How Can a Wellness Technology Company Play It Smart?

Nozhin Azarpanah

Scenario

Bellabeat is a high-tech manufacturer of health-focused products for women. It offers different smart devices that collect data on activity, sleep, stress, and reproductive health to empower women with knowledge about their own health and habits.

Bellabeat offers a range of products, among which is the Bellabeat app. The Bellabeat app provides users with health data related to their activity, sleep, stress, menstrual cycle, and mindfulness habits. This data can help users better understand their current habits and make healthy decisions. The Bellabeat app connects to their line of smart wellness products. Bellabeat is a successful small company, however, Urška Sršen, co-founder and Chief Creative Officer of Bellabeat, believes that it has the potential to become even a larger player in the global smart device market.

In this project, I will present my analysis on the smart devices' fitness data to the Bellabeat executive team along with my high-level recommendations for Bellabeat's marketing strategy to help unlock new growth opportunities for the company.

---

1. Ask Phase

Business task

Identify potential opportunities for growth and recommendations for the Bellabeat marketing strategy improvement based on the trends in smart device usage.

Questions to answer

1. What are some trends in smart device usage?
2. How could these trends apply to Bellabeat customers?
3. How could these trends help influence Bellabeat marketing strategy?

Deliverables

This project will contain the following deliverables:

1. A clear summary of the business task
2. A description of all data sources used
3. Documentation of any cleaning or manipulation of data
4. A summary of your analysis
5. Supporting visualizations and key findings
6. Top high-level content recommendations based on analysis

---

2. Prepare Phase

Source of data:

The data source used for this case study is FitBit Fitness Tracker Data. The data used in this project is stored in Kaggle and was made available through Mobius. This Kaggle data set contains personal fitness tracker from thirty Fitbit users.

Accessibility and privacy of data:

This data set is open-source. The owner has dedicated the work to the public domain by waiving all of his 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.

Information about the data set:

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. Variation between output represents use of different types of Fitbit trackers and individual tracking behaviors/preferences.

Data organization and verification:

There are 18 .CSV documents in this data set. Each document represents different quantitative data tracked by Fitbit. Every user has a unique ID and different rows since data is tracked by day and time.

Data credibility and integrity:

Due to the limitation of size (30 users) and not having any demographic information, I could encounter a sampling bias. I am not sure if the sample is representative of the population as a whole. Furthermore, it's important to note that the data set is not current, and there's the additional constraint of the survey being limited to a duration of two months. In light of these considerations, I will adopt an operational approach for our case study.

---

3. Process Phase

Install & load packages

We first install and load the packages we are going to use during this project.

```r
library("readr")
library("tidyverse")
library("dplyr")
library("janitor")
library("RColorBrewer")
library("ggplot2")
library("ggpubr")
library("scales")
```

Read the data files and create data frames

Throughout this project, our focus will center around five key files containing users’ data on daily activity, calories, intensities, sleep, and steps.

```
daily_activity <- read.csv("~/dataFiles/dailyActivity.csv")
daily_calories <- read.csv("~/dataFiles/dailyCalories.csv")
daily_intensities <- read.csv("~/dataFiles/dailyIntensities.csv")
daily_sleep <- read.csv("~/dataFiles/dailySleep.csv")
daily_steps <- read.csv("~/dataFiles/hourlySteps.csv")
weight_log <- read.csv("~/dataFiles/weightLog.csv")
heart_rate <- read.csv("~/dataFiles/heartRatePerSecond.csv")
```

Here is a summary of the data sets and the first few rows of each:

```
str(daily_activity)
```

```
## ’data.frame':    940 obs. of  15 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps              : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ TotalDistance           : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ Calories                : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```
str(daily_calories)
```

```
## ’data.frame’:    940 obs. of  3 variables:
##  $ Id         : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDay: chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ Calories   : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```
str(daily_intensities)
```

```
## ’data.frame’:    940 obs. of  10 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDay             : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ LightlyActiveMinutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ FairlyActiveMinutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ VeryActiveMinutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LightActiveDistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ VeryActiveDistance      : num  1.88 1.57 2.44 2.14 2.71 ...
```

```r
str(daily_sleep)
```

