, 1.57, 5.29, 2~
## $ moderately_active_distance <dbl> 2.12, 0.25, 1.60, 1.42, 0.55, 0.69, 0.57, 1~
## $ light_active_distance       <dbl> 3.13, 5.81, 2.92, 4.27, 6.06, 4.71, 3.94, 4~
## $ sedentary_active_distance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ very_active_minutes         <dbl> 37, 24, 33, 52, 25, 21, 73, 45, 48, 16, 31,~
## $ fairly_active_minutes       <dbl> 46, 6, 35, 34, 13, 19, 14, 24, 28, 12, 23, ~
## $ lightly_active_minutes      <dbl> 175, 289, 246, 217, 328, 217, 216, 250, 189~
## $ sedentary_minutes           <dbl> 833, 754, 730, 712, 728, 776, 814, 857, 782~
## $ calories                    <dbl> 1821, 1896, 1820, 1947, 1985, 1797, 2013, 1~
## $ days_used                   <int> 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25,~
## $ usage                       <chr> "high usage", "high usage", "high usage", "~
```

```
high_usage <- subset(usage_vs_steps, usage == "high usage")
mild_usage <- subset(usage_vs_steps, usage == "mild usage")
low_usage <- subset(usage_vs_steps, usage == "low usage")
```

Graphing

```
ggplot(usage_vs_steps, aes(x=date, y=total_steps, fill=usage)) +
  geom_point(aes(shape=usage, color=usage))+
  scale_shape_manual(values=c(6, 16, 17))+
  scale_color_manual(values=c('green4','red', 'purple4'))+
  scale_size_manual(values=c(2,3,4))+
  theme(legend.position="right", plot.title = element_text(size=22), legend.text = element_text(size=15)
  geom_smooth(method="lm", se=FALSE, aes(color=factor(usage)))+
  labs(title = "Device usage vs Steps", x = "Date", y = "Steps")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Insght We notice that the users with high usage of devices attain consistent amount of steps that is very close to the recommended. However, users with medium and low usages have a decreasing amount of total steps through the time period.

6. Solutions

With the completed analysis we have reached the following conclusions: a big chunk of the users(37.50%) are only somewhat active, with an average between 7500 and 9999 steps a day. Around 50% of the users get less than the recommended amount of sleep of 7 to 9 hours. Around half of the users attained high usage of the devices within the mentioned time period. However, most of the other have attain low device usage of 10 days or less. Finally, looking at the correlation between steps and device usage, it is clear that participants with high usage also maintained a consistent amount of steps very close to the recommended.

Based on these finding, users have the following areas for improvement:

- exceed 10,000 steps a day
- sleep at least 7 hours a night
- increase device usage throughout a month

My recommendations: Exceeding 10,000 steps a day

- Daily reminders to take longer routes while doing daily routines such as travelling to work by foot, taking the stairs, getting of public transportation a stop or two early and walk the rest.

- Guides of simple workouts at home
- Presents in the form of promo-codes, when achieving a desired amount of steps in a month
- Health articles on the importance on achieving a minimum of 10,000 steps a day

Sleeping at least 7 hours a night - Setting up bedtime within the app and sending reminders a certain amount of time prior to bedtime to reduce light and workload

- Tips on how to improve pre-bed time routine (ex. disconnecting from electronic devices)
- Encourage relaxing activities such as reading
- Monitoring caffeine intake hours before bedtime
- Reducing stress levels by practicing meditation

Increasing device usage: - Creating more stylish options for different occasions

- Ability to connect different apps within a phone to the Bellabeat app (ex. Health app)

For further analysis, I would recommend Bellabeat to store more data regarding their users such as demographics, age, occupation, lifestyle. One of the ways of obtaining such data is periodic surveys within the app