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

```r
str(daily_steps)
```

```
## ’data.frame’:    22099 obs. of  3 variables:
##  $ Id          : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour: chr  "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00 AM" "4/12/20
##  $ StepTotal   : int  373 160 151 0 0 0 0 0 250 1864 ...
```

```r
str(weight_log)
```

```
## ’data.frame’:    67 obs. of  8 variables:
##  $ Id            : num  1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...
##  $ Date          : chr  "5/2/2016 11:59:59 PM" "5/3/2016 11:59:59 PM" "4/13/2016 1:08:52 AM" "4/21/20
##  $ WeightKg      : num  52.6 52.6 133.5 56.7 57.3 ...
##  $ WeightPounds  : num  116 116 294 125 126 ...
##  $ Fat           : int  22 NA NA NA NA 25 NA NA NA NA ...
##  $ BMI           : num  22.6 22.6 47.5 21.5 21.7 ...
##  $ IsManualReport: chr  "True" "True" "False" "True" ...
##  $ LogId         : num  1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
```

```r
str(heart_rate)
```

```
## ’data.frame’:    1048575 obs. of  3 variables:
##  $ Id   : num  2.02e+09 2.02e+09 2.02e+09 2.02e+09 2.02e+09 ...
##  $ Time : chr  "4/12/2016 7:21" "4/12/2016 7:21" "4/12/2016 7:21" "4/12/2016 7:21" ...
##  $ Value: int  97 102 105 103 101 95 91 93 94 93 ...
```

```r
head(daily_activity)
```

```
##           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(daily_calories)
```

```
##           Id ActivityDay Calories
## 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
```

```
head(daily_intensities)
```

```
##           Id ActivityDay SedentaryMinutes LightlyActiveMinutes
## 1 1503960366   4/12/2016              728                  328
## 2 1503960366   4/13/2016              776                  217
## 3 1503960366   4/14/2016             1218                  181
## 4 1503960366   4/15/2016              726                  209
## 5 1503960366   4/16/2016              773                  221
## 6 1503960366   4/17/2016              539                  164
##   FairlyActiveMinutes VeryActiveMinutes SedentaryActiveDistance
## 1                  13                25                       0
## 2                  19                21                       0
## 3                  11                30                       0
## 4                  34                29                       0
## 5                  10                36                       0
## 6                  20                38                       0
##   LightActiveDistance ModeratelyActiveDistance VeryActiveDistance
## 1                6.06                     0.55               1.88
## 2                4.71                     0.69               1.57
## 3                3.91                     0.40               2.44
## 4                2.83                     1.26               2.14
## 5                5.04                     0.41               2.71
## 6                2.51                     0.78               3.19
```

```r
head(daily_sleep)
```

```
##           Id              SleepDay TotalSleepRecords TotalMinutesAsleep
## 1 1503960366 4/12/2016 12:00:00 AM                 1                327
## 2 1503960366 4/13/2016 12:00:00 AM                 2                384
## 3 1503960366 4/15/2016 12:00:00 AM                 1                412
## 4 1503960366 4/16/2016 12:00:00 AM                 2                340
## 5 1503960366 4/17/2016 12:00:00 AM                 1                700
## 6 1503960366 4/19/2016 12:00:00 AM                 1                304
##   TotalTimeInBed
## 1            346
## 2            407
## 3            442
## 4            367
## 5            712
## 6            320
```

```r
head(daily_steps)
```

```
##           Id          ActivityHour StepTotal
## 1 1503960366 4/12/2016 12:00:00 AM       373
## 2 1503960366  4/12/2016 1:00:00 AM       160
## 3 1503960366  4/12/2016 2:00:00 AM       151
## 4 1503960366  4/12/2016 3:00:00 AM         0
## 5 1503960366  4/12/2016 4:00:00 AM         0
## 6 1503960366  4/12/2016 5:00:00 AM         0
```

```r
head(weight_log)
```

```
##           Id                  Date WeightKg WeightPounds Fat   BMI
## 1 1503960366  5/2/2016 11:59:59 PM     52.6     115.9631  22 22.65
## 2 1503960366  5/3/2016 11:59:59 PM     52.6     115.9631  NA 22.65
## 3 1927972279  4/13/2016 1:08:52 AM    133.5     294.3171  NA 47.54
## 4 2873212765 4/21/2016 11:59:59 PM     56.7     125.0021  NA 21.45
## 5 2873212765 5/12/2016 11:59:59 PM     57.3     126.3249  NA 21.69
## 6 4319703577 4/17/2016 11:59:59 PM     72.4     159.6147  25 27.45
##   IsManualReport        LogId
## 1           True 1.462234e+12
## 2           True 1.462320e+12
## 3          False 1.460510e+12
## 4           True 1.461283e+12
## 5           True 1.463098e+12
## 6           True 1.460938e+12
```

```r
head(heart_rate)
```

```
##           Id           Time Value
## 1 2022484408 4/12/2016 7:21    97
## 2 2022484408 4/12/2016 7:21   102
## 3 2022484408 4/12/2016 7:21   105
## 4 2022484408 4/12/2016 7:21   103
## 5 2022484408 4/12/2016 7:21   101
## 6 2022484408 4/12/2016 7:22    95
```

Verify the number of users

Let's check out the number of users in each data set:

```
n_distinct(daily_activity$Id)
```

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

```
n_distinct(daily_calories$Id)
```

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

```
n_distinct(daily_intensities$Id)
```

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

```
n_distinct(daily_sleep$Id)
```

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

```
n_distinct(daily_steps$Id)
```

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

```
n_distinct(weight_log$Id)
```

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

```
n_distinct(heart_rate$Id)
```

```
## [1] 7
```

Considering the limited data available in the weight log and heart rate, we refrain from conducting an analysis on users' weights and heart rate due to the data sets' small size.

Remove duplicates and nulls

To make sure the data is clean, we inspect each data set for duplicate and null values, removing any instances found.

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

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

```
sum(duplicated(daily_calories))
```

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

```
sum(duplicated(daily_intensities))
```

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

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

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

```
sum(duplicated(daily_steps))
```

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

```
sum(is.na(daily_activity))
```

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

```
sum(is.na(daily_calories))
```

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

```
sum(is.na(daily_intensities))
```

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

```
sum(is.na(daily_sleep))
```

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

```
sum(is.na(daily_steps))
```

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

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

sum(duplicated(daily_sleep))
```

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

We standardize the column names by converting them to lowercase and employ the clean_names() function to ensure uniqueness and consistency across all columns.

```
clean_names(daily_activity)
daily_activity <- rename_with(daily_activity, tolower)

clean_names(daily_calories)
daily_calories <- rename_with(daily_calories, tolower)

clean_names(daily_intensities)
daily_intensities <- rename_with(daily_intensities, tolower)

clean_names(daily_sleep)
daily_sleep <- rename_with(daily_sleep, tolower)

clean_names(daily_steps)
daily_steps <- rename_with(daily_steps, tolower)
```

Data standardization and type conversion

Additionally, we verify that the column containing the date or time of an activity is consistently labeled as 'date' in each data set, we enhance data integrity by converting the data type of the 'date' column from 'character' ('chr') to 'Date' across all data sets.

```
# Rename columns in daily_activity
daily_activity <- daily_activity %>%
  rename(date = activitydate) %>%
  mutate(date = as.Date(date, format = "%m/%d/%Y"))

# Rename columns in daily_calories
daily_calories <- daily_calories %>%
  rename(date = activityday) %>%
  mutate(date = as.Date(date, format = "%m/%d/%Y"))

# Rename columns in daily_intensities
daily_intensities <- daily_intensities %>%
  rename(date = activityday) %>%
  mutate(date = as.Date(date, format = "%m/%d/%Y"))

# Rename columns in daily_sleep
daily_sleep <- daily_sleep %>%
  rename(date = sleepday) %>%
  mutate(date = as.Date(date, format = "%m/%d/%Y"))

# Rename columns in daily_steps
daily_steps <- daily_steps %>%
  rename(date_time = activityhour) %>%
  mutate(date_time = as.POSIXct(date_time, format = "%m/%d/%Y %I:%M:%S %p" , tz=Sys.timezone())) %>%
  mutate(time = ifelse(format(date_time, "%H:%M:%S") == "00:00:00", "00:00:00", format(date_time, "%H:%
         date = as.Date(date_time)) %>%
  select(-date_time)
```

Here is a summary of our cleaned data sets:

```r
str(daily_activity)
```

```
## ’data.frame’:    940 obs. of  15 variables:
##  $ id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ date                    : Date, format: "2016-04-12" "2016-04-13" ...
##  $ totalsteps              : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ totaldistance           : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ trackerdistance         : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ loggedactivitiesdistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ veryactivedistance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ moderatelyactivedistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ lightactivedistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ sedentaryactivedistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ veryactiveminutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ fairlyactiveminutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ lightlyactiveminutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ sedentaryminutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ calories                : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```r
str(daily_calories)
```

```
## ’data.frame’:    940 obs. of  3 variables:
##  $ id      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ date    : Date, format: "2016-04-12" "2016-04-13" ...
##  $ calories: int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```r
str(daily_intensities)
```

```
## ’data.frame’:    940 obs. of  10 variables:
##  $ id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ date                    : Date, format: "2016-04-12" "2016-04-13" ...
##  $ sedentaryminutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ lightlyactiveminutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ fairlyactiveminutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ veryactiveminutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ sedentaryactivedistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ lightactivedistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ moderatelyactivedistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ veryactivedistance      : num  1.88 1.57 2.44 2.14 2.71 ...
```

```r
str(daily_sleep)
```

```
## ’data.frame’:    410 obs. of  5 variables:
##  $ id                : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ date              : Date, format: "2016-04-12" "2016-04-13" ...
##  $ totalsleeprecords : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ totalminutesasleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ totaltimeinbed    : int  346 407 442 367 712 320 377 364 384 449 ...
```

```
str(daily_steps)
```

```
## 'data.frame':    22099 obs. of  4 variables:
##  $ id       : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ steptotal: int  373 160 151 0 0 0 0 0 250 1864 ...
##  $ time     : chr  "00:00:00" "01:00:00" "02:00:00" "03:00:00" ...
##  $ date     : Date, format: "2016-04-12" "2016-04-12" ...
```

---

4. Analyze and Share Phase

Statistical summary

Let's start off by getting a statistical summary of the data sets:

```
daily_activity %>%
  select(totaldistance, totalsteps,sedentaryminutes,lightlyactiveminutes, fairlyactiveminutes,veryactive
  summary()
```

```
##  totaldistance      totalsteps    sedentaryminutes lightlyactiveminutes
##  Min.   : 0.000   Min.   :    0   Min.   :   0.0   Min.   :  0.0
##  1st Qu.: 2.620   1st Qu.: 3790   1st Qu.: 729.8   1st Qu.:127.0
##  Median : 5.245   Median : 7406   Median :1057.5   Median :199.0
##  Mean   : 5.490   Mean   : 7638   Mean   : 991.2   Mean   :192.8
##  3rd Qu.: 7.713   3rd Qu.:10727   3rd Qu.:1229.5   3rd Qu.:264.0
##  Max.   :28.030   Max.   :36019   Max.   :1440.0   Max.   :518.0
##  fairlyactiveminutes veryactiveminutes sedentaryactivedistance
##  Min.   :  0.00      Min.   :  0.00    Min.   :0.000000
##  1st Qu.:  0.00      1st Qu.:  0.00    1st Qu.:0.000000
##  Median :  6.00      Median :  4.00    Median :0.000000
##  Mean   : 13.56      Mean   : 21.16    Mean   :0.001606
##  3rd Qu.: 19.00      3rd Qu.: 32.00    3rd Qu.:0.000000
##  Max.   :143.00      Max.   :210.00    Max.   :0.110000
##  lightactivedistance moderatelyactivedistance veryactivedistance    calories
##  Min.   : 0.000      Min.   :0.0000           Min.   : 0.000     Min.   :   0
##  1st Qu.: 1.945      1st Qu.:0.0000           1st Qu.: 0.000     1st Qu.:1828
##  Median : 3.365      Median :0.2400           Median : 0.210     Median :2134
##  Mean   : 3.341      Mean   :0.5675           Mean   : 1.503     Mean   :2304
##  3rd Qu.: 4.782      3rd Qu.:0.8000           3rd Qu.: 2.053     3rd Qu.:2793
##  Max.   :10.710      Max.   :6.4800           Max.   :21.920     Max.   :4900
```

```
daily_sleep %>%
  select(totalminutesasleep, totaltimeinbed) %>%
  summary()
```

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

These are key insights drawn from the summarized statistics of the data sets:

total_distance:

The minimum distance covered is 0, indicating there are instances where no distance was traveled. The majority of the data falls below 7.713 units, with a median of 5.245. The mean distance is slightly higher than the median, suggesting a slightly right-skewed distribution. The maximum recorded distance is 28.030 units.

total_steps:

The number of steps taken ranges from 0 to 36019, with an average of 7638 steps. The distribution appears positively skewed, with a median of 7406 and a mean higher than the median.

sedentary_minutes:

Sedentary minutes range from 0 to 1440, with an average of 991.2 minutes. The majority of data falls within the first quartile, suggesting a right-skewed distribution. The maximum value of 1440 indicates instances of the entire day spent sedentary.

lightly_active_minutes, fairly_active_minutes, very_active_minutes:

These columns indicate the minutes spent in different activity levels. There is a wide range of activity levels, with some users having 0 minutes in certain categories. The maximum values for fairly and very active minutes are 143 and 210, respectively.

sedentary_active_distance, light_active_distance, moderately_active_distance, very_active_distance:

These columns represent the distance covered in different activity levels. Most of the data in sedentary_active_distance is 0, indicating negligible distance covered during sedentary activities. The maximum values for moderately_active_distance and very_active_distance are 6.48 and 21.92, respectively.

calories:

The minimum recorded calories are 0, indicating cases where no calories were burned. The majority of the data falls between 1828 and 2793 calories. The mean calorie consumption is 2304, with a median of 2134, indicating a right-skewed distribution. The maximum recorded calorie consumption is 4900.

total_minutes_asleep, total_time_in_bed:

The time spent asleep and in bed varies widely, with a minimum of 58 minutes and a maximum of 796 minutes for total_minutes_asleep The average time spent asleep is 419.2 minutes, with an average total time in bed of 458.5 minutes.

User classification by activity level

According to 1000 Steps, the following pedometer indices have been developed to provide a guideline on steps and activity levels:

- 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

We generate a new data frame that contains the average values for distance covered, calories burned, minutes asleep, and total steps taken by individual users.

```r
merged_data <- merge(daily_activity, daily_sleep, by = c("id","date"))

daily_average <- merged_data %>%
  group_by(id) %>%
  summarise(
    mean_daily_distance = mean(totaldistance),
    mean_daily_calories = mean(calories),
    mean_daily_sleep = mean(totalminutesasleep),
    mean_daily_steps = mean(totalsteps)
  )

head(daily_average)
```

```
## # A tibble: 6 x 5
##           id mean_daily_distance mean_daily_calories mean_daily_sleep
##        <dbl>               <dbl>               <dbl>            <dbl>
## 1 1503960366                7.97               1872.             360.
## 2 1644430081                5.79               2978.             294
## 3 1844505072                2.30               1676.             652
## 4 1927972279                1.03               2316.             417
## 5 2026352035                3.49               1541.             506.
## 6 2320127002                3.42               1804               61
## # i 1 more variable: mean_daily_steps <dbl>
```

We categorize each user based on their average daily number of steps, assigning them a specific user type.

```r
user_type <- daily_average %>%
  mutate(user_type = case_when(
    mean_daily_steps < 5000 ~ "Sedentary",
    mean_daily_steps >= 5000 & mean_daily_steps < 7499 ~ "Lightly Active",
    mean_daily_steps >= 7500 & mean_daily_steps < 9999 ~ "Fairly Active",
    mean_daily_steps >= 10000 ~ "Very Active"
  ))

head(user_type)
```

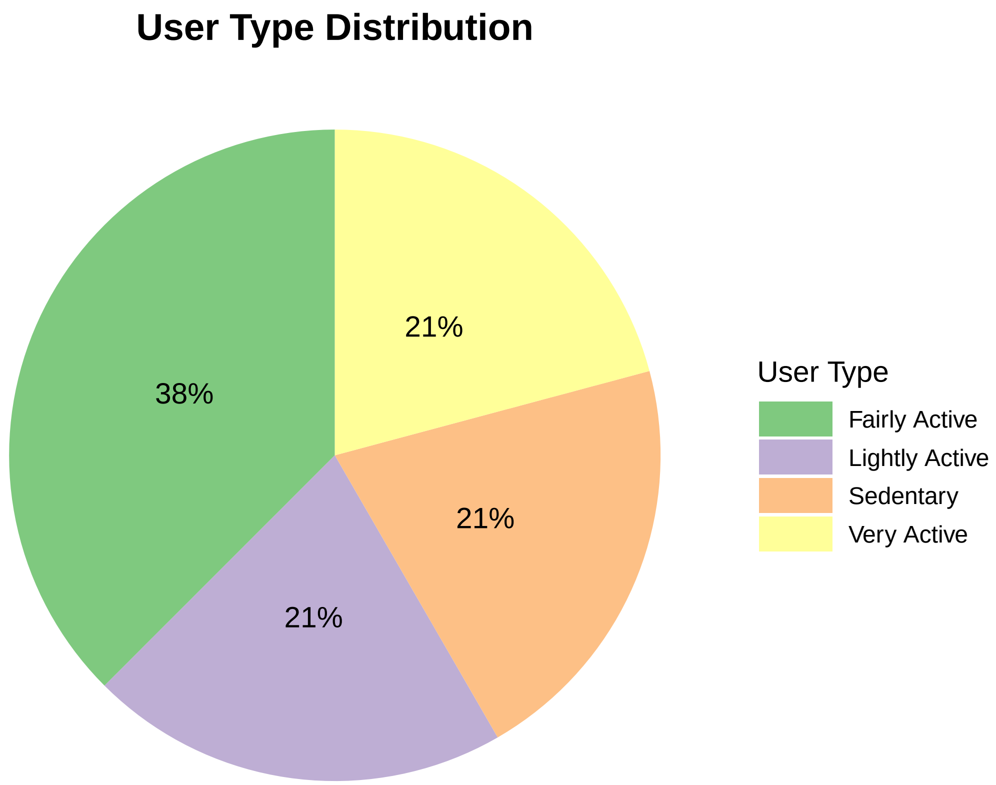
```
## # A tibble: 6 x 6
##           id mean_daily_distance mean_daily_calories mean_daily_sleep
##        <dbl>               <dbl>               <dbl>            <dbl>
## 1 1503960366                7.97               1872.             360.
## 2 1644430081                5.79               2978.             294
## 3 1844505072                2.30               1676.             652
## 4 1927972279                1.03               2316.             417
## 5 2026352035                3.49               1541.             506.
## 6 2320127002                3.42               1804               61
## # i 2 more variables: mean_daily_steps <dbl>, user_type <chr>
```

To visualize the distribution of user types, we compute the total percentage for each user type and represent it in a pie chart.

```r
user_type_percent <- user_type %>%
  group_by(user_type) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(user_type) %>%
  summarise(total_percent = total / totals) %>%
  mutate(labels = percent(total_percent))

head(user_type_percent)
```

```
## # A tibble: 4 x 3
##   user_type      total_percent labels
##   <chr>                  <dbl> <chr>
## 1 Fairly Active          0.375 38%
## 2 Lightly Active         0.208 21%
## 3 Sedentary              0.208 21%
## 4 Very Active            0.208 21%
```

```r
user_type_percent %>%
  ggplot(aes(x = "", y = total_percent, fill = user_type)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  theme_void() +
  theme(plot.title = element_text(hjust = 0.5, size = 14, face = "bold")) +
  scale_fill_manual(values = brewer.pal(4, "Accent")) +
  geom_text(aes(label = labels), position = position_stack(vjust = 0.5)) +
  labs(title="User Type Distribution") +
  guides(fill = guide_legend(title = "User Type", keywidth = 2, keyheight = 1,))
```



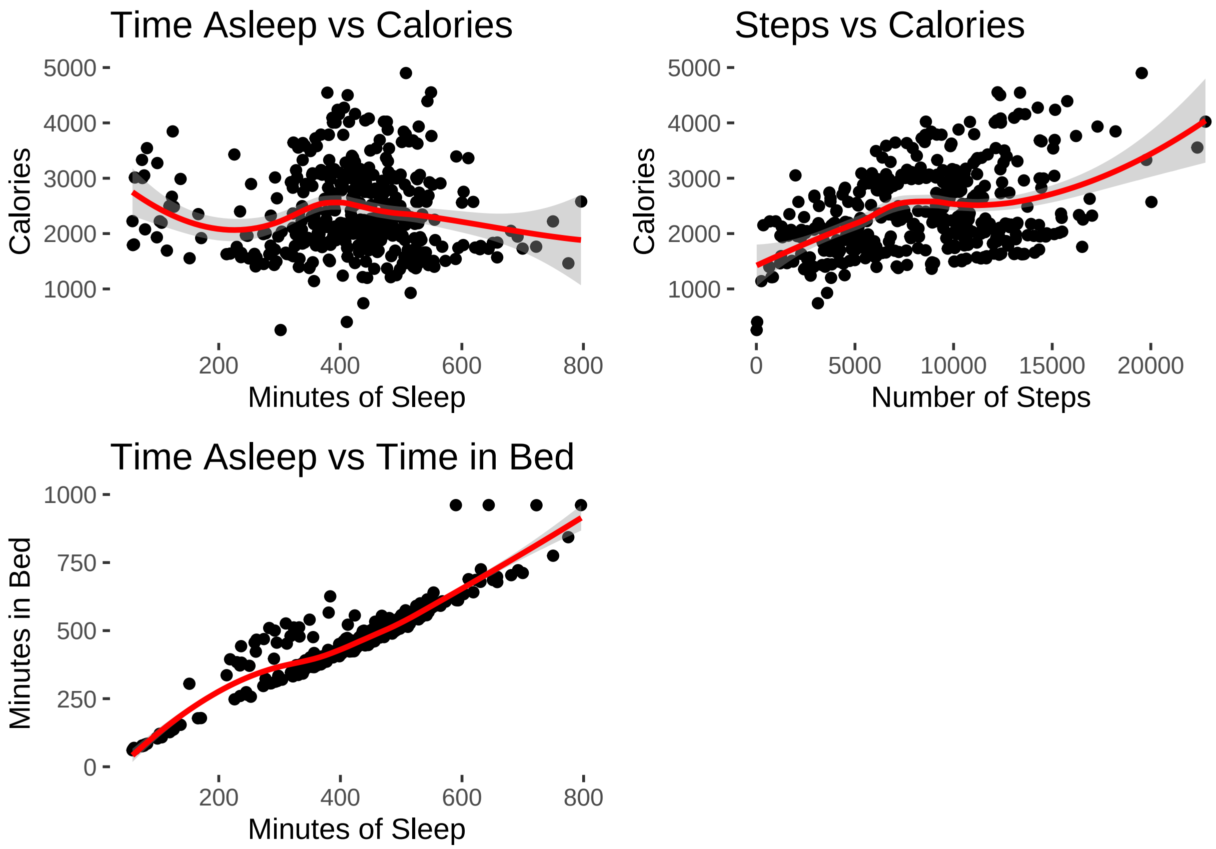
As observed from above, the majority of users can be considered fairly active.

Correlations

Let's explore the correlation between the duration of sleep and calories burned, as well as the relationship between the number of steps taken and calories burned.

```
ggarrange(
ggplot(merged_data, aes(x = totalminutesasleep, y = calories))+
  geom_jitter() +
  geom_smooth(color = "red") +
  labs(title = "Time Asleep vs Calories", x = "Minutes of Sleep", y = "Calories") +
   theme(panel.background = element_blank(),
        plot.title = element_text(size = 14)),
ggplot(merged_data, aes(x = totalsteps, y = calories))+
  geom_jitter() +
  geom_smooth(color = "red") +
  labs(title = "Steps vs Calories", x = "Number of Steps", y = "Calories") +
   theme(panel.background = element_blank(),
        plot.title = element_text(size = 14)),
ggplot(merged_data, aes(x = totalminutesasleep, y = totaltimeinbed))+
  geom_jitter() +
  geom_smooth(color = "red") +
  labs(title = "Time Asleep vs Time in Bed", x = "Minutes of Sleep", y = "Minutes in Bed") +
   theme(panel.background = element_blank(),
        plot.title = element_text(size = 14))
)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



While the data reveals no distinct correlation between calories burned and sleep duration, a positive correlation emerges between the number of steps taken and calories burned. Moreover, the positive correlation between total time asleep and total time in bed suggests that people who spend more time in bed tend to sleep longer.

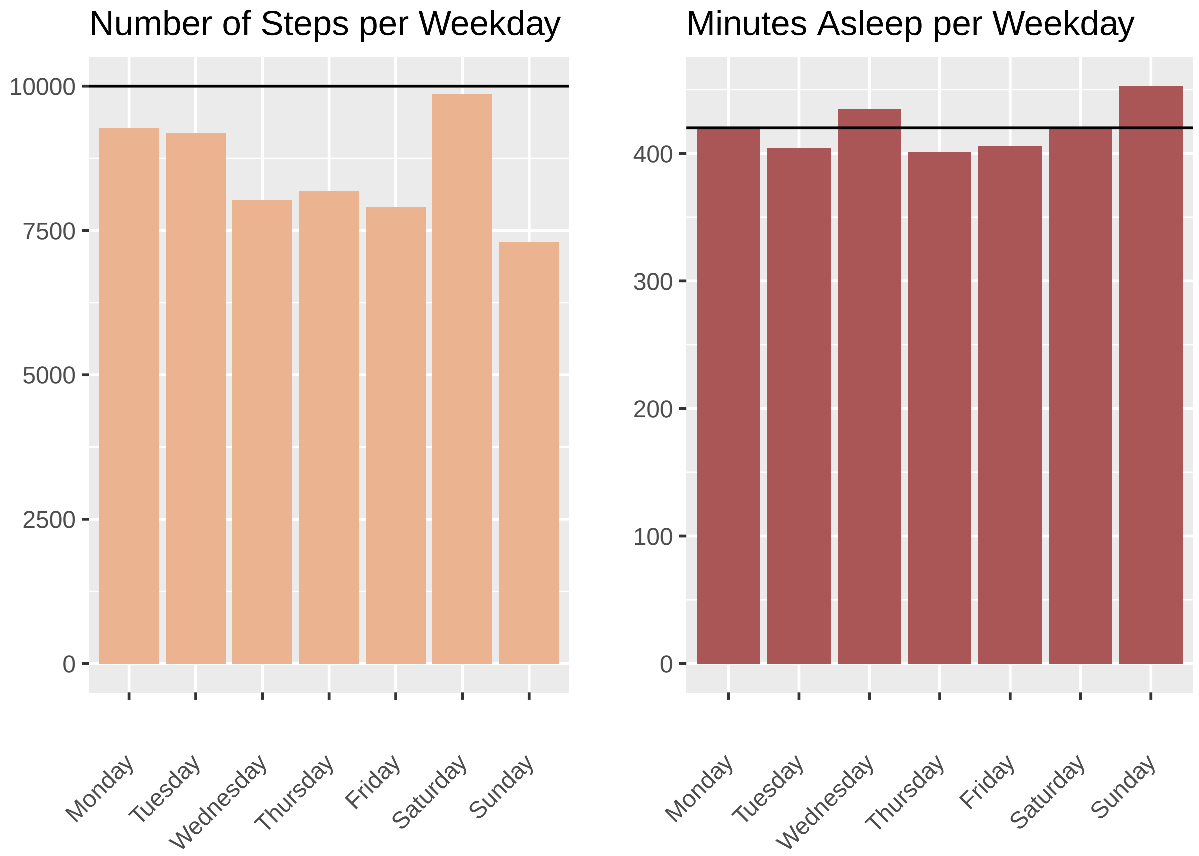
Observations of users' daily sleep and activity patterns over the week

Next, we analyze the users' sleep and activity during the week.

```
weekday_steps_sleep <- merged_data %>%
  mutate(weekday = weekdays(date))

weekday_steps_sleep$weekday <- ordered(weekday_steps_sleep$weekday, levels = c("Monday", "Tuesday", "Wed

 weekday_steps_sleep <- weekday_steps_sleep %>%
  group_by(weekday) %>%
  summarize (daily_steps = mean(totalsteps), daily_sleep = mean(totalminutesasleep))

head(weekday_steps_sleep)
```

```
## # A tibble: 6 x 3
##   weekday   daily_steps daily_sleep
##   <ord>           <dbl>       <dbl>
## 1 Monday          9273.        420.
## 2 Tuesday         9183.        405.
## 3 Wednesday       8023.        435.
## 4 Thursday        8184.        401.
## 5 Friday          7901.        405.
## 6 Saturday        9871.        419.
```

```
ggarrange(
    ggplot(weekday_steps_sleep) +
      geom_col(aes(weekday, daily_steps), fill = "#ECB390") +
      geom_hline(yintercept = 10000) +
      labs(title = "Number of Steps per Weekday", x= "", y = "") +
      theme(axis.text.x = element_text(angle = 45,vjust = 0.5, hjust = 1)),
    ggplot(weekday_steps_sleep, aes(weekday, daily_sleep)) +
      geom_col(fill = "#AA5656") +
      geom_hline(yintercept = 420) +
      labs(title = "Minutes Asleep per Weekday", x = "", y = "") +
      theme(axis.text.x = element_text(angle = 45,vjust = 0.5, hjust = 1))
  )
```

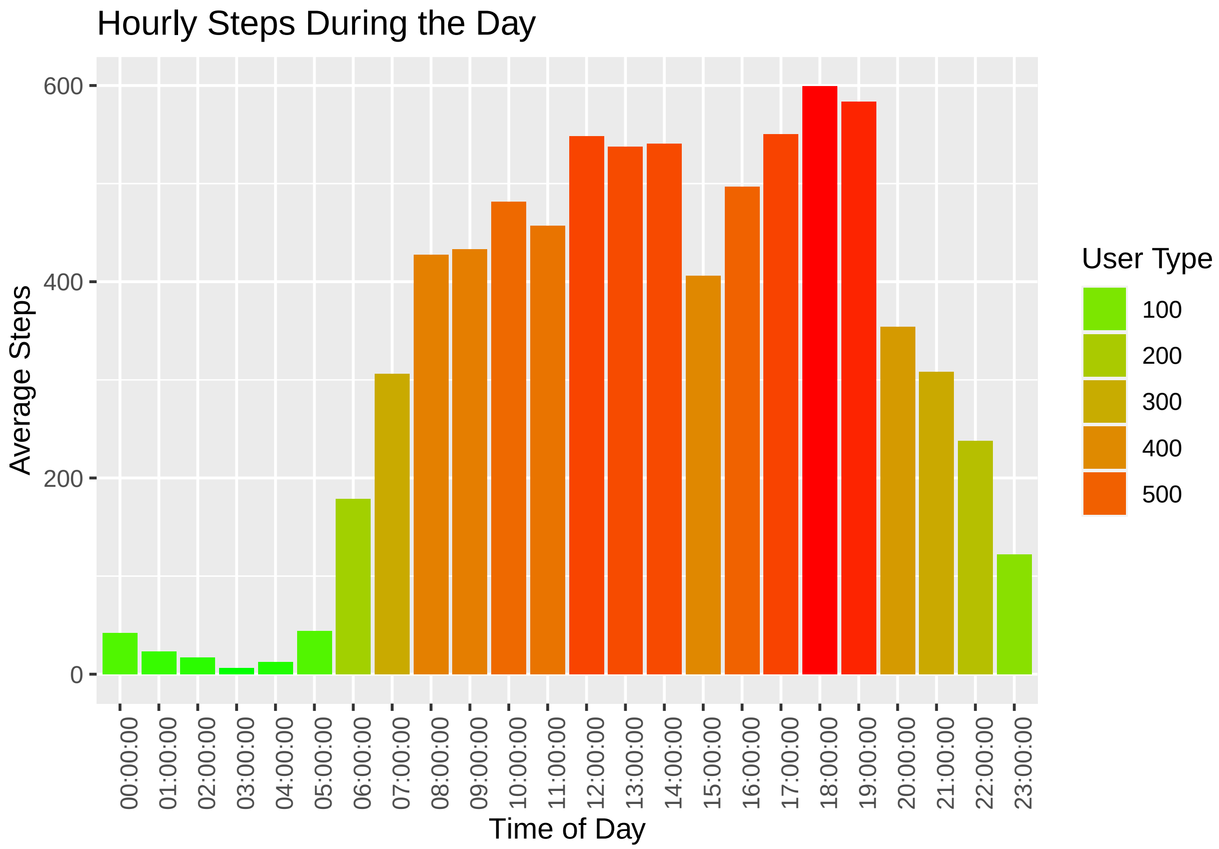


Based on the analysis, it’s evident that users exhibit increased activity levels on Saturdays, while Sundays see users getting more sleep. Additionally, users tend to fall short of the recommended daily step count (10,000), but generally achieve the recommended sleep duration of 7 hours or more.

Observations of users' step patterns throughout the day.

We now delve into the analysis of users' daily step patterns.

```
daily_steps %>%
  group_by(time) %>%
  summarize(average_steps = mean(steptotal)) %>%
  ggplot() +
  geom_col(mapping = aes(x = time, y = average_steps, fill = average_steps)) +
  labs(title = "Hourly Steps During the Day", x = "Time of Day", y = "Average Steps") +
  scale_fill_gradient(low = "green", high = "red")+
  guides(fill = guide_legend(title = "User Type"))+
  theme(axis.text.x = element_text(angle = 90))
```



The visualization shows users are most active during the day, peaking at 6:00 PM. Notably, heightened activity is observed from 12:00 PM to 2:00 PM and 5:00 PM to 7:00 PM.

Observations of users' device utilization

Now that we've observed patterns in activity, sleep, and calorie expenditure, our focus shifts to understanding the frequency of device usage among our sampled users. This exploration is crucial for shaping our marketing strategy and identifying features that enhance smart device utilization.

To quantify usage frequency, we'll categorize our sample based on a 31-day interval:

- High use: Users employing their devices between 21 and 31 days.

- Moderate use: Users utilizing their devices for 10 to 20 days.
- Low use: Users engaging with their devices for 1 to 10 days.

```
daily_use <- merged_data %>%
  group_by(id) %>%
  summarize(days_used = sum(n())) %>%
  mutate(usage = case_when(
    days_used >= 1 & days_used <= 10 ~ "Low Use",
    days_used >= 11 & days_used <= 20 ~ "Moderate Use",
    days_used >= 21 & days_used <= 31 ~ "High Use",
  ))

head(daily_use)
```

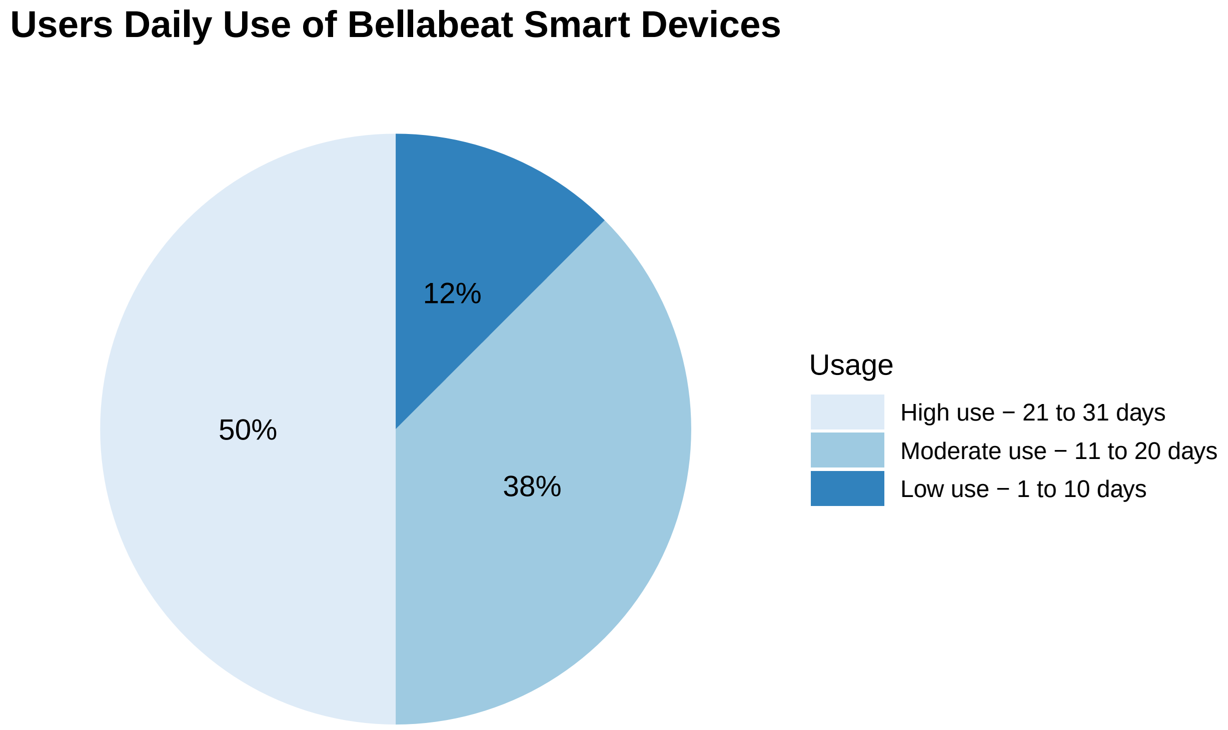
```
## # A tibble: 6 x 3
##           id days_used usage
##        <dbl>     <int> <chr>
## 1 1503960366        25 High Use
## 2 1644430081         4 Low Use
## 3 1844505072         3 Low Use
## 4 1927972279         5 Low Use
## 5 2026352035        28 High Use
## 6 2320127002         1 Low Use
```

To visually represent the diversity in users' utilization of the smart product, we calculate the overall percentage for each category of usage and illustrate it through a pie chart.

```
daily_use_percent <- daily_use %>%
  group_by(usage) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(usage) %>%
  summarise(total_percent = total / totals) %>%
  mutate(labels = percent(total_percent))

head(daily_use_percent)
```

```
## # A tibble: 3 x 3
##   usage        total_percent labels
##   <chr>                <dbl> <chr>
## 1 High Use             0.5   50%
## 2 Low Use              0.375 38%
## 3 Moderate Use         0.125 12%
```

```
daily_use_percent %>%
  ggplot(aes(x = "",y = total_percent, fill = usage)) +
  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")) +
  geom_text(aes(label = labels),
            position = position_stack(vjust = 0.5))+
  scale_fill_brewer(palette = "Blues",
                    labels = c("High use - 21 to 31 days",
                                 "Moderate use - 11 to 20 days",
                                 "Low use - 1 to 10 days"))+
  labs(title = "Users Daily Use of Bellabeat Smart Devices")+
  guides(fill = guide_legend(title = "Usage", keywidth = 2, keyheight = 1,))
```

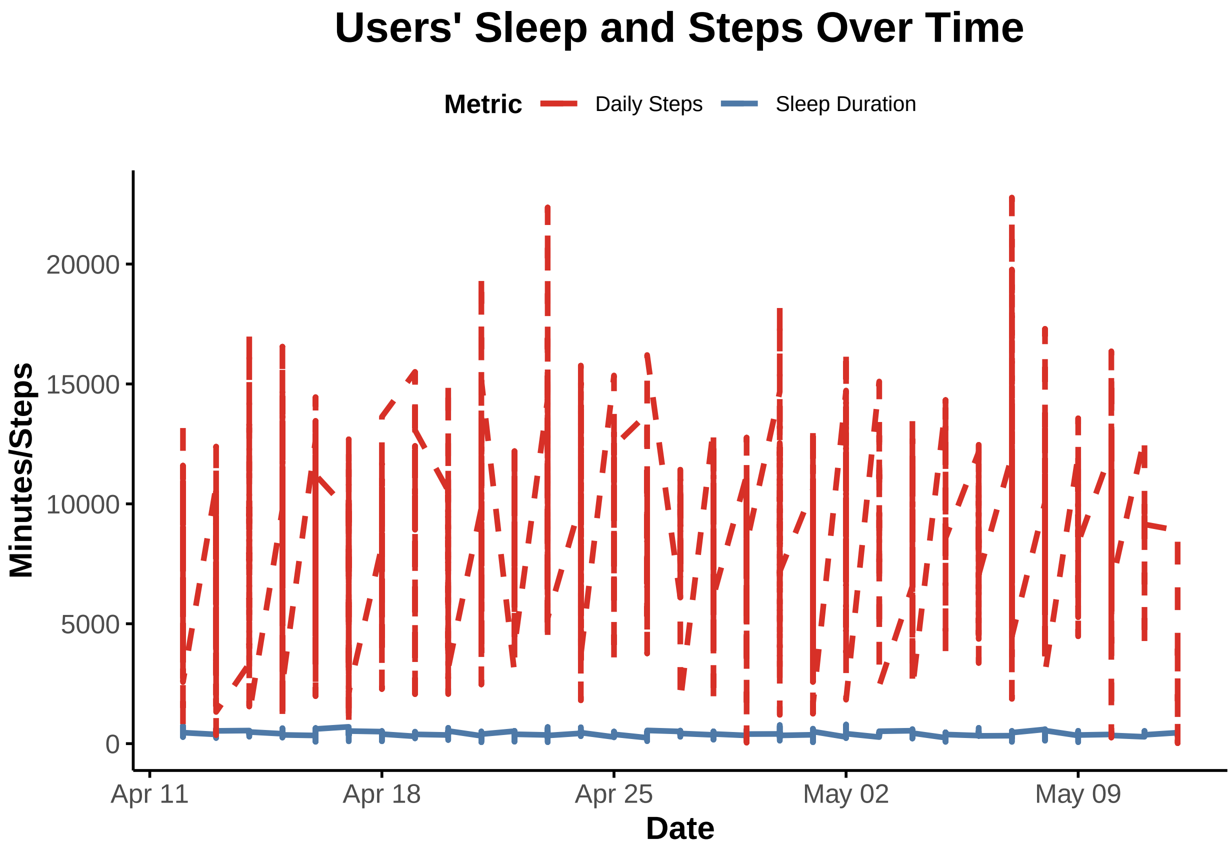


As shown above, a majority of users exhibit a high level of product engagement, with a notable portion demonstrating moderate use that presents an opportunity for improvement.

Observations of trends in users' sleep and step habits during this timeframe.

Let's examine whether users have increased or decreased their sleep and activity levels during this period.

```
ggplot(merged_data, aes(x = date)) +
  geom_line(aes(y = totalminutesasleep, color = "Sleep Duration"), linewidth = 1, linetype = "solid") +
  geom_line(aes(y = totalsteps, color = "Daily Steps"), linewidth = 1, linetype = "dashed") +
  labs(title = "Users' Sleep and Steps Over Time",
       x = "Date",
       y = "Minutes/Steps",
       color = "Metric") +
  theme_minimal() +
  scale_color_manual(values = c("Sleep Duration" = "#4e79a7", "Daily Steps" = "#d73027")) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.line = element_line(color = "black"),
        legend.position = "top",
        legend.title = element_text(face = "bold", size = 10),
        legend.text = element_text(size = 8),
        plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
        axis.title = element_text(size = 12, face = "bold"),
        axis.text = element_text(size = 10),
        axis.ticks = element_line(color = "black"))
```



Based on the provided line chart depicting users' sleep duration and daily steps over time, it appears that there is no clear or specific trend in the relationship between these two variables. The lines representing sleep duration and daily steps show fluctuations without a consistent upward or downward pattern.

---

5. Act Phase

Condsidering the analysis, we can make the following recommendations to help improve Bellabeat marketing strategy:

- Daily Step Notifications: We categorized users into four groups and found that, on average, users walk over 7,500 steps daily, excluding Sundays. To encourage meeting the CDC (Centers for Disease Control and Prevention)'s recommended daily step goal of 10,000, we propose sending alarms to users falling short and creating app posts highlighting the benefits. CDC states higher step count correlates with lower mortality rates, and we also observed a positive link between steps and calories.

- Sleep Improvement Notifications: Our data indicates users sleep less than 8 hours daily. To address this, users can set a bedtime and receive pre-sleep notifications. Additionally, we can provide resources such as breathing exercises, calming music podcasts, and sleep techniques to aid in better sleep.

- Reward System for Activity: Recognizing that not all users respond to notifications, we suggest a time-limited in-app game. Users progress through levels based on daily step counts, requiring consistent activity for rewards. Earned stars can be redeemed for merchandise or discounts on Bellabeat products.

- Product Features Promotion: Despite our analysis revealing that only 50% of users engage with their device daily, we continue to promote Bellabeat's product features, including water resistance, long-lasting batteries, and fashionable designs suitable for any occasion without battery concerns